# MODULE 1

# INTRODUCTION AND PHYSICAL LAYER

## Introduction

Each of the past 3 centuries was dominated by a single new technology. The 18th century was the era of the great mechanical systems accompanying the Industrial Revolution. The 19th century was the age of the steam engine. During the 20th century, the key technology was information gathering, processing, & distribution. Among other developments, we saw the installation of worldwide telephone networks, the invention of radio & television, the birth & unprecedented growth of the computer industry, the launching of communication satellites, &, of course, the Internet.

As a result of rapid technological progress, these areas are rapidly converging in the 21st century & the differences between collecting, transporting, storing, & processing information are quickly disappearing. Organizations with hundreds of offices spread over a wide geographical area routinely expect to be able to examine the current status of even their most remote outpost at the push of a button. As our ability to gather, process, & distribute information grows, the demand for ever more sophisticated information processing grows even faster.

Although the computer industry is still young compared to other industries (e.g., automobiles & air transportation), computers have made spectacular progress in a short time. During the 1st 2 decades of their existence, computer systems were highly centralized, usually within a single large room. Not infrequently, this room had glass walls, through which visitors could gawk at the great electronic wonder inside. A medium-sized company or university might have had 1 or 2 computers, while very large institutions had at most a few dozen. The idea that within 40 years vastly more powerful computers smaller than postage stamps would be mass produced by the billions was pure science fiction.

The merging of computers & communications has had a profound influence on the way computer systems are organized. The once-dominant concept of the ‘‘computer center’’ as a room with a large computer to which users bring their work for processing is now totally obsolete (although data centers holding thousands of Internet servers are becoming common). The old model of a single computer serving all of the organization’s computational needs has

been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called **computer networks**.

‘‘Computer network’’ means a collection of autonomous computers interconnected by a single technology. 2 computers are said to be interconnected if they can exchange information. The connection need not be via a copper wire; fiber optics, microwaves, infrared, & communication satellites can also be used. Networks come in many sizes, shapes & forms, as we will see later. They are usually connected to make larger networks, with the **Internet** being the most well-known example of a network of networks.

There is considerable confusion in the literature between a computer network and a **distributed system**. The key distinction is that in a distributed system, a collection of independent computers appears to its users as a single coherent system. Usually, it has a single model or paradigm that it presents to the users. Often a layer of software on top of the operating system, called **middleware**, is responsible for implementing this model. A well-known example of a distributed system is the **World Wide Web**. It runs on top of the Internet & presents a model in which everything looks like a document (Web page).

In a computer network, this coherence, model, & software are absent. Users are exposed to the actual machines, without any attempt by the system to make the machines look & act in a coherent way. If the machines have different hardware & different operating systems, that is fully visible to the users. If a user wants to run a program on a remote machine, he must log onto that machine & run it there.

In effect, a distributed system is a software system built on top of a network. The software gives it a high degree of cohesiveness & transparency. Thus, the distinction between a network & a distributed system lies with the software (especially the operating system), rather than with the hardware.

Nevertheless, there is considerable overlap between the 2 subjects. For example, both distributed systems & computer networks need to move files around. The difference lies in who invokes the movement, the system or the user.

## 1.1 Uses of Computer Networks

Before we start to examine the technical issues in detail, it is worth devoting some time to pointing out why people are interested in computer networks & what they can be used for. After all, if nobody were interested in computer networks, few of them would be built. We will start with traditional uses at companies, then move on to home networking and recent developments regarding mobile users, & finish with social issues.

### 1.1.1 Business Applications

Most companies have a substantial number of computers. For example, a company may have a computer for each worker & use them to design products, write brochures, & do the payroll. Initially, some of these computers may have worked in isolation from the others, but at some point, management may have decided to connect them to be able to distribute information throughout the company.

Put in slightly more general form, the issue here is **resource sharing**. The goal is to make all programs, equipment, & especially data available to anyone on the network without regard to the physical location of the resource or the user. An obvious & widespread example is having a group of office workers share a common printer. None of the individuals really needs a private printer, & a high-volume networked printer is often cheaper, faster, & easier to maintain than a large collection of individual printers.

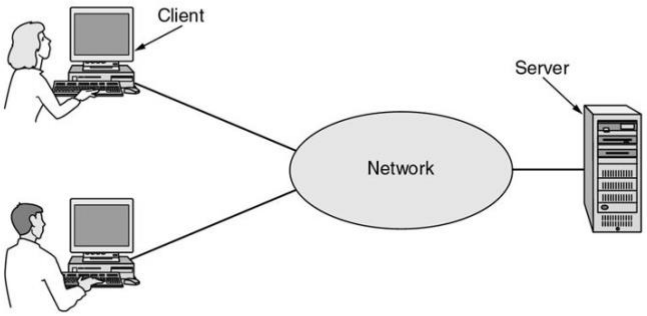
However, probably even more important than sharing physical resources such as printers, & tape backup systems, is sharing information. Companies small & large are vitally dependent on computerized information. Most companies have customer records, product information, inventories, financial statements, tax information, & much more online. If all its computers suddenly went down, a bank could not last more than 5 minutes. A modern manufacturing plant, with a computer-controlled assembly line, would not last even 5 seconds. Even a small travel agency or 3-person law firm is now highly dependent on computer networks for allowing employees to access relevant information & documents instantly.

For smaller companies, all the computers are likely to be in a single office or perhaps a single building, but for larger ones, the computers & employees may be scattered over dozens of offices & plants in many countries. Nevertheless, a salesperson in New York might sometimes need access to a product inventory database in Singapore. Networks called **VPNs (Virtual Private**

**Networks)** may be used to join the individual networks at different sites into one extended network. In other words, the mere fact that a user happens to be 15,000 km away from his data should not prevent him from using the data as though they were local. This goal may be summarized by saying that it is an attempt to end the ‘‘tyranny of geography”.

In the simplest of terms, one can imagine a company’s information system as consisting of 1 or more databases with company information & some number of employees who need to access them remotely. In this model, the data are stored on powerful computers called **servers**. Often these are centrally housed & maintained by a system administrator. In contrast, the employees have simpler machines, called **clients**, on their desks, with which they access remote data, for example, to include in spreadsheets they are constructing. (Sometimes we will refer to the human user of the client machine as the ‘‘client,’’ but it should be clear from the context whether we mean the computer or its user.) The client & server machines are connected by a network, as illustrated in Fig. 1-1. Note that we have shown the network as a simple oval, without any detail. We will use this form when we mean a network in the most abstract sense. When more detail is required, it will be provided.

## Business Applications of Networks



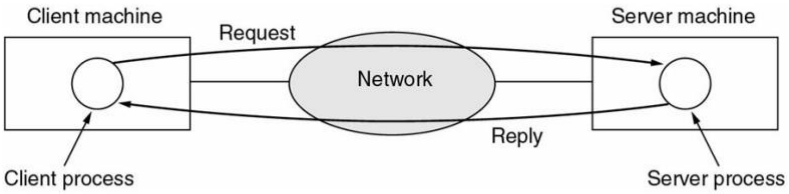

A network with two clients and one server.

This whole arrangement is called the **client-server model**. It is widely used & forms the basis of much network usage. The most popular realization is that of a **Web application**, in which the server generates Web pages based on its database in response to client requests that may update the database. The

client-server model is applicable when the client & server are both in the same building (& belong to the same company), but also when they are far apart. For example, when a person at home accesses a page on the World Wide Web, the same model is employed, with the remote Web server being the server & the user's personal computer being the client. Under most conditions, one server can handle a large number (hundreds or thousands) of clients simultaneously.

If we look at the client-server model in detail, we see that, 2 processes (i.e., running programs) are involved, 1 on the client machine & 1 on the server machine. Communication takes the form of the client process sending a message over the network to the server process. The client process then waits for a reply message. When the server process gets the request, it performs the requested work or looks up the requested data & sends back a reply. These messages are shown in Fig. 1-2.

## Business Applications of Networks (2)




The client-server model involves requests and replies.

A 2nd goal of setting up a computer network has to do with people rather than information or even computers. A computer network can provide a powerful **communication medium** among employees. Virtually every company that has 2 or more computers now has **email (electronic mail),** which employees generally use for a great deal of daily communication. In fact, a common gripe around the water cooler is how much email everyone must deal with, much of it quite meaningless because bosses have discovered that they can send the same (often content-free) message to all their subordinates at the push of a button.

Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called **IP telephony** or **Voice over IP (VoIP)** when Internet technology is used. The microphone & speaker at each end may belong to a VoIP-enabled phone or the employee's computer. Companies find this a wonderful way to save on their telephone bills.

Other, richer forms of communication are made possible by computer networks. Video can be added to audio so that employees at distant locations can see & hear each other as they hold a meeting. This technique is a powerful tool for eliminating the cost & time previously devoted to travel. **Desktop sharing** lets remote workers see & interact with a graphical computer screen. This makes it easy for 2 or more people who work far apart to read & write a shared blackboard or write a report together. When 1 worker makes a change to an online document, the others can see the change immediately, instead of waiting several days for a letter. Such a speedup makes cooperation among far-flung groups of people easy where it previously had been impossible. More ambitious forms of remote coordination such as telemedicine are only now starting to be used (e.g., remote patient monitoring) but may become much more important. It is sometimes said that communication & transportation are having a race, & whichever win will make the other obsolete.

A 3rd goal for many companies is doing business electronically, especially with customers & suppliers. This new model is called **e-commerce (electronic commerce)** & it has grown rapidly in recent years. Airlines, bookstores, & other retailers have discovered that many customers like the convenience of shopping from home. Consequently, many companies provide catalogs of their goods & services online & take orders online. Manufacturers of automobiles, aircraft, & computers, among others, buy subsystems from a variety of suppliers & then assemble the parts. Using computer networks, manufacturers can place orders electronically as needed. This reduces the need for large inventories & enhances efficiency.

### 1.1.2 Home Applications

In 1977, Ken Olsen was president of the Digital Equipment Corporation, then the number 2 computer vendor in the world (after IBM). When asked why Digital was not going after the personal computer market in a big way, he said: "There is no reason for any individual to have a computer in his home".

History showed otherwise & Digital no longer exists. People initially bought computers for word processing & games. Recently, the biggest reason to buy a home computer was probably for Internet access. Now, many consumers electronic devices, such as set-top boxes, game consoles, & clock radios, come with embedded computers & computer networks, especially wireless networks, & home networks are broadly used for entertainment, including listening to, looking at, & creating music, photos, & videos.

Internet access provides home users with **connectivity** to remote computers. As with companies, home users can access information, communicate with other people, & buy products & services with e-commerce. The main benefit now comes from connecting outside of the home. Bob Metcalfe, the inventor of Ethernet, hypothesized that the value of a network is proportional to the square of the number of users because this is roughly the number of different connections that may be made (Gilder, 1993). This hypothesis is known as "Metcalfe's law". It helps to explain how the tremendous popularity of the Internet comes from its size.

Access to remote information comes in many forms. It can be surfing the World Wide Web for information or just for fun. Information available includes the arts, business, cooking, government, health, history, hobbies, recreation, science, sports, travel, & many others. Fun comes in too many ways to mention, plus some ways that are better left unmentioned.

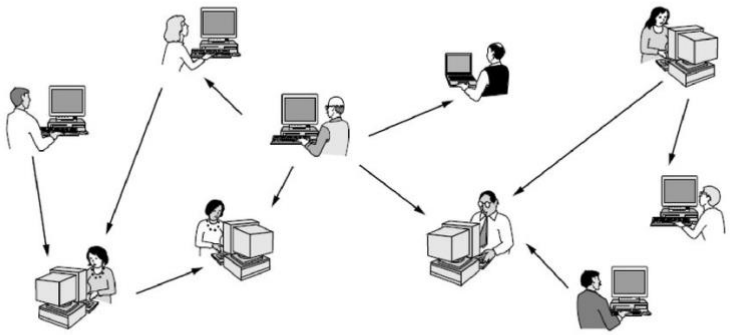
Many newspapers have gone online & can be personalized. For example, it is sometimes possible to tell a newspaper that you want everything about corrupt politicians, big fires, scandals involving celebrities, & epidemics, but no football, thank you. Sometimes it is possible to have the selected articles downloaded to your computer while you sleep. As this trend continues, it will cause massive unemployment among 12-year-old paperboys, but newspapers like it because distribution has always been the weakest link in the whole production chain. Of course, to make this model work, they will 1st have to figure out how to make money in this new world, something not entirely obvious since Internet users expect everything to be free.

The next step beyond newspapers (plus magazines & scientific journals) is the online digital library. Many professional organizations, such as the ACM (www.acm.org) & the IEEE Computer Society (www.computer.org), already have all their journals & conference proceedings online. Electronic book readers & online libraries may make printed books obsolete. Skeptics should

take note of the effect the printing press had on the medieval illuminated manuscript.

Much of this information is accessed using the client-server model, but there is different, popular model for accessing information that goes by the name of **peer-to-peer** communication. In this form, individuals who form a loose group can communicate with others in the group, as shown in Fig. 1-3. Every person can, in principle, communicate with one or more other people; there is no fixed division into clients & servers.

Home Applications (1)



In a peer-to-peer system there are no fixed clients and servers.

*Computer Networks*, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Many peer-to-peer systems, such BitTorrent (Cohen, 2003), don't have any central database of content. Instead, each user maintains his own database locally & provides a list of other nearby people who are members of the system. A new user can then go to any existing member to see what he has & get the names of other members to inspect for more content & more names. This lookup process can be repeated indefinitely to build up a large local database of what is out there. It is an activity that would get tedious for people, but computers excel at it.

Peer-to-peer communication is often used to share music & videos. It really hit the big time around 2000 with a music sharing service called Napster that was shut down after what was probably the biggest copyright infringement case in all recorded history. Legal applications for peer-to-peer communication also exist. These include fans sharing public domain music,

families sharing photos & movies, & users downloading public software packages. In fact, one of the most popular Internet applications of all, email, is inherently peer-to-peer. This form of communication is likely to grow considerably in the future.

All the above applications involve interactions between a person & a remote database full of information. The 2nd broad category of network use is person-to-person communication, basically the 21st century's answer to the 19th century's telephone. E-mail is already used daily by millions of people all over the world & its use is growing rapidly. It already routinely contains audio & video as well as text & pictures. Smell may take a while.

Any teenager worth his or her salt is addicted to **instant messaging**. This facility, derived from the UNIX talk program in use since around 1970, allows 2 people to type messages at each other in real time. There are multi-person messaging services too, such as the **Twitter** service that lets people send short text messages called ''tweets'' to their circle of friends or other willing audiences.

The Internet can be used by applications to carry audio (e.g., Internet radio stations) & video (e.g., YouTube). Besides being a cheap way to call to distant friends, these applications can provide rich experiences such as telelearning, meaning attending 8 A.M. classes without the inconvenience of having to get out of bed 1st. In the long run, the use of networks to enhance human-to-human communication may prove more important than any of the others. It may become hugely important to people who are geographically challenged, giving them the same access to services as people living in the middle of a big city.

Between person-to-person communications & accessing information are **social network** applications. Here, the flow of information is driven by the relationships that people declare between each other. One of the most popular social networking sites is **Facebook**. It lets people update their personal profiles & shares the updates with other people who they have declared to be their friends. Other social networking applications can make introductions via friends of friends, send news messages to friends such as Twitter above, & much more.

Even more loosely, groups of people can work together to create content. A **wiki**, for example, is a collaborative Web site that the members of a

community edit. The most famous wiki is the **Wikipedia**, an encyclopedia anyone can edit, but there are thousands of other wikis.

Our 3rd category is electronic commerce in the broadest sense of the term. Home shopping is already popular & enables users to inspect the online catalogs of thousands of companies. Some of these catalogs are interactive, showing products from different viewpoints a in configurations that can be personalized. After the customer buys a product electronically but can't figure out how to use it, online technical support may be consulted.

Another area in which e-commerce is widely used is access to financial institutions. Many people already pay their bills, manage their bank accounts, & handle their investments electronically. This trend will surely continue as networks become more secure.

One area that virtually nobody foresaw is electronic flea markets (e-flea?). Online auctions of 2nd-hand goods have become a massive industry. Unlike traditional e-commerce, which follows the client-server model, online auctions are peer-to-peer in the sense that consumers can act as both buyers & sellers.

Some of these forms of e-commerce have acquired cute little tags since "to" & ''2'' are pronounced the same. The most popular ones are listed in Fig. 1-4.

**Electronic commerce**

Our fourth category is electronic commerce in the broadest sense of the term, such as, home shopping, access to financial institutions, electronic flea markets.

| Tag | Full name | Example |
|---|---|---|
| B2C | Business-to-consumer | Ordering books on-line |
| B2B | Business-to-business | Car manufacturer ordering tires from supplier |
| G2C | Government-to-consumer | Government distributing tax forms electronically |
| C2C | Consumer-to-consumer | Auctioning second-hand products on-line |
| P2P | Peer-to-peer | File sharing |

Fig 1.4 Some forms of e-commerce.

Our 4th category is entertainment. This has made huge strides in the home in recent years, with the distribution of music, radio & television programs, &

movies over the Internet beginning to rival that of traditional mechanisms. Users can find, buy, & download MP3 songs & DVD-quality movies & add them to their personal collection. TV shows now reach many homes via **IPTV (IP Tele Vision)** systems that are based on IP technology instead of cable TV or radio transmissions. Media streaming applications let users tune into Internet radio stations or watch recent episodes of their favorite TV shows. Naturally, all this content can be moved around your house between different devices, displays & speakers, usually with a wireless network.

Soon, it may be possible to search for any movie or television program ever made, in any country, & have it displayed on your screen instantly. New films may become interactive, where the user is occasionally prompted for the story direction (should Macbeth murder Duncan or just bide his time?) with alternative scenarios provided for all cases. Live television may also become interactive, with the audience participating in quiz shows, choosing among contestants, & so on.

Another form of entertainment is game playing. Already we have multiperson real-time simulation games, like hide-&-seek in a virtual dungeon, & flight simulators with the players on 1 team trying to shoot down the players on the opposing team. Virtual worlds provide a persistent setting in which thousands of users can experience a shared reality with three-dimensional graphics.

Our last category is **ubiquitous computing**, in which computing is embedded into everyday life, as in the vision of Mark Weiser (1991). Many homes are already wired with security systems that include door & window sensors, & there are many more sensors that can be folded into a smart home monitor, such as energy consumption. Your electricity, gas & water meters could also report usage over the network. This would save money as there would be no need to send out meter readers. And your smoke detectors could call the fire department instead of making a big noise (which has little value if no one is home). As the cost of sensing & communication drops, more & more measurement & reporting will be done with networks.

Increasingly, consumer electronic devices are networked. For example, some high-end cameras already have a wireless network capability & use it to send photos to a nearby display for viewing. Professional sports photographers can also send their photos to their editors in real-time, 1$^{st}$ wirelessly to an access point then over the Internet. Devices such as televisions that plug into the wall can use **power-line networks** to send information throughout the house over

the wires that carry electricity. It may not be very surprising to have these objects on the network, but objects that we don't think of as computers may sense & communicate information too. For example, your shower may record water usage, give you visual feedback while you lather up, & report to a home environmental monitoring application when you are done to help save on your water bill.

A technology called **RFID (Radio Frequency IDentification)** will push this idea even further in the future. RFID tags are passive (i.e., have no battery) chips the size of stamps & they can already be affixed to books, passports, pets, credit cards, & other items in the home & out. This lets RFID readers locate & communicate with the items over up to several meters, depending on the kind of RFID. Originally, RFID was commercialized to replace barcodes. It has not succeeded yet because barcodes are free & RFID tags cost a few cents. Of course, RFID tags offer much more & their price is rapidly declining. They may turn the real world into the Internet of things (ITU, 2005).

### 1.1.2 Mobile Users

Mobile computers, such as laptop and handheld computers, are one of the fastest-growing segments of the computer industry. Their sales have already overtaken those of desktop computers. Why would anyone want one? People on the go often want to use their mobile devices to read & send email, tweet, watch movies, download music, play games, or simply to surf the Web for information. They want to do all the things they do at home & in the office. Naturally, they want to do them from anywhere on land, sea or in the air.

**Connectivity** to the Internet enables many of these mobile uses. Since having a wired connection is impossible in cars, boats, & airplanes, there is a lot of interest in wireless networks. Cellular networks operated by the telephone companies are one familiar kind of wireless network that blankets us with coverage for mobile phones. Wireless **hotspots** based on the 802.11 standard are another kind of wireless network for mobile computers. They have sprung up everywhere that people go, resulting in a patchwork of coverage at cafes, hotels, airports, schools, trains & planes. Anyone with a laptop computer & a wireless modem can just turn on their computer on & be connected to the Internet through the hotspot, as though the computer were plugged into a wired network.

Wireless networks are of great value to fleets of trucks, taxis, delivery vehicles, & repairpersons for keeping in contact with their home base. For example, in many cities, taxi drivers are independent businessmen, rather than being employees of a taxi company. In some of these cities, the taxis have a display the driver can see. When a customer calls up, a central dispatcher types in the pickup & destination points. This information is displayed on the drivers' displays & a beep sounds. The 1st driver to hit a button on the display gets the call.

Wireless networks are also important to the military. If you must be able to fight a war anywhere on Earth at short notice, counting on using the local networking infrastructure is probably not a good idea. It is better to bring your own.

Although wireless networking & mobile computing are often related, they are not identical, as Fig. 1-5 shows. Here we see a distinction between **fixed wireless** & **mobile wireless** networks. Even notebook computers are sometimes wired. For example, if a traveler plugs a notebook computer into the wired network jack in a hotel room, he has mobility without a wireless network.

Mobile Users

| Wireless | Mobile | Typical applications |
|---|---|---|
| No | No | Desktop computers in offices |
| No | Yes | A notebook computer used in a hotel room |
| Yes | No | Networks in unwired buildings |
| Yes | Yes | Store inventory with a handheld computer |

Combinations of wireless networks and mobile computing

Conversely, some wireless computers are not mobile. In the home, & in offices or hotels that lack suitable cabling, it can be more convenient to connect desktop computers or media players wirelessly than to install wires. Installing a wireless network may require little more than buying a small box with some

electronics in it, unpacking it, & plugging it in. This solution may be far cheaper than having workmen put in cable ducts to wire the building.

Finally, there are also true mobile, wireless applications, such as people walking around stores with a handheld computer recording inventory. At many busy airports, car rental return clerks work in the parking lot with wireless mobile computers. They scan the barcodes or RFID chips of returning cars, & their mobile device, which has a built-in printer, calls the main computer, gets the rental information, & prints out the bill on the spot.

Perhaps the key driver of mobile, wireless applications is the mobile phone. **Text messaging** or **texting** is tremendously popular. It lets a mobile phone user type a short message that is then delivered by the cellular network to another mobile subscriber. Few people would have predicted 10 years ago that having teenagers tediously typing short text messages on mobile phones would be an immense money maker for telephone companies. But texting (or **Short Message Service** as it is known outside the U.S.) is very profitable since it costs the carrier but a tiny fraction of one cent to relay a text message, a service for which they charge far more.

The long-awaited convergence of telephones & the Internet has finally arrived, & it will accelerate the growth of mobile applications. **Smart phones**, such as the popular iPhone, combine aspects of mobile phones & mobile computers. The (3G & 4G) cellular networks to which they connect can provide fast data services for using the Internet as well as handling phone calls. Many advanced phones connect to wireless hotspots too, & automatically switch between networks to choose the best option for the user.

Other consumer electronics devices can also use cellular & hotspot networks to stay connected to remote computers. Electronic book readers can download a newly purchased book or the next edition of a magazine or today's newspaper wherever they roam. Electronic picture frames can update their displays on cue with fresh images.

Since mobile phones know their locations, often because they are equipped with **GPS (Global Positioning System)** receivers, some services are intentionally location dependent. Mobile maps & directions are an obvious candidate as your GPS-enabled phone & car probably have a better idea of where you are than you do. So, too, are searches for a nearby bookstore or Chinese restaurant, or a local weather forecast. Other services may record

location, such as annotating photos & videos with the place at which they were made. This annotation is known as “geo-tagging”.

An area in which mobile phones are now starting to be used is **m-commerce (mobile-commerce).** Short text messages from the mobile are used to authorize payments for food in vending machines, movie tickets, & other small items instead of cash & credit cards. The charge then appears on the mobile phone bill. When equipped with **NFC (Near Field Communication)** technology the mobile can act as an RFID smartcard & interact with a nearby reader for payment. The driving forces behind this phenomenon are the mobile device makers & network operators, who are trying hard to figure out how to get a piece of the e-commerce pie. From the store’s point of view, this scheme may save them most of the credit card company’s fee, which can be several percent. Of course, this plan may backfire, since customers in a store might use the RFID or barcode readers on their mobile devices to check out competitors’ prices before buying & use them to get a detailed report on where else an item can be purchased nearby & at what price.

One huge thing that m-commerce has going for it is that mobile phone users are accustomed to paying for everything (in contrast to Internet users, who expect everything to be free). If an Internet Web site charged a fee to allow its customers to pay by credit card, there would be an immense howling noise from the users. If, however, a mobile phone operator its customers to pay for items in a store by waving the phone at the cash register & then tacked on a fee for this convenience, it would probably be accepted as normal. Time will tell.

No doubt the uses of mobile & wireless computers will grow rapidly in the future as the size of computers shrinks, probably in ways no one can now foresee. Let us take a quick look at some possibilities. **Sensor networks** are made up of nodes that gather & wirelessly relay information they sense about the state of the physical world. The nodes may be part of familiar items such as cars or phones, or they may be small separate devices. For example, your car might gather data on its location, speed, vibration, & fuel efficiency from its on-board diagnostic system & upload this information to a database. Those data can help find potholes, plan trips around congested roads, & tell you if you are a “gas guzzler” compared to other drivers on the same stretch of road.

Sensor networks are revolutionizing science by providing a wealth of data on behavior that could not previously be observed. One example is tracking the

migration of individual zebras by placing a small sensor on each animal. Researchers have packed a wireless computer into a cube 1 mm on edge. With mobile computers this small, even small birds, rodents, & insects can be tracked.

Even mundane uses, such as in parking meters, can be significant because they make use of data that were not previously available. Wireless parking meters can accept credit or debit card payments with instant verification over the wireless link. They can also report when they are in use over the wireless network. This would let drivers download a recent parking map to their car so they can find an available spot more easily. Of course, when a meter expires, it might also check for the presence of a car (by bouncing a signal off it) & report the expiration to parking enforcement. It has been estimated that city governments in the U.S. alone could collect an additional $10 billion this way.

**Wearable computers** are another promising application. Smart watches with radios have been part of our mental space since their appearance in the Dick Tracy comic strip in 1946; now you can buy them. Other such devices may be implanted, such as pacemakers & insulin pumps. Some of these can be controlled over a wireless network. This lets doctors test & reconfigure them more easily. It could also lead to some nasty problems if the devices are as insecure as the average PC & can be hacked easily.

### 1.1.3 Social Issues

Computer networks, like the printing press 500 years ago, allow ordinary citizens to distribute & view content in ways that were not previously possible. But along with the good comes the bad, as this new-found freedom brings with it many unsolved social, political, & ethical issues. Let us just briefly mention a few of them; a thorough study would require a full book, at least.

Social networks, message boards, content sharing sites, & a host of other applications allow people to share their views with like-minded individuals. If the subjects are restricted to technical topics or hobbies like gardening, not too many problems will arise.

The trouble comes with topics that people care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Worse yet, they may not be politically correct. Furthermore, opinions need not be limited to text; high-resolution color photographs & video clips are easily

shared over computer networks. Some people take a live-&-let-live view, but others feel that posting certain material (e.g., verbal attacks on countries or religions, pornography, etc.) is simply unacceptable & that such content must be censored. Different countries have different & conflicting laws in this area. Thus, the debate rages.

In the past, people have sued network operators, claiming that they are responsible for the contents of what they carry, just as newspapers & magazines are. The inevitable response is that a network is like a telephone company, or the post office & cannot be expected to police what its users say.

It should now come only as a slight surprise to learn that some network operators block content for their own reasons. Some users of peer-to-peer applications had their network service cut off because the network operators didn't find it profitable to carry the large amounts of traffic sent by those applications. Those same operators would probably like to treat different companies differently. If you are a big company & pay well then you get good service, but if you are a small-time player, you get poor service. Opponents of this practice argue that peer-to-peer & other content should be treated in the same way because they are all just bits to the network. This argument for communications that are not differentiated by their content or source or who is providing the content is known as **network neutrality**. It is probably safe to say that this debate will go on for a while.

Many other parties are involved in the tussle over content. For instance, pirated music & movies fueled the massive growth of peer-to-peer networks, which didn't please the copyright holders, who have threatened (& sometimes taken) legal action. There are now automated systems that search peer-to-peer networks & fire off warnings to network operators & users who are suspected of infringing copyright. In the United States, these warnings are known as **DMCA takedown notices** after the **Digital Millennium Copyright Act**. This search is an arms' race because it is hard to reliably catch copyright infringement. Even your printer might be mistaken for a culprit.

Computer networks make it very easy to communicate. They also make it easy for the people who run the network to snoop on the traffic. This sets up conflicts over issues such as employee rights versus employer rights. Many people read & write email at work. Many employers have claimed the right to read & possibly censor employee messages, including messages sent from a

home computer outside working hours. Not all employees agree with this, especially the latter part.

Another conflict is centered around government versus citizen's rights. The FBI has installed systems at many Internet service providers to snoop on all incoming & outgoing email for nuggets of interest. One early system was originally called Carnivore, but bad publicity caused it to be renamed to the more innocent-sounding DCS1000. The goal of such systems is to spy on millions of people in the hope of perhaps finding information about illegal activities. Unfortunately for the spies, the Fourth Amendment to the U.S. Constitution prohibits government searches without a search warrant, but the government often ignores it.

Of course, the government doesn't have a monopoly on threatening people's privacy. The private sector does its bit too by **profiling** users. For example, small files called **cookies** that Web browsers store on users' computers allow companies to track users' activities in cyberspace & may also allow credit card numbers, social security numbers, & other confidential information to leak all over the Internet. Companies that provide Web-based services may maintain large amounts of personal information about their users that allows them to study user activities directly. For example, Google can read your email & show you advertisements based on your interests if you use its email service, **Gmail**.

A new twist with mobile devices is location privacy. As part of the process of providing service to your mobile device the network operators learn where you are at different times of day. This allows them to track your movements. They may know which nightclub you frequent & which medical center you visit.

Computer networks also offer the potential to increase privacy by sending anonymous messages. In some situations, this capability may be desirable. Beyond preventing companies from learning your habits, it provides, for example, a way for students, soldiers, employees, & citizens to blow the whistle on illegal behavior on the part of professors, officers, superiors, & politicians without fear of reprisals. On the other hand, in the United States & most other democracies, the law specifically permits an accused person the right to confront & challenge his accuser in court so anonymous accusations cannot be used as evidence.

The Internet makes it possible to find information quickly, but a great deal of it is ill considered, misleading, or downright wrong. That medical advice you plucked from the Internet about the pain in your chest may have come from a Nobel Prize winner or from a high-school dropout.

Other information is frequently unwanted. Electronic junk mail (spam) has become a part of life because spammers have collected millions of email addresses & would-be marketers can cheaply send computer-generated messages to them. The resulting flood of spam rivals the flow messages from real people. Fortunately, filtering software can read & discard the spam generated by other computers, with lesser or greater degrees of success.

Still other content is intended for criminal behavior. Web pages & email messages containing active content (basically, programs or macros that execute on the receiver's machine) can contain viruses that take over your computer. They might be used to steal your bank account passwords, or to have your computer send spam as part of a **botne**t or pool of compromised machines.

**Phishing** messages masquerade as originating from a trustworthy party, for example, your bank, to try to trick you into revealing sensitive information, for example, credit card numbers. Identity theft is becoming a serious problem as thieves collect enough information about a victim to obtain credit cards & other documents in the victim's name.

It can be difficult to prevent computers from impersonating people on the Internet. This problem has led to the development of **CAPTCHA**s, in which a computer asks a person to solve a short recognition task, for example, typing in the letters shown in a distorted image, to show that they are human. This process is a variation on the famous Turing test in which a person asks questions over a network to judge whether the entity responding is human.

A lot of these problems could be solved if the computer industry took computer security seriously. If all messages were encrypted & authenticated, it would be harder to commit mischief. The problem is that hardware & software vendors know that putting in security features costs money & their customers arn't demanding such features. In addition, a substantial number of the problems are caused by buggy software, which occurs because vendors keep adding more & more features to their programs, which inevitably means more code & thus more bugs. A tax on new features might help, but that might

be a tough sell in some quarters. A refund for defective software might be nice, except it would bankrupt the entire software industry in the 1st year.

Computer networks raise new legal problems when they interact with old laws. Electronic gambling provides an example. Computers have been simulating things for decades, so why not simulate slot machines, roulette wheels, blackjack dealers, & more gambling equipment? Well, because it is illegal in a lot of places. The trouble is, gambling is legal in a lot of other places (England, for example) & casino owners there have grasped the potential for Internet gambling. But what happens if the gambler, the casino, & the server are all in different countries, with conflicting laws?

## 1.2 Network Hardware

It's now time to turn our attention from the applications & social aspects of networking (the dessert) to the technical issues involved in network design (the spinach). There is no generally accepted taxonomy into which all computer networks fit, but 2 dimensions stand out as important: transmission technology & scale. We will now examine each of these in turn.

Broadly speaking, there are 2 types of transmission technology that are in widespread use: broadcast links & point-to-point links.

Point-to-point links connect individual pairs of machines. To go from the source to the destination on a network made up of point-to-point links, short messages, called packets in certain contexts, may have to 1st visit one or more intermediate machines. Often multiple routes, of different lengths, are possible, so finding good ones is important in point-to-point networks. Point-to-point transmission with exactly 1 sender & exactly 1 receiver is sometimes called unicasting.

In contrast, on a broadcast network, the communication channel is shared by all the machines on the network; packets sent by any machine are received by all the others. An address field within each packet specifies the intended recipient. Upon receiving a packet, a machine checks the address field. If the packet is intended for the receiving machine, that machine processes the packet; if the packet is intended for some other machine, it is just ignored.

A wireless network is a common example of a broadcast link, with communication shared over a coverage region that depends on the wireless channel & the transmitting machine. As an analogy, consider someone standing in a meeting room & shouting "Watson, come here. I want you". Although the packet may be received (heard) by many people, only Watson will respond; the others just ignore it.

Broadcast systems usually also allow the possibility of addressing a packet to all destinations by using a special code in the address field. When a packet with this code is transmitted, it is received & processed by every machine on the network. This mode of operation is called broadcasting. Some broadcast systems also support transmission to a subset of the machines, which known as multicasting.

An alternative criterion for classifying networks is by scale. Distance is important as a classification metric because different technologies are used at different scales.

In Fig. 1-6 we classify multiple processor systems by their rough physical size. At the top are the personal area networks, networks that are meant for 1 person. Beyond these come longer-range networks. These can be divided into local, metropolitan, & wide area networks, each with increasing scale. Finally, the connection of 2 or more networks is called an internetwork. The worldwide Internet is certainly the best-known (but not the only) example of an internetwork. Soon we will have even larger internetworks with the Interplanetary Internet that connects networks across space.

## Broadcast Networks (2)

| Interprocessor distance | Processors located in same | Example |
|---|---|---|
| 1 m | Square meter | Personal area network |
| 10 m | Room | Local area network |
| 100 m | Building | |
| 1 km | Campus | |
| 10 km | City | Metropolitan area network |
| 100 km | Country | Wide area network |
| 1000 km | Continent | |
| 10,000 km | Planet | The Internet |

Classification of interconnected processors by scale.

### 1.2.1 Personal Area Networks (not important; not present in the syllabus)

PANs (Personal Area Networks) let devices communicate over the range of a person. A common example is a wireless network that connects a computer with its peripherals. Almost every computer has an attached monitor, keyboard, mouse, & printer. Without using wireless, this connection must be done with cables. So many new users have a hard time finding the right cables & plugging them into the right little holes (even though they are usually color coded) that most computer vendors offer the option of sending a technician to the user's home to do it. To help these users, some companies got together to design a short-range wireless network called Bluetooth to connect these components without wires. The idea is that if your devices have Bluetooth, then you need no cables. You just put them down, turn them on, & they work together. For many people, this ease of operation is a big plus.

In the simplest form, Bluetooth networks use the master-slave paradigm of Fig. 1-7. The system unit (the PC) is normally the master, talking to the mouse, keyboard, etc., as slaves. The master tells the slaves what addresses to use, when they can broadcast, how long they can transmit, what frequencies they can use, & so on.


Bluetooth PAN configuration

*Computer Networks*, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

Bluetooth can be used in other settings, too. It is often used to connect a headset to a mobile phone without cords & it can allow your digital music player to connect to your car merely being brought within range. A

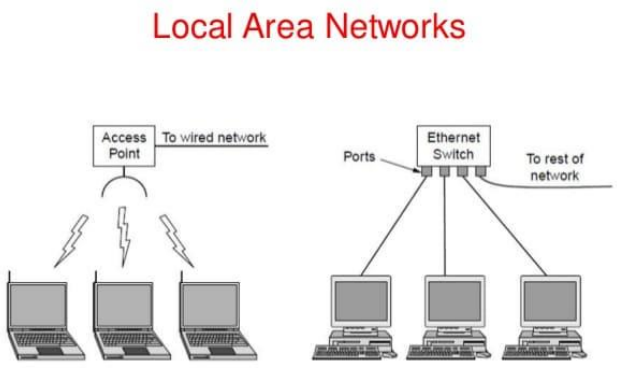
completely different kind of PAN is formed when an embedded medical device such as a pacemaker, insulin pump, or hearing aid talks to a user-operated remote control.

PANs can also be built with other technologies that communicate over short ranges, such as RFID on smartcards & library books.

### 1.2.2 Local Area Networks

The next step up is the LAN (Local Area Network). A LAN is a privately owned network that operates within & nearby a single building like a home, office or factory. LANs are widely used to connect personal computers & consumer electronics to let them share resources (e.g., printers) & exchange information. When LANs are used by companies, they are called enterprise networks.

Wireless LANs are very popular these days, especially in homes, older office buildings, cafeterias, & other places where it is too much trouble to install cables. In these systems, every computer has a radio modem & an antenna that it uses to communicate with other computers. In most cases, each computer talks to a device in the ceiling as shown in Fig. 1-8(a). This device, called an AP (Access Point), wireless router, or base station, relays packets between the wireless computers & also between them & the Internet. Being the AP is like being the popular kid as school because everyone wants to talk to you. However, if other computers are close enough, they can communicate directly with one another in a peer-to-peer configuration.




Wireless and wired LANs. (a) 802.11. (b) Switched Ethernet.

There is a standard for wireless LANs called IEEE 802.11, popularly known as Wi-Fi, which has become very widespread. It runs at speeds anywhere from 11 to hundreds of Mbps. We will adhere to tradition & measure line speeds in megabits/sec, where 1 Mbps is 1,000,000 bits/sec, & gigabits/sec, where 1 Gbps is 1,000,000,000 bits/sec.

Wired LANs use a range of different transmission technologies. Most of them use copper wires, but some use optical fiber. LANs are restricted in size, which means that the worst-case transmission time is bounded & known in advance. Knowing these bounds helps with the task of designing network protocols. Typically, wired LANs run at speeds of 100 Mbps to 1 Gbps, have low delay (microseconds or nanoseconds), & make very few errors. Newer LANs can operate at up to 10 Gbps. Compared to wireless networks, wired LANs exceed them in all dimensions of performance. It is just easier to send signals over a wire or through a fiber than through the air.

The topology of many wired LANs is built from point-to-point links. IEEE 802.3, popularly called Ethernet, is, by far, the most common type of wired LAN. Fig. 1-8(b) shows a sample topology of switched Ethernet. Each computer speaks the Ethernet protocol & connects to a box called a switch with a point-to-point link. Hence the name. A switch has multiple ports, each of which can connect to 1 computer. The job of the switch is to relay packets between computers that are attached to it, using the address in each packet to determine which computer to send it to.

To build larger LANs, switches can be plugged into each other using their ports. What happens if you plug them together in a loop? Will the network still work? Luckily, the designers thought of this case. It's the job of the protocol to sort out what paths packets should travel to safely reach the intended computer.

It is also possible to divide 1 large physical LAN into 2 smaller logical LANs. You might wonder why this would be useful. Sometimes, the layout of the network equipment does not match the organization's structure. For example, the engineering & finance departments of a company might have computers on the same physical LAN because they are in the same wing of the building, but it might be easier to manage the system if engineering & finance logically each had its own network Virtual LAN or VLAN. In this design each port is tagged with a "color", say green for engineering &

red for finance. The switch then forwards packets so that computers attached to the green ports are separated from the computers attached to the red ports. Broadcast packets sent on a red port, for example, won't be received on a green port, just as though there were 2 different LANs.

There are other wired LAN topologies too. In fact, switched Ethernet is a modern version of the original Ethernet design that broadcast all the packets over a single linear cable. At most 1 machine could successfully transmit at a time, & a distributed arbitration mechanism was used to resolve conflicts. It used a simple algorithm: computers could transmit whenever the cable was idle. If 2 or more packets collided, each computer just waited a random time & tried later. We will call that version classic Ethernet for clarity.

Both wireless & wired broadcast networks can be divided into static & dynamic designs, depending on how the channel is allocated. A typical static allocation would be to divide time into discrete intervals & use a round-robin algorithm, allowing each machine to broadcast only when its time slot comes up. Static allocation wastes channel capacity when a machine has nothing to say during its allocated slot, so most systems attempt to allocate the channel dynamically (i.e., on demand).

Dynamic allocation methods for a common channel are either centralized or decentralized. In the centralized channel allocation method, there is a single entity, for example, the base station in cellular networks, which determines who goes next. It might do this by accepting multiple packets & prioritizing them according to some internal algorithm. In the decentralized channel allocation method, there is no central entity; each machine must decide for itself whether to transmit. You might think that this approach would lead to chaos, but it does not.

It is worth spending a little more time discussing LANs in the home. In the future, it is likely that every appliance in the home will be capable of communicating with every other appliance, & all of them will be accessible over the Internet. This development is likely to be one of those visionary concepts that nobody asked for (like TV remote controls or mobile phones), but once they arrived nobody can imagine how they lived without them.

Many devices are already capable of being networked. These include computers, entertainment devices such as TVs & DVDs, phones & other consumer electronics such as cameras, appliances like clock radios, & infrastructure like utility meters & thermostats. This trend will only continue. For instance, the average home probably has a dozen clocks (e.g., in appliances), all of which could adjust to daylight savings time automatically if the clocks were on the Internet. Remote monitoring of the home is a likely winner, as many grown children would be willing to spend some money to help their aging parents live safely in their own homes.

While we could think of the home network as just another LAN, it is more likely to have different properties than other networks. 1st, the networked devices must be very easy to install. Wireless routers are the most returned consumer electronic item. People buy 1 because they want a wireless network at home, find that it does not work "out of the box", & then return it rather than listen to elevator music while on hold on the technical helpline.

2nd, the network & devices must be foolproof in operation. Air conditioners used to have 1 knob with 4 settings: OFF, LOW, MEDIUM, & HIGH. Now they have 30-page manuals. Once they are networked, expect the chapter on security alone to be 30 pages. This is a problem because only computer users are accustomed to putting up with products that don't work; the car-, television-, & refrigerator-buying public is far less tolerant. They expect products to work 100% without the need to hire a geek.

3rd, low price is essential for success. People will not pay a $50 premium for an Internet thermostat because few people regard monitoring their home temperature from work that important. For $5 extra, though, it might sell.

4th, it must be possible to start out with 1 or 2 devices & expand the reach of the network gradually. This means no format wars. Telling consumers to buy peripherals with IEEE 1394 (FireWire) interfaces & a few years later retracting that & saying USB 2.0 is the interface-of-the-month & then switching that to 802.11g—oops, no, make that 802.11n—I mean 802.16 (different wireless networks)—is going to make consumers very skittish. The network interfaces will have to remain stable for decades, like the television broadcasting standards.

5th, security & reliability will be very important. Losing a few files to an email virus is 1 thing; having a burglar disarm your security system from his mobile computer & then plunder your house is something quite different.

An interesting question is whether home networks will be wired or wireless. Convenience & cost favors wireless networking because there are no wires to fit, or worse, retrofit. Security favors wired networking because the radio waves that wireless networks use are quite good at going through walls. Not everyone is overjoyed at the thought of having the neighbors piggybacking on their Internet connection & reading their email.

A 3rd option that may be appealing is to reuse the networks that are already in the home. The obvious candidate is the electric wires that are installed throughout the house. Power-line networks let devices that plug into outlets broadcast information throughout the house. You must plug in the TV anyway, & this way it can get Internet connectivity at the same time. The difficulty is how to carry both power & data signals at the same time. Part of the answer is that they use different frequency bands.

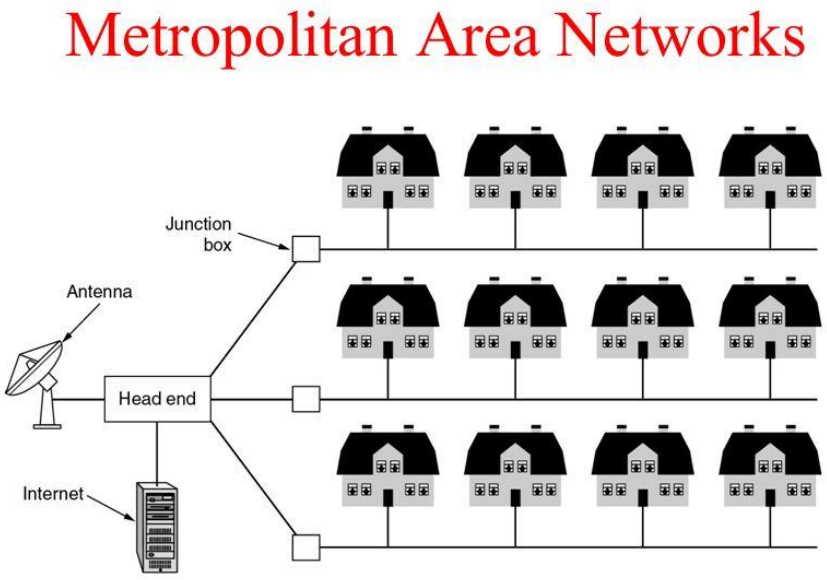
In short, home LANs offer many opportunities & challenges. Most of the latter relate to the need for the networks to be easy to manage, dependable, & secure, especially in the hands of nontechnical users, as well as low cost.

### 1.2.3 Metropolitan Area Networks

A MAN (Metropolitan Area Network) covers a city. The best-known examples of MANs are the cable television networks available in many cities. These systems grew from earlier community antenna systems used in areas with poor over-the-air television reception. In those early systems, a large antenna was placed on top of a nearby hill & a signal was then piped to the subscribers' houses.

At 1st, these were locally designed, ad hoc systems. Then companies began jumping into the business, getting contracts from local governments to wire up entire cities. The next step was television programming & even entire channels designed for cable only. Often these channels were highly specialized, such as all news, all sports, all cooking, all gardening, & so on. But from their inception until the late 1990s, they were intended for television reception only.

When the Internet began attracting a mass audience, the cable TV network operators began to realize that with some changes to the system, they could provide 2-way Internet service in unused parts of the spectrum. At that point, the cable TV system began to morph from simply a way to distribute television to a metropolitan area network. To a 1st approximation, a MAN might look something like the system shown in Fig. 1-9. In this figure we see both television signals & Internet being fed into the centralized cable headend for subsequent distribution to people's homes.




A metropolitan area network based on cable TV.

Cable television is not the only MAN, though. Recent developments in high-speed wireless Internet access have resulted in another MAN, which has been standardized as IEEE 802.16 & is popularly known as WiMAX.

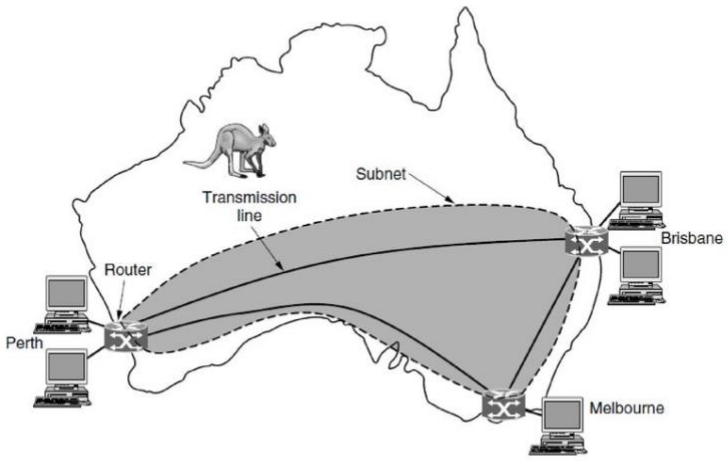
### 1.2.4 Wide Area Networks

A WAN (Wide Area Network) spans a large geographical area, often a country or continent. We will begin our discussion with wired WANs using the example of a company with branch offices in different cities.

The WAN in Fig. 1-10 is a network that connects offices in Perth, Melbourne, & Brisbane. Each of these offices contains computers intended

for running user (i.e., application) programs. We will follow traditional usage & call these machines hosts. The rest of the network that connects these hosts is then called the communication subnet, or just subnet for short. The job of the subnet is to carry messages from host to host, just as the telephone system carries words (just sounds) from speaker to listener.

## Wide Area Networks




WAN that connects three branch offices in Australia.

In most WANs, the subnet consists of 2 distinct components: transmission lines & switching elements. Transmission lines move bits between machines. They can be made of copper wire, optical fiber, or even radio links. Most companies do not have transmission lines lying about, so instead they lease the lines from a telecommunications company. Switching elements, or just switches, are specialized computers that connect 2 or more transmission lines. When data arrive on an incoming line, the switching element must choose an outgoing line on which to forward them. These switching computers have been called by various names in the past; the name router is now most used.

A short comment about the term ‘‘subnet’’ is in order here. Originally, its only meaning was the collection of routers & communication lines that moved packets from the source host to the destination host. Readers should be aware that it has acquired a 2nd, more recent meaning in conjunction with network addressing.

The WAN as we have described it looks similar to a large, wired LAN, but there are some important differences that go beyond long wires. Usually in a WAN, the hosts & subnet are owned & operated by different people. In our example, the employees might be responsible for their own computers, while the company's IT department is in charge of the rest of the network. We will see clearer boundaries in the coming examples, in which the network provider or telephone company operates the subnet. Separation of the pure communication aspects of the network (the subnet) from the application aspects (the hosts) greatly simplifies the overall network design.

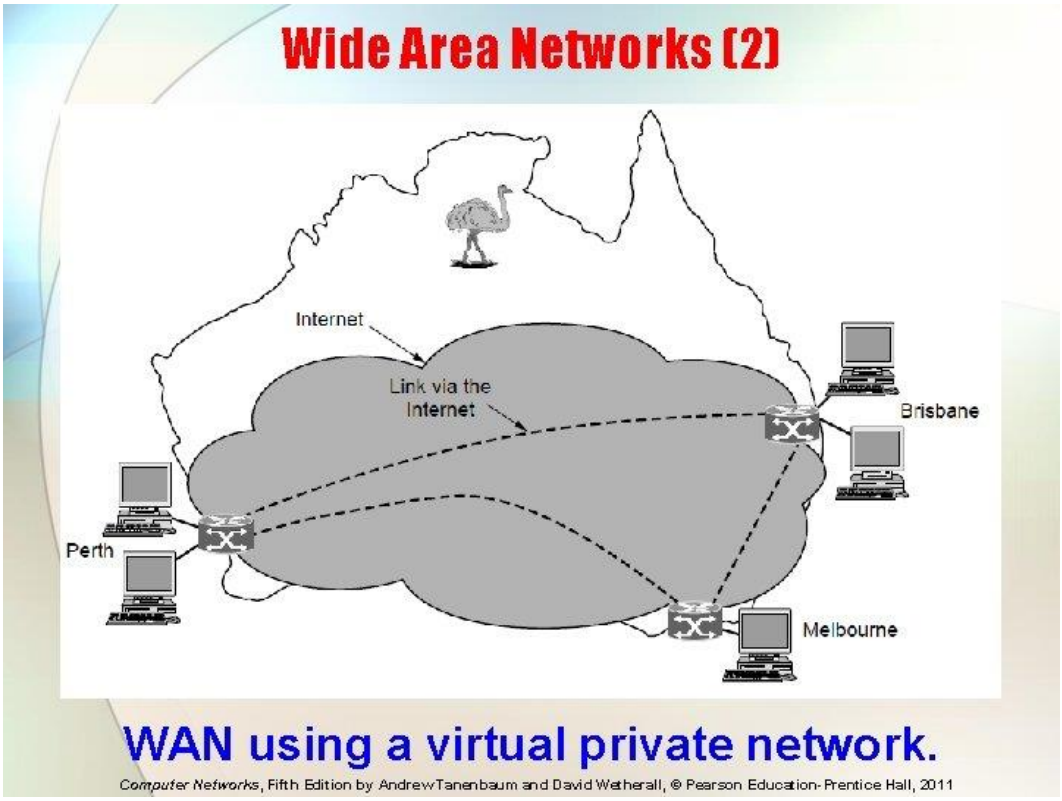
A 2nd difference is that the routers will usually connect different kinds of networking technology. The networks inside the offices may be switched Ethernet, for example, while the long-distance transmission lines may be SONET links. Some device needs to join them. This goes beyond our definition of a network. This means that many WANs will in fact be internetworks, or composite networks that are made up of more than 1 network.

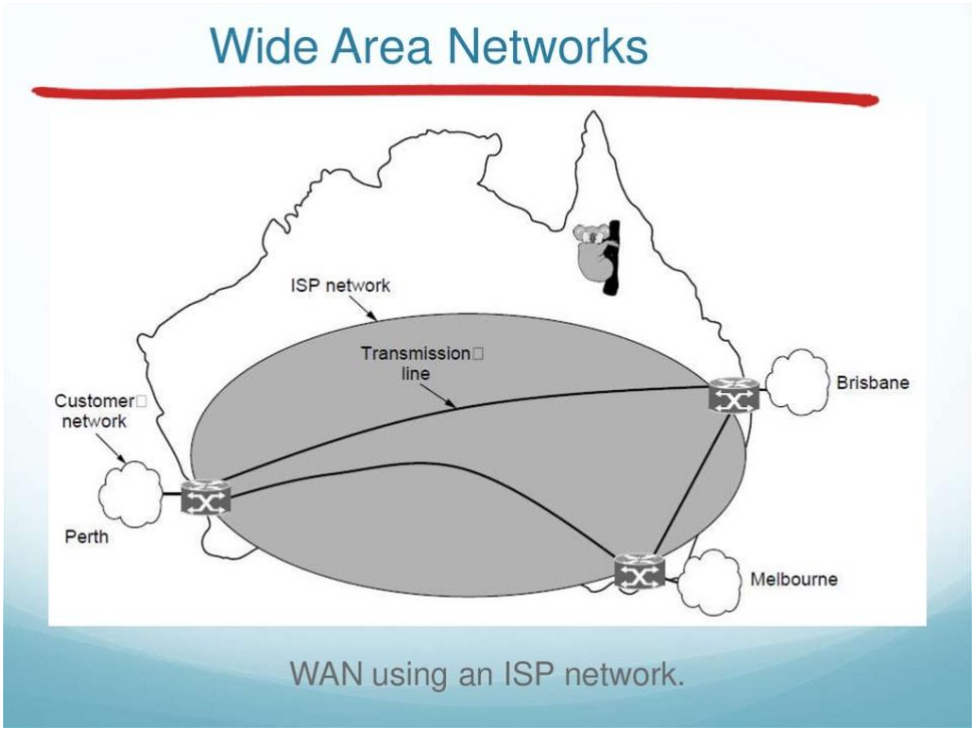
A final difference is in what is connected to the subnet. This could be individual computers, as was the case for connecting to LANs, or it could be entire LANs. This is how larger networks are built from smaller ones. As far as the subnet is concerned, it does the same job.

We are now able to look at 2 other varieties of WANs. 1st, rather than lease dedicated transmission lines, a company might connect its offices to the Internet. This allows connections to be made between the offices as virtual links that use the underlying capacity of the Internet. This arrangement, shown in Fig. 1-11, is called a VPN (Virtual Private Network). Compared to the dedicated arrangement, a VPN has the usual advantage of virtualization, which is that it provides flexible reuse of a resource (Internet connectivity). Consider how easy it is to add a 4th office to see this. A VPN also has the usual disadvantage of virtualization, which is a lack of control over the underlying resources. With a dedicated line, the capacity is clear. With a VPN your mileage may vary with your Internet service.

The 2nd variation is that the subnet may be run by a different company. The subnet operator is known as network service provider & the offices are its customers. This structure is shown in Fig. 1-12. The subnet operator will connect to other customers too, if they can pay & it can provide service.

Since it would be a disappointing network service if the customers could only send packets to each other, the subnet operator will also connect to other networks that are part of the Internet. Such a subnet operator is called an ISP (Internet Service Provider) & the subnet is an ISP network. Its customers who connect to the ISP receive Internet service.



In most WANs, the network contains many transmission lines, each connecting a pair of routers. If 2 routers that do not share a transmission line wish to communicate, they must do this indirectly, via other routers. There may be many paths in the network that connect these 2 routers. How the network makes the decision as to which path to use is called the routing algorithm. Many such algorithms exist. How each router makes the decision as to where to send a packet next is called the forwarding algorithm. Many of them exist too.

Other kinds of WANs make heavy use of wireless technologies. In satellite systems, each computer on the ground has an antenna through which it can send data to & receive data from to a satellite in orbit. All computers can hear the output *from* the satellite, & in some cases they can also hear the upward transmissions of their fellow computers to the satellite as well. Satellite networks are inherently broadcast & are most useful when the broadcast property is important.

The cellular telephone network is another example of a WAN that uses wireless technology. This system has already gone through 3 generations & a $4^{th}$ one is on the horizon. The $1^{st}$ generation was analog & for voice only. The $2^{nd}$ generation was digital & for voice only. The $3^{rd}$ generation is digital & is for both voice & data. Each cellular base station covers a distance much larger than a wireless LAN, with a range measured in kilometers rather than tens of meters. The base stations are connected to each other by a backbone network that is usually wired. The data rates of cellular networks are often on the order of 1 Mbps, much smaller than a wireless LAN that can range up to on the order of 100 Mbps.

### 1.2.5 Wireless Networks

Computer networks that are not connected by cables are called wireless networks. They generally use radio waves for communication between the network nodes. They allow devices to be connected to the network while roaming around within the network coverage.


Types of Wireless Networks include:

- Wireless LANs − Connects 2 or more network devices using wireless distribution techniques.
- Wireless MANs − Connects 2 or more wireless LANs spreading over a metropolitan area.
- Wireless WANs − Connects large areas comprising LANs, MANs & personal networks.

Advantages of Wireless Networks include:

- It provides clutter-free desks due to the absence of wires & cables.
- It increases the mobility of network devices connected to the system since the devices need not be connected to each other.
- Accessing network devices from any location within the network coverage or Wi-Fi hotspot becomes convenient since laying out cables is not needed.
- Installation & setup of wireless networks are easier.
- New devices can be easily connected to the existing setup since they needn't be wired to the present equipment. Also, the number of equipment that can be added or removed to the system can vary considerably since they are not limited by the cable capacity. This makes wireless networks very scalable.
- Wireless networks require very limited or no wires. Thus, it reduces the equipment & setup costs.

Examples of wireless networks include:

- Mobile phone networks
- Wireless sensor networks
- Satellite communication networks
- Terrestrial microwave networks

### 1.2.6 Home Networks

A home network is a small sized LAN that is used to connected devices within the small area of a home. It facilitates sharing of files, peripheral devices, programs & Internet access among the computers in a home. Home networks may be wired, i.e., connections within devices are done with cables; or wireless, i.e., connections are provided using Wi-Fi and Bluetooth.

A home network or home area network (HAN) is a type of computer network that facilitates communication among devices within the close vicinity of a home. Devices capable of participating in this network, for example, smart devices such as network printers & handheld mobile computers, often gain enhanced emergent capabilities through their ability to interact. These additional capabilities can be used to increase the quality of life inside the home in a variety of ways, such as automation of repetitive tasks, increased personal productivity, enhanced home security, & easier access to entertainment

Purpose of Home Networks include:

- Modem
- Router
- Network Switch
- Network Bridge
- Home Automation Controller


### 1.2.7 Internetworks

Many networks exist in the world, often with different hardware & software. People connected to one network often want to communicate with people attached to a different one. The fulfillment of this desire requires that different, & frequently incompatible, networks be connected. A collection of interconnected networks is called an internetwork or internet. These terms will be used in a generic sense, in contrast to the worldwide Internet (which is one specific internet), which we will always capitalize. The Internet uses ISP networks to connect enterprise networks, home networks, & many other networks.

Subnets, networks, & internetworks are often confused. The term “subnet” makes the most sense in the context of a wide area network, where it refers to the collection of routers & communication lines owned by the network operator. As an analogy, the telephone system consists of telephone switching offices connected to one another by high-speed lines, & to houses & businesses by low-speed lines. These lines & equipment, owned & managed by the telephone company, form the subnet of the telephone system. The telephones themselves (the hosts in this analogy) are not part of the subnet.

A network is formed by the combination of a subnet & its hosts. However, the word “network” is often used in a loose sense as well. A subnet might be described as a network, as in the case of the ‘‘ISP network’’ of Fig. 1-12. An internetwork might also be described as a network, as in the case of the WAN in Fig. 1-10. We will follow similar practice, & if we are distinguishing a network from other arrangements, we will stick with our original definition of a collection of computers interconnected by a single technology.

Let us say more about what constitutes an internetwork. We know that an internet is formed when distinct networks are interconnected. In our view, connecting a LAN & a WAN or connecting 2 LANs is the usual way to form an internetwork, but there is little agreement in the industry over terminology in this area. There are 2 rules of thumb that are useful. 1st, if different organizations have paid to construct different parts of the network & each maintains its part, we have an internetwork rather than a single network. 2nd, if the underlying technology is different in different parts (e.g., broadcast versus point-to-point & wired versus wireless), we probably have an internetwork.

To go deeper, we need to talk about how 2 different networks can be connected. The general name for a machine that makes a connection between 2 or more networks & provides the necessary translation, both in terms of hardware & software, is a gateway. Gateways are distinguished by the layer at which they operate in the protocol hierarchy. Higher layers are more tied to applications, such as the Web, & lower layers are more tied to transmission links, such as Ethernet.

Since the benefit of forming an internet is to connect computers across networks, we don’t want to use too low-level a gateway or we will be unable to make connections between different kinds of networks. We don’t want to use too high-level a gateway either, or the connection will only work for applications. The level in the middle that is “just right” is often called the network layer, & a router is a gateway that switches packets at the network layer. We can now spot an internet by finding a network that has routers.

## 1.3 Network Software

The 1st computer networks were designed with the hardware as the main concern & the software as an afterthought. This strategy no longer works. Network software is now highly structured.

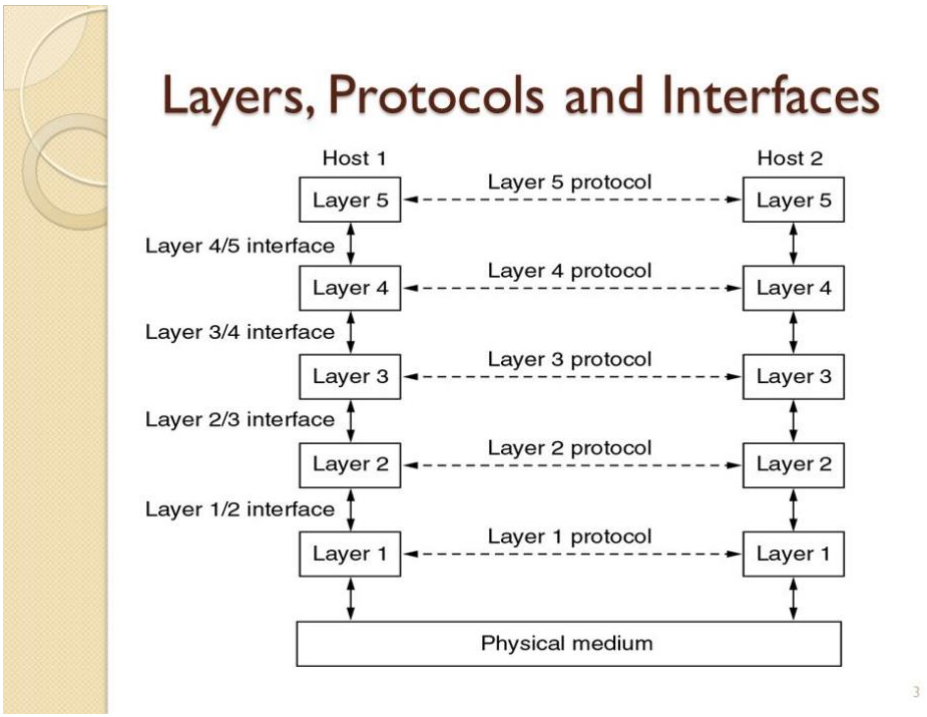
### 1.3.1 Protocol Hierarchies

To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, & the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers while shielding those layers from the details of how the offered services are implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

This concept is a familiar one & is used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, & object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state & algorithms hidden from them.

When layer *n* on one machine carries on a conversation with layer *n* on another machine, the rules & conventions used in this conversation are collectively known as the layer *n* protocol. Basically, a **protocol** is an agreement between the communicating parties on how communication is to proceed. As an analogy, when a woman is introduced to a man, she may choose to stick out her hand. He, in turn, may decide to either shake it or kiss it, depending, for example, on whether she is an American lawyer at a business meeting or a European princess at a formal ball. Violating the protocol will make communication more difficult, if not completely impossible.

A 5-layer network is illustrated in Fig. 1-13. The entities comprising the corresponding layers on different machines are called peers. The peers may be software processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol to talk to each other.




No data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data & control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. In Fig. 1-13, virtual communication is shown by dotted lines & physical communication by solid lines.

Between each pair of adjacent layers is an interface. The interface defines which primitive operations & services the lower layer makes available to the upper one. When network designers decide how many layers to include in a network & what each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer perform a specific collection of well-understood functions. In addition to minimizing the amount of information that must be passed between layers, clear-cut interfaces also make it simpler to replace one layer with a completely different protocol or implementation (e.g., replacing all the telephone lines by satellite channels) because all that is required of the new protocol or implementation is that it offers the same set of services to its upstairs neighbor as the old one did. It is common that different hosts use different implementations of the same

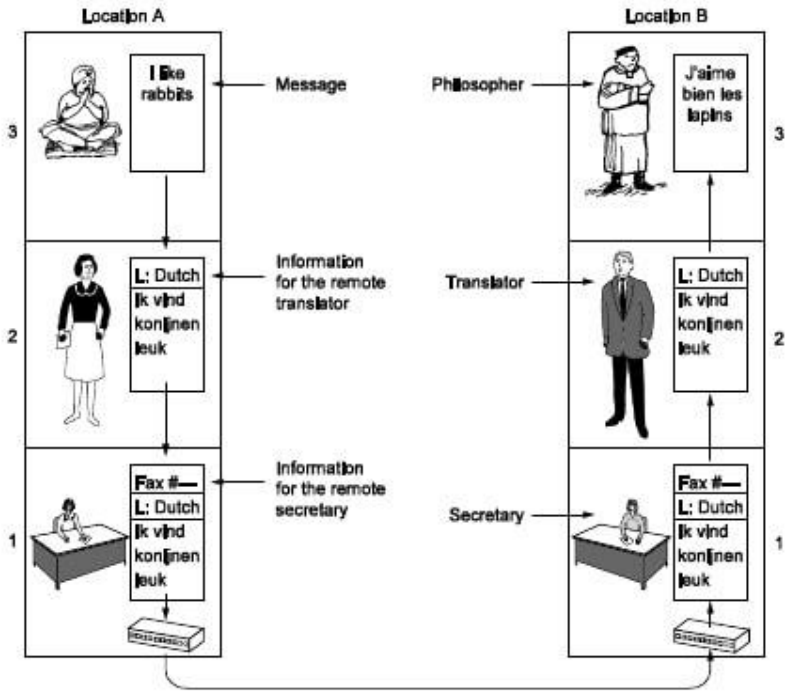
protocol (often written by different companies). In fact, the protocol itself can change in some layer without the layers above & below it even noticing.

A set of layers & protocols is called a network architecture. The specification of an architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Neither the details of the implementation nor the specification of the interfaces is part of the architecture because these are hidden away inside the machines & not visible from the outside. It is not even necessary that the interfaces on all machines in a network be the same, provided that each machine can correctly use all the protocols. A list of the protocols used by a certain system, one protocol per layer, is called a protocol stack.

An analogy may help explain the idea of multilayer communication. Imagine 2 philosophers (peer processes in layer 3), one of whom speaks Urdu & English & one of whom speaks Chinese & French. Since they have no common language, they each engage a translator (peer processes at layer 2), each of whom in turn contacts a secretary (peer processes in layer 1). Philosopher 1 wishes to convey his affection for Oryctolagus cuniculus to his peer. To do so, he passes a message (in English) across the 2/3 interface to his translator, saying "I like rabbits", as illustrated in Fig. 1-14. The translators have agreed on a neutral language known to both, Dutch, so the message is converted to "Ik vind konijnen leuk". The choice of the language is the layer 2 protocol & is up to the layer 2 peer processes.

The translator then gives the message to a secretary for transmission, for example, by email (the layer 1 protocol). When the message arrives at the other secretary, it is passed to the local translator, who translates it into French & passes it across the 2/3 interface to the second philosopher. Note that each protocol is completely independent of the other ones if the interfaces aren't changed. The translators can switch from Dutch to, say, Finnish, at will, if they both agree & neither changes his interface with either layer 1 or layer 3. Similarly, the secretaries can switch from email to telephone without disturbing (or even informing) the other layers. Each process may add some information intended only for its peer. This information is not passed up to the layer above.

# Protocol Hierarchies (2)

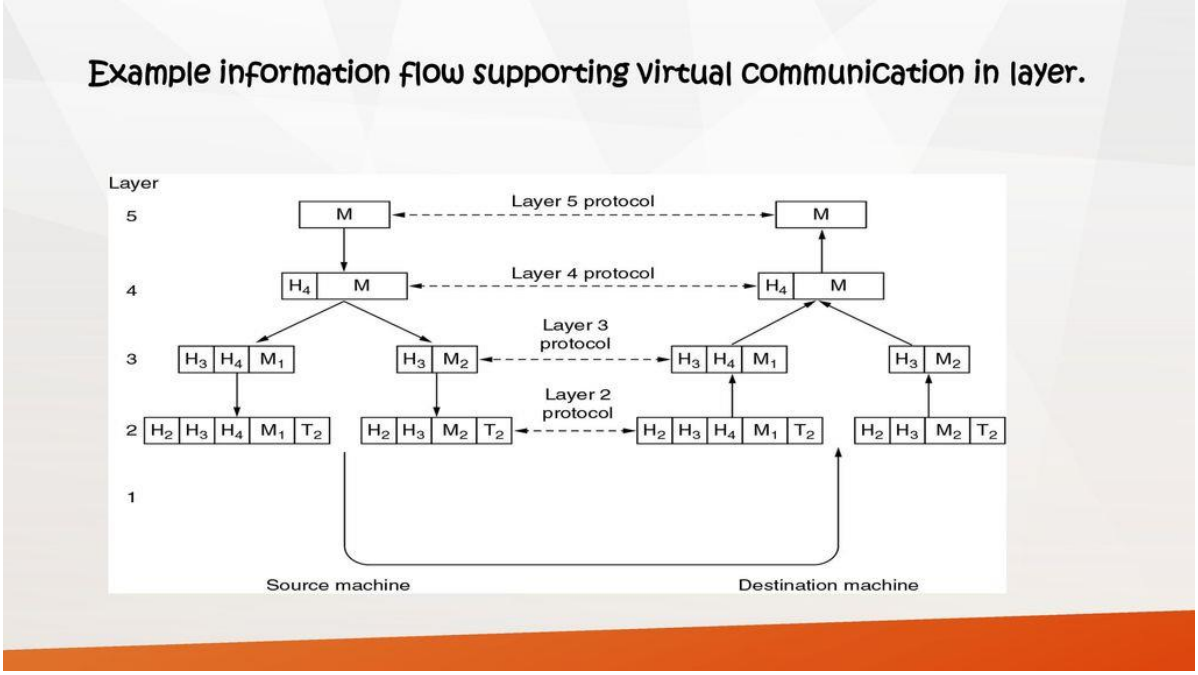



The philosopher-translator-secretary architecture.

Now consider a more technical example: how to provide communication to the top layer of the 5-layer network in Fig. 1-15. A message, M, is produced by an application process running in layer 5 & given to layer 4 for transmission. Layer 4 puts a header in front of the message to identify the message & passes the result to layer 3. The header includes control information, such as addresses, to allow layer 4 on the destination machine to deliver the message. Other examples of control information used in some layers are sequence numbers (in case the lower layer doesn't preserve message order), sizes, & times.

In many networks, no limit is placed on the size of messages transmitted in the layer 4 protocol but there is nearly always a limit imposed by the layer 3 protocol. Consequently, layer 3 must break up the incoming messages into smaller units, packets, prepending a layer 3 header to each packet. In this example, M is split into 2 parts, M1 & M2, that will be transmitted separately.

Layer 3 decides which of the outgoing lines to use & passes the packets to layer 2. Layer 2 adds to each piece not only a header but also a trailer & gives the resulting unit to layer 1 for physical transmission. At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses. None of the headers for layers below n are passed up to layer n.




The important thing to understand about Fig. 1-15 is the relation between the virtual & actual communication & the difference between protocols & interfaces. The peer processes in layer 4, for example, conceptually think of their communication as being ‘‘horizontal,’’ using the layer 4 protocol. Each one is likely to have procedures called something like *SendToOtherSide* and *GetFromOtherSide*, even though these procedures communicate with lower layers across the 3/4 interface, & not with the other side.

The peer process abstraction is crucial to all network design. Using it, the unmanageable task of designing the complete network can be broken into several smaller, manageable design problems, namely, the design of the individual layers.

The lower layers of a protocol hierarchy are frequently implemented in hardware or firmware. Nevertheless, complex protocol algorithms are involved, even if they are embedded (in whole or in part) in hardware.

### 1.3.2 Design Issues for the Layers

Some of the key design issues that occur in computer networks will come up in layer after layer. Below, we will briefly mention the more important ones.

Reliability is the design issue of making a network that operates correctly even though it is made up of a collection of components that are themselves unreliable. Think about the bits of a packet traveling through the network. There is a chance that some of these bits will be received damaged (inverted) due to fluke electrical noise, random wireless signals, hardware flaws, software bugs & so on. How is it possible that we find & fix these errors?

One mechanism for finding errors in received information uses codes for error detection. Information that is incorrectly received can then be retransmitted until it is received correctly. More powerful codes allow for error correction, where the correct message is recovered from the possibly incorrect bits that were originally received. Both mechanisms work by adding redundant information. They are used at low layers, to protect packets sent over individual links, & high layers, to check that the right contents were received.

Another reliability issue is finding a working path through a network. Often there are multiple paths between a source & destination, and in a large network, there may be some links or routers that are broken. Suppose that the network is down in Germany. Packets sent from London to Rome via Germany will not get through, but we could instead send packets from London to Rome via Paris. The network should automatically make this decision. This topic is called routing.

A 2$^{nd}$ design issue concerns the evolution of the network. Over time, networks grow larger & new designs emerge that need to be connected to the existing network. We have recently seen the key structuring mechanism used to support change by dividing the overall problem & hiding implementation details: **protocol layering**. There are many other strategies as well.

Since there are many computers on the network, every layer needs a mechanism for identifying the senders & receivers that are involved in a particular message. This mechanism is called addressing or naming, in the low and high layers, respectively.

An aspect of growth is that different network technologies often have different limitations. For example, not all communication channels preserve the order of messages sent on them, leading to solutions that number messages. Another example is differences in the maximum size of a message that the networks can transmit. This leads to mechanisms for disassembling, transmitting, & then reassembling messages. This overall topic is called internetworking.

When networks get large, new problems arise. Cities can have traffic jams, a shortage of telephone numbers, & it is easy to get lost. Not many people have these problems in their own neighborhood, but citywide they may be a big issue. Designs that continue to work well when the network gets large are said to be scalable.

A 3rd design issue is resource allocation. Networks provide a service to hosts from their underlying resources, such as the capacity of transmission lines. To do this well, they need mechanisms that divide their resources so that one host doesn't interfere with another too much.

Many designs share network bandwidth dynamically, according to the short-term needs of hosts, rather than by giving each host a fixed fraction of the bandwidth that it may or may not use. This design is called statistical multiplexing, meaning sharing based on the statistics of demand. It can be applied at low layers for a single link, or at high layers for a network or even applications that use the network.

An allocation problem that occurs at every level is how to keep a fast sender from swamping a slow receiver with data. Feedback from the receiver to the sender is often used. This subject is called flow control. Sometimes the problem is that the network is oversubscribed because too many computers want to send too much traffic, & the network can't deliver it all. This overloading of the network is called congestion. One strategy is for each computer to reduce its demand when it experiences congestion. It, too, can be used in all layers.

It is interesting to observe that the network has more resources to offer than simply bandwidth. For uses such as carrying live video, the timeliness of delivery matters a great deal. Most networks must provide service to applications that want this real-time delivery while they provide service to applications that want high throughput. Quality of service is the name given to mechanisms that reconcile these competing demands.

The last major design issue is to secure the network by defending it against different kinds of threats. One of the threats we have mentioned previously is that of eavesdropping on communications. Mechanisms that provide confidentiality defend against this threat, & they are used in multiple layers. Mechanisms for authentications prevent someone from impersonating someone else. They might be used to tell fake banking Web sites from the real one, or to let the cellular network check that a call is really coming from your phone so that you will pay the bill. Other mechanisms for integrity prevent surreptitious changes to messages, such as altering "debit my account $10" to "debit my account $1000". All these designs are based on cryptography.

### 1.3.3 Connection-Oriented & Connectionless Services

Layers can offer 2 different types of service to the layers above them: connection-oriented & connectionless.

Connection-oriented service is modeled after the telephone system. To talk to someone, you pick up the phone, dial the number, talk, & then hang up. Similarly, to use a connection-oriented network service, the service user 1st establishes a connection, uses the connection, & then releases the connection. The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at 1 end, & the receiver takes them out at the other end. In most cases the order is preserved so that the bits arrive in the order they were sent.

In some cases when a connection is established, the sender, receiver, & subnet conduct a negotiation about the parameters to be used, such as maximum message size, quality of service required, & other issues. Typically, 1 side makes a proposal & the other side can accept it, reject it, or make a counterproposal. A circuit is another name for a connection with associated resources, such as a fixed bandwidth. This dates from the

telephone network in which a circuit was a path over copper wire that carried a phone conversation.

In contrast to connection-oriented service, connectionless service is modeled after the postal system. Each message (letter) carries the full destination address, & each 1 is routed through the intermediate nodes inside the system independent of all the subsequent messages. There are different names for messages in different contexts; a packet is a message at the network layer. When the intermediate nodes receive a message in full before sending it on to the next node, this is called store-and-forward switching. The alternative, in which the onward transmission of a message at a node starts before it is completely received by the node, is called cut-through switching. Normally, when 2 messages are sent to the same destination, the 1st one sent will be the 1st one to arrive. However, it is possible that the 1st one sent can be delayed so that the 2nd one arrives 1st.

Each kind of service can further be characterized by its reliability. Some services are reliable in the sense that they never lose data. Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message, so the sender is sure that it arrived. The acknowledgement process introduces overhead & delays, which are often worth it but are sometimes undesirable.

A typical situation in which a reliable connection-oriented service is appropriate is file transfer. The owner of the file wants to be sure that all the bits arrive correctly & in the same order they were sent. Very few file transfer customers would prefer a service that occasionally scrambles or loses a few bits, even if it is much faster.

Reliable connection-oriented service has 2 minor variations: message sequences & byte streams. In the former variant, the message boundaries are preserved. When 2 1024-byte messages are sent, they arrive as 2 distinct 1024-byte messages, never as 1 2048-byte message. In the latter, the connection is simply a stream of bytes, with no message boundaries. When 2048 bytes arrive at the receiver, there is no way to tell if they were sent as 1 2048-byte message, 2 1024-byte messages, or 2048 1-byte messages. If the pages of a book are sent over a network to a phototypesetter as separate messages, it might be important to preserve the message boundaries. On the other hand, to download a DVD movie, a byte

stream from the server to the user's computer is all that is needed. Message boundaries within the movie are not relevant.

For some applications, the transit delays introduced by acknowledgements are unacceptable. One such application is digitized voice traffic for voice over IP. It is less disruptive for telephone users to hear a bit of noise on the line from time to time than to experience a delay waiting for acknowledgements. Similarly, when transmitting a video conference, having a few pixels wrong is no problem, but having the image jerk along as the flow stops and starts to correct errors is irritating.

Not all applications require connections. For example, spammers send electronic junk-mail to many recipients. The spammer probably doesn't want to go to the trouble of setting up & later tearing down a connection to a recipient just to send them one item. Nor is 100 percent reliable delivery essential, especially if it costs more. All that is needed is a way to send a single message that has a high probability of arrival, but no guarantee. Unreliable (meaning not acknowledged) connectionless service is often called datagram service, in analogy with telegram service, which also doesn't return an acknowledgement to the sender. Despite it being unreliable, it is the dominant form in most networks.

In other situations, the convenience of not having to establish a connection to send one message is desired, but reliability is essential. The acknowledged datagram service can be provided for these applications. It is like sending a registered letter & requesting a return receipt. When the receipt comes back, the sender is sure that the letter was delivered to the intended party & not lost along the way. Text messaging on mobile phones is an example.

Still another service is the request-reply service. In this service the sender transmits a single datagram containing a request; the reply contains the answer. Request-reply is commonly used to implement communication in the client-server model: the client issues a request & the server responds to it. For example, a mobile phone client might send a query to a map server to retrieve the map data for the current location. Figure 1-16 summarizes the types of services discussed above.

# Connection-Oriented and Connectionless Services

| | Service | Example |
|---|---|---|
| Connection-oriented | Reliable message stream | Sequence of pages |
| | Reliable byte stream | Remote login |
| | Unreliable connection | Digitized voice |
| Connection-less | Unreliable datagram | Electronic junk mail |
| | Acknowledged datagram | Registered mail |
| | Request-reply | Database query |

Six different types of service.

The concept of using unreliable communication may be confusing at 1st. After all, why would anyone prefer unreliable communication to reliable communication? First, reliable communication (in our sense, that is, acknowledged) may not be available in each layer. For example, Ethernet doesn't provide reliable communication. Packets can occasionally be damaged in transit. It is up to higher protocol levels to recover from this problem. Many reliable services are built on top of an unreliable datagram service. 2nd, the delays inherent in providing a reliable service may be unacceptable, especially in real-time applications such as multimedia. For these reasons, both reliable & unreliable communication coexist.

### 1.3.4 Service Primitives

A service is formally specified by a set of primitives (operations) available to user processes to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets.

The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connectionless service. As a minimal example of the service primitives that might provide a reliable byte stream, consider the primitives listed in Fig. 1-17. They will be familiar to fans of the Berkeley socket interface, as the primitives are a simplified version of that interface.

Service Primitives

Six service primitives that provide a simple connection-oriented service

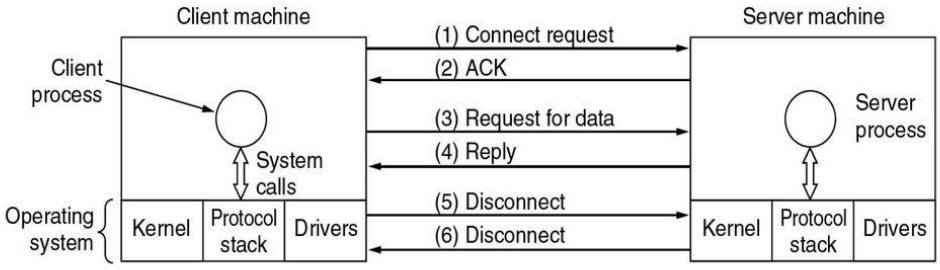
| Primitive | Meaning |
|---|---|
| LISTEN | Block waiting for an incoming connection |
| CONNECT | Establish a connection with a waiting peer |
| ACCEPT | Accept an incoming connection from a peer |
| RECEIVE | Block waiting for an incoming message |
| SEND | Send a message to the peer |
| DISCONNECT | Terminate a connection |

These primitives might be used for a request-reply to interaction in a client-server environment. To illustrate how, we sketch a simple protocol that implements the service using acknowledged datagrams.

1st, the server executes LISTEN to indicate that it is prepared to accept incoming connections. A common way to implement LISTEN is to make it a blocking system call. After executing the primitive, the server process is blocked until a request for connection appears.

Next, the client process executes CONNECT to establish a connection with the server. The CONNECT call needs to specify who to connect to, so it might have a parameter giving the server's address. The operating system then typically sends a packet to the peer asking it to connect, as shown by (1) in Fig. 1-18. The client process is suspended until there is a response.

# Service Primitives (2)




A simple client-server interaction using acknowledged datagrams.

When the packet arrives at the server, the operating system sees that the packet is requesting a connection. It checks to see if there is a listener, & if so, it unblocks the listener. The server process can then establish the connection with the ACCEPT call. This sends a response (2) back to the client process to accept the connection. The arrival of this response then releases the client. At this point the client & server are both running & they have a connection established.

The obvious analogy between this protocol & real life is a customer (client) calling a company's customer service manager. At the start of the day, the service manager sits next to his telephone in case it rings. Later, a client places a call. When the manager picks up the phone, the connection is established.

The next step is for the server to execute RECEIVE to prepare to accept the 1st request. Normally, the server does this immediately upon being released from the LISTEN, before the acknowledgement can get back to the client. The RECEIVE call blocks the server.

Then the client executes SEND to transmit its request (3) followed by the execution of RECEIVE to get the reply. The arrival of the request packet at the server machine unblocks the server so it can handle the request. After

it has done the work, the server uses SEND to return the answer to the client (4). The arrival of this packet unblocks the client, which can now inspect the answer. If the client has additional requests, it can make them now.

When the client is done, it executes DISCONNECT to terminate the connection (5). Usually, an initial DISCONNECT is a blocking call, suspending the client & sending a packet to the server saying that the connection is no longer needed. When the server gets the packet, it also issues a DISCONNECT of its own, acknowledging the client & releasing the connection (6). When the server's packet gets back to the client machine, the client process is released & the connection is broken. In a nutshell, this is how connection-oriented communication works.

Of course, life is not so simple. Many things can go wrong here. The timing can be wrong (e.g., the CONNECT is done before the LISTEN), packets can get lost, & much more. Fig. 1-18 briefly summarizes how client-server communication might work with acknowledged datagrams so that we can ignore lost packets.

Given that 6 packets are required to complete this protocol, one might wonder why a connectionless protocol is not used instead. The answer is that in a perfect world it could be, in which case only 2 packets would be needed: 1 for the request & 1 for the reply. However, in the face of large messages in either direction (e.g., a megabyte file), transmission errors, & lost packets, the situation changes. If the reply consisted of hundreds of packets, some of which could be lost during transmission, how would the client know if some pieces were missing? How would the client know whether the last packet received was really the last packet sent? Suppose the client wanted a second file. How could it tell packet 1 from the 2nd file from a lost packet 1 from the 1st file that suddenly found its way to the client? In short, in the real world, a simple request-reply protocol over an unreliable network is often inadequate. Having a reliable, ordered byte stream between processes is sometimes very convenient.

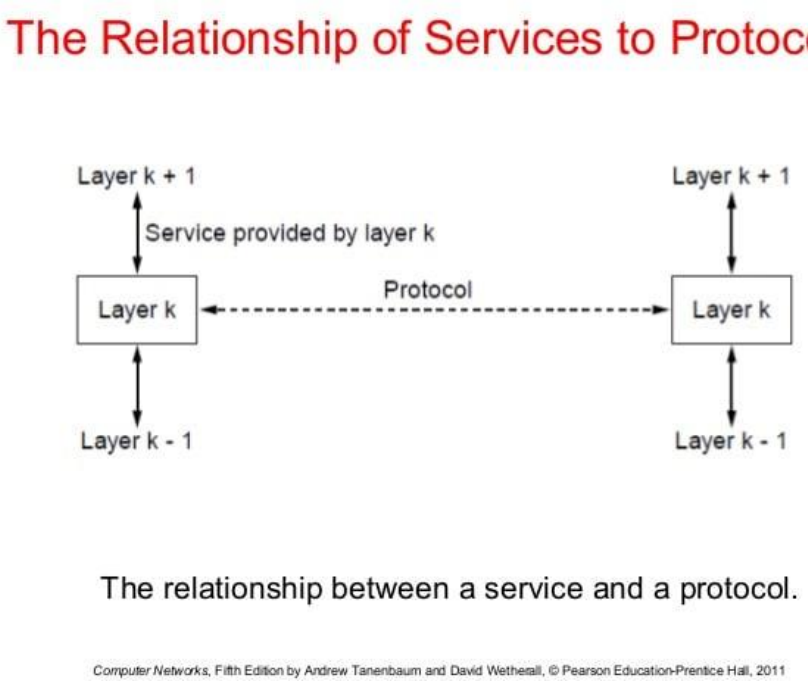
### 1.3.5 Relationship of Services to Protocols

Services & protocols are distinct concepts. This distinction is so important that we emphasize it again here. A service is a set of primitives (operations)

that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between 2 layers, with the lower layer being the service provider & the upper layer being the service user.

A *protocol*, in contrast, is a set of rules governing the format & meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols to implement their service definitions. They are free to change their protocols at will, provided they do not change the service visible to their users. In this way, the service & the protocol are completely decoupled. This is a key concept that any network designer should understand well.

To repeat this crucial point, services relate to the interfaces between layers, as illustrated in Fig. 1-19. In contrast, protocols relate to the packets sent between peer entities on different machines. It is very important not to confuse the 2 concepts.




The relationship between a service and a protocol.



An analogy with programming languages is worth making. A service is like an abstract data type or an object in an object-oriented language. It defines operations that can be performed on an object but doesn't specify how these operations are implemented. In contrast, a protocol relates to the

implementation of the service & as such is not visible to the user of the service.

Many older protocols didn't distinguish the service from the protocol. In effect, a typical layer might have had a service primitive SEND PACKET with the user providing a pointer to a fully assembled packet. This arrangement meant that all changes to the protocol were immediately visible to the users. Most network designers now regard such a design as a serious blunder.

## 1.4 Reference Models

We will discuss 2 important network architectures: the OSI reference model & the TCP/IP reference model. Although the protocols associated with the OSI model are not used any more, the model itself is quite general & still valid, & the features discussed at each layer are still very important. The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used. For this reason, we will look at both in detail. Also, sometimes you can learn more from failures than from successes.

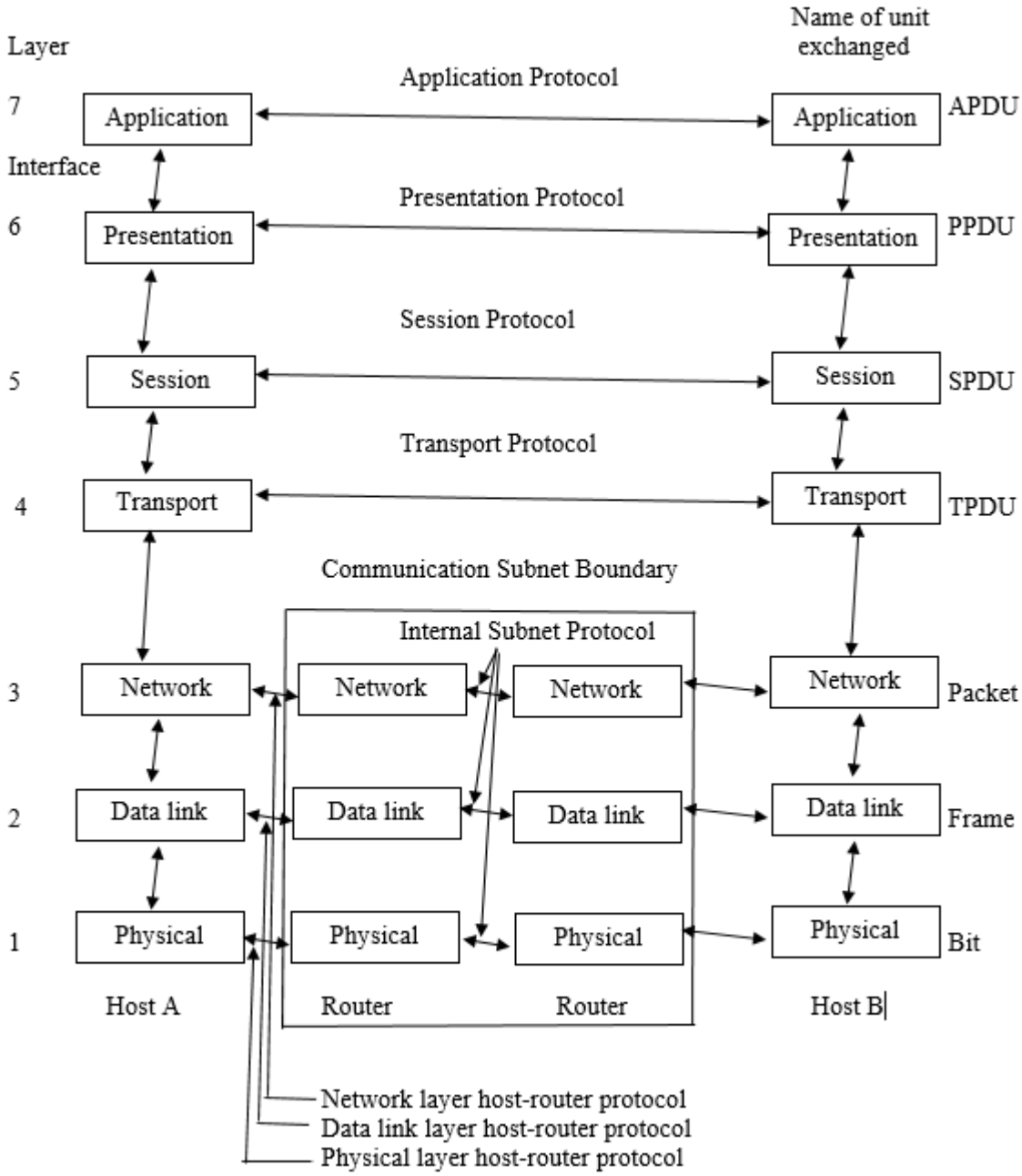
### 1.4.1 The OSI Reference Model

The OSI model (minus the physical medium) is shown in Fig. 1-20. This model is based on a proposal developed by the International Standards Organization (ISO) as a 1st step toward international standardization of the protocols used in the various layers. It was revised in 1995. The model is called the ISO OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will just call it the OSI model for short.

The OSI model has seven layers. The principles that were applied to arrive at the 7 layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.

4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity &small enough that the architecture does not become unwieldy.




Below we will discuss each layer of the model in turn, starting at the bottom layer. Note that the OSI model itself is not a network architecture because it doesn't specify the exact services & protocols to be used in each layer. It just tells what each layer should do. However, ISO has also produced

standards for all the layers, although these are not part of the reference model itself. Each one has been published as a separate international standard. The model (in part) is widely used although the associated protocols have been long forgotten.

**The Physical Layer**

The **physical layer** is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are what electrical signals should be used to represent a 1 and a 0, how many nanoseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established, how it is torn down when both sides are finished, how many pins the network connector has, & what each pin is used for. These design issues largely deal with mechanical, electrical, & timing interfaces, as well as the physical transmission medium, which lies below the physical layer.

**The Data Link Layer**

The main task of the **data link layer** is to transform a raw transmission facility into a line that appears free of undetected transmission errors. It does so by masking the real errors, so the network layer doesn't see them. It accomplishes this task by having the sender break up the input data into **data frames** (typically a few hundred or a few thousand bytes) & transmit the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an **acknowledgement frame**.

Another issue that arises in the data link layer (& most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism may be needed to let the transmitter know when the receiver can accept more data.

Broadcast networks have an additional issue in the data link layer: how to control access to the shared channel. A special sublayer of the data link layer, the **medium access control** sublayer, deals with this problem.

**The Network Layer**

The **network layer** controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into" the network & rarely changed, or more often they can be updated automatically to avoid failed components. They can also be determined at the start of each conversation, for example, a terminal session, such as a login to a remote machine. Finally, they can be highly dynamic, being determined anew for each packet to reflect the current network load.

If too many packets are present in the subnet at the same time, they will get in one another's way, forming bottlenecks. Handling congestion is also a responsibility of the network layer, in conjunction with higher layers that adapt the load they place on the network. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

When a packet must travel from 1 network to another to get to its destination, many problems can arise. The addressing used by the 2nd network may be different from that used by the 1st one. The 2nd one may not accept the packet at all because it is too large. The protocols may differ, & so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected.

In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

**The Transport Layer**

The basic function of the **transport layer** is to accept data from above it, split it up into smaller units, if need be, pass these to the network layer, & ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently & in a way that isolates the upper layers from the inevitable changes in the hardware technology over the course of time.

The transport layer also determines what type of service to provide to the session layer, &, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport services exist, such as the transporting of

isolated messages with no guarantee about the order of delivery, & the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established. (As an aside, an error-free channel is completely impossible to achieve what people really mean by this term is that the error rate is low enough to ignore in practice.)

The transport layer is a true end-to-end layer; it carries data all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers & control messages. In the lower layers, each protocol is between a machine & its immediate neighbors, & not between the ultimate source & destination machines, which may be separated by many routers. The difference between layers 1 through 3, which are chained, & layers 4 through 7, which are end-to-end, is illustrated in Fig. 1-20.

**The Session Layer**

The session layer allows users on different machines to establish **sessions** between them. Sessions offer various services, including **dialog control** (keeping track of whose turn it is to transmit), **token management** (preventing 2 parties from attempting the same critical operation simultaneously), & **synchronization** (checkpointing long transmissions to allow them to pick up from where they left off in the event of a crash & subsequent recovery).

**The Presentation Layer**

Unlike the lower layers, which are mostly concerned with moving bits around, the **presentation layer** is concerned with the syntax & semantics of the information transmitted. In order to make it possible for computers with different internal data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used “on the wire”. The presentation layer manages these abstract data structures & allows higher-level data structures (e.g., banking records) to be defined & exchanged.

**The Application Layer**

The **application layer** contains a variety of protocols that are commonly needed by users. One widely used application protocol is **HTTP (Hypertext Transfer Protocol),** which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server hosting the page using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, & network news.

### 1.4.2 The TCP/IP Reference Model

It is the reference model used in the grandparent of all wide area computer networks, the ARPANET, and its successor, the worldwide Internet. The ARPANET was a research network sponsored by the DoD (U.S. Department of Defense). It eventually connected hundreds of universities & government installations, using leased telephone lines. When satellite & radio networks were added later, the existing protocols had trouble interworking with them, so a new reference architecture was needed. Thus, from nearly the beginning, the ability to connect multiple networks in a seamless way was one of the major design goals. This architecture later became known as the TCP/IP Reference Model, after its 2 primary protocols. It was 1st described by Cerf & Kahn (1974), & later refined & defined as a standard in the Internet community (Braden, 1989). The design philosophy behind the model is discussed by Clark (1988).

Given the DoD's worry that some of its precious hosts, routers, & internetwork gateways might get blown to pieces at a moment's notice by an attack from the Soviet Union, another major goal was that the network be able to survive loss of subnet hardware, without existing conversations being broken off. In other words, the DoD wanted connections to remain intact if the source & destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation. Furthermore, since applications with divergent requirements were envisioned, ranging from transferring files to real-time speech transmission, a flexible architecture was needed.

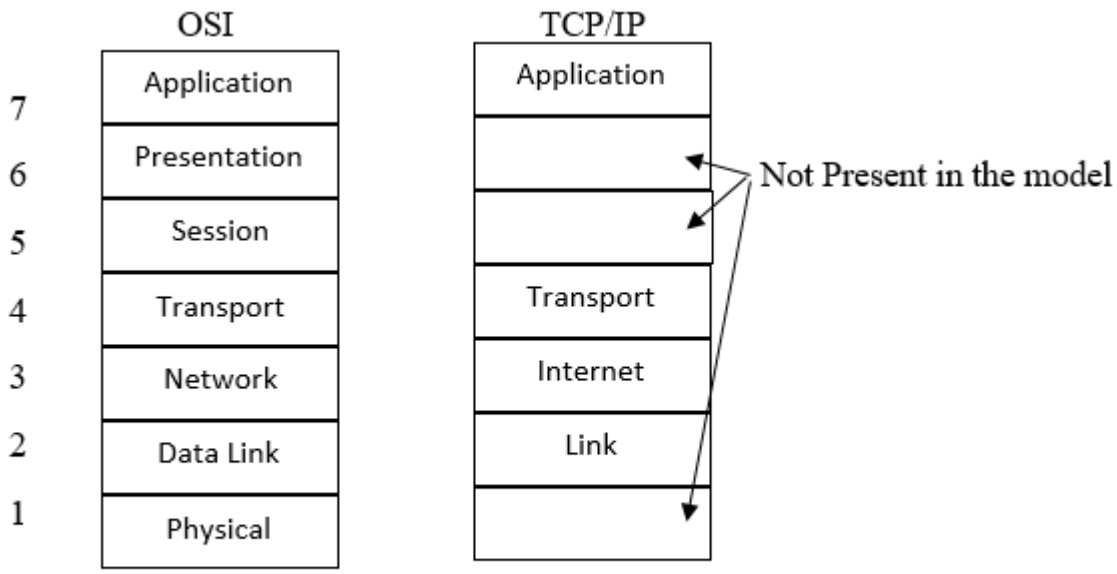
#### The Link Layer

All these requirements led to the choice of a packet-switching network based on a connectionless layer that runs across different networks. The lowest layer in the model, the **link layer** describes what links such as serial

lines & classic Ethernet must do to meet the needs of this connectionless internet layer. It is not really a layer at all, in the normal sense of the term, but rather an interface between hosts & transmission links. Early material on the TCP/IP model has little to say about it.

**The Internet Layer**

The internet layer is the linchpin that holds the whole architecture together. It is shown in Fig. 1-21 as corresponding roughly to the OSI network layer. Its job is to permit hosts to inject packets into any network & have them travel independently to the destination (potentially on a different network). They may even arrive in a completely different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. "Internet" is used here in a generic sense, even though this layer is present on the Internet.




The TCP Reference Model.

The analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mailbox in 1 country, & with a little luck, most of them will be delivered to the correct address in the destination country. The letters will probably travel through 1 or more international mail gateways along the way, but this is transparent to the users. Furthermore, that each country (i.e., each network) has its own stamps, preferred envelope sizes, & delivery rules is hidden from the users.

The internet layer defines an official packet format & protocol called IP (Internet Protocol), plus a companion protocol called ICMP (Internet Control Message Protocol) that helps it function. The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly a major issue here, as is congestion (though IP has not proven effective at avoiding congestion).

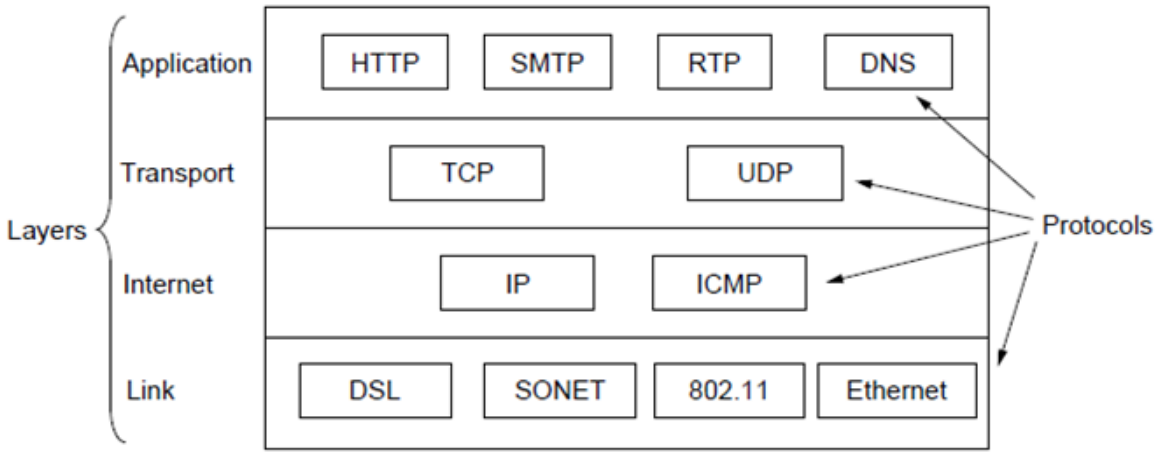
## The Transport Layer

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source & destination hosts to carry on a conversation, just as in the OSI transport layer. 2 end-to-end transport protocols have been defined here. The 1st one, TCP (Transmission Control Protocol), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine on the internet. It segments the incoming byte stream into discrete messages & passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

The 2nd protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control & wish to provide their own. It is also widely used for one-shot, client-server-type request-reply to queries & applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, & UDP is shown in Fig. 1-22. Since the model was developed, IP has been implemented on many other networks.

## The Application Layer

The TCP/IP model doesn't have session or presentation layers. No need for them was perceived. Instead, applications simply include any session & presentation functions that they require. Experience with the OSI model has proven this view correct: these layers are of little use to most applications.




On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), & electronic mail (SMTP). Many other protocols have been added to these over the years. Some important ones that we will study, shown in Fig. 1-22, include the Domain Name System (DNS), for mapping host names onto their network addresses, HTTP, the protocol for fetching pages on the World Wide Web, & RTP, the protocol for delivering real-time media such as voice or movies.

### 1.4.3 Comparison of OSI & TCP/IP Reference Models

The OSI & TCP/IP reference models have much in common. Both are based on the concept of a stack of independent protocols. Also, the functionality of the layers is roughly similar. For example, in both models the layers up through & including the transport layer are there to provide an end-to-end, network-independent transport service to processes wishing to communicate. These layers form the transport provider. Again, in both models, the layers above transport are application-oriented users of the transport service.

Despite these fundamental similarities, the 2 models also have many differences. It is important to note that we are comparing the reference models here, not the corresponding protocol stacks.

3 concepts are central to the OSI model:

1. Services.
2. Interfaces.
3. Protocols.

Probably the biggest contribution of the OSI model is that it makes the distinction between these 3 concepts explicit. Each layer performs some services for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics.

| OSI Model | TCP-IP Model |
|---|---|
| 7 layers | 4 layers |
| Model was initially defined before the implementation of the stack. | Model was defined after protocol stack was implemented. |
| OSI does not support internet working. | TCP-IP supports internet working. |
| Strict layered | Lossely layered |
| Support connectionless and connection oriented communication in the network layer. | Support only connection oriented communication in the transport layer. |
| Horizontal layer | Vertical approach |
| Separate session layer and presentation layer exist. | There are no session and presentation layers. Characteristics of session layer are provided by transport layer where as characteristics of presentation layer are provided by application layer. |

A layer's interface tells the processes above it how to access it. It specifies what the parameters are & what results to expect. It, too, says nothing about how the layer works inside.

Finally, the peer protocols used in a layer are the layer's own business. It can use any protocols it wants to, if it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers.

These ideas fit very nicely with modern ideas about object-oriented programming. An object, like a layer, has a set of methods (operations) that processes outside the object can invoke. The semantics of these methods define the set of services that the object offers. The methods' parameters & results form the object's interface. The code internal to the object is its protocol & is not visible or of any concern outside the object.

The TCP/IP model didn't originally clearly distinguish between services, interfaces, & protocols, although people have tried to retrofit it after the fact to make it more OSI-like. For example, the only real services offered by the internet layer are SEND IP PACKET & RECEIVE IP PACKET. Consequently, the protocols in the OSI model are better hidden than in the TCP/IP model & can be replaced relatively easily as the technology changes. Being able to make such changes transparently is one of the main purposes of having layered protocols in the 1st place.

The OSI reference model was devised before the corresponding protocols were invented. This ordering meant that the model was not biased toward one set of protocols, a fact that made it quite general. The downside of this ordering was that the designers didn't have much experience with the subject & didn't have a good idea of which functionality to put in which layer.

For example, the data link layer originally dealt only with point-to-point networks. When broadcast networks came around, a new sublayer had to be hacked into the model. Furthermore, when people started to build real networks using the OSI model & existing protocols, it was discovered that these networks didn't match the required service specifications (wonder of wonders), so convergence sublayers had to be grafted onto the model to provide a place for papering over the differences. Finally, the committee originally expected that each country would have one network, run by the government & using the OSI protocols, so no thought was given to internetworking. To make a long story short, things didn't turn out that way.

With TCP/IP the reverse was true: the protocols came 1st, & the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. They fit perfectly. The only trouble was that the model didn't fit any other protocol stacks. Consequently, it was not especially useful for describing other, non-TCP/IP networks.

Turning from philosophical matters to more specific ones, an obvious difference between the 2 models is the number of layers: the OSI model has 7 layers & the TCP/IP model has 4. Both have (inter)network, transport, & application layers, but the other layers are different.

Another difference is in the area of connectionless versus connection-oriented communication. The OSI model supports both connectionless & connection-oriented communication in the network layer, but only connection-oriented communication in the transport layer, where it counts (because the transport service is visible to the users). The TCP/IP model supports only 1 mode in the network layer (connectionless) but in the transport layer, giving the users a choice. This choice is especially important for simple request-response protocols.

## Physical Layer

We will look at the lowest layer in our protocol model, the physical layer. It defines the electrical, timing & other interfaces by which bits are sent as signals over channels. The physical layer is the foundation on which the network is built. The properties of different kinds of physical channels determine the performance (e.g., throughput, latency, & error rate) so it is a good place to start our journey into network land. We will begin with a theoretical analysis of data transmission, only to discover that Mother Nature puts some limits on what can be sent over a channel. Then we will cover 3 kinds of transmission media: guided (copper wire & fibre optics), wireless (terrestrial radio), & satellite. Each of these technologies has different properties that affect the design & performance of the networks that use them. This material will provide background information on the key transmission technologies used in modern networks. Next comes digital modulation, which is all about how analogue signals are converted into digital bits & back again. After that we will look at multiplexing schemes, exploring how multiple conversations can be put on the same transmission medium at the same time without interfering with one another. Finally, we will look at 3 examples of communication systems used in practice for wide area computer networks: the (fixed) telephone system, the mobile phone system, & the cable television system. Each of

these is important in practice, so we will devote a fair amount of space to each one.

Physical layer in the OSI model plays the role of interacting with actual hardware & signalling mechanism. Physical layer is the only layer of OSI network model which deals with the physical connectivity of 2 different stations. This layer defines the hardware equipment, cabling, wiring, frequencies, pulses used to represent binary signals etc.

Physical layer provides its services to Data-link layer. Data-link layer hands over frames to physical layer. Physical layer converts them to electrical pulses, which represent binary data. The binary data is then sent over the wired or wireless media.

When data is sent over physical medium, it needs to be 1st converted into electromagnetic signals. Data itself can be analogue such as human voice, or digital such as file on the disk. Both analogue & digital data can be represented in digital or analogue signals.

- Digital Signals
  Digital signals are discrete in nature and represent sequence of voltage pulses. Digital signals are used within the circuitry of a computer system.
- Analog Signals
  Analog signals are in continuous wave form in nature and represented by continuous electromagnetic waves.

When signals travel through the medium they tend to deteriorate (Transmission Impairment). This may have many reasons as given:

1. Attenuation
   For the receiver to interpret the data accurately, the signal must be sufficiently strong. When the signal passes through the medium, it tends to get weaker. As it covers distance, it loses strength.
2. Dispersion
   As signal travels through the media, it tends to spread and overlaps. The amount of dispersion depends upon the frequency used.
3. Delay distortion

Signals are sent over media with pre-defined speed and frequency. If the signal speed and frequency do not match, there are possibilities that signal reaches destination in arbitrary fashion. In digital media, this is very critical that some bits reach earlier than the previously sent ones.

4. Noise

Random disturbance or fluctuation in analogue or digital signal is said to be Noise in signal, which may distort the actual information being carried. Noise can be characterized in one of the following classes:

- Thermal Noise

Heat agitates the electronic conductors of a medium which may introduce noise in the media. Up to a certain level, thermal noise is unavoidable.

- Intermodulation

When multiple frequencies share a medium, their interference can cause noise in the medium. Intermodulation noise occurs if two different frequencies are sharing a medium and one of them has excessive strength or the component itself is not functioning properly, then the resultant frequency may not be delivered as expected.

- Crosstalk

This sort of noise happens when a foreign signal enters the media. This is because signal in one medium affects the signal of second medium.

- Impulse

This noise is introduced because of irregular disturbances such as lightening, electricity, short-circuit, or faulty components. Digital data is mostly affected by this sort of noise.

## 1.5 Modes of Communication

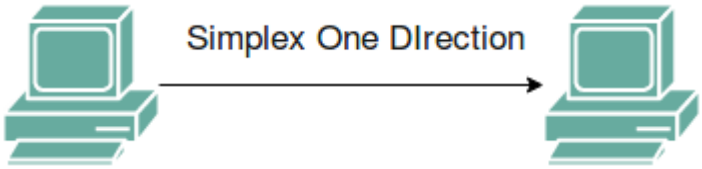
Transmission mode means transferring data between 2 devices. It is also known as a communication mode. Buses & networks are designed to allow communication to occur between individual devices that are interconnected. There are 3 types of transmission mode.

These are explained as following below.

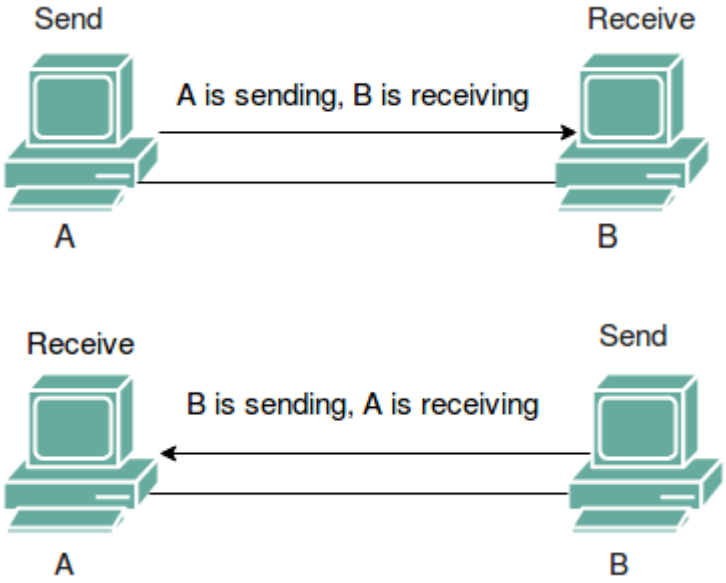
- **Simplex Mode**

  In Simplex mode, the communication is unidirectional, as on a one-way street. Only one of the 2 devices on a link can transmit, the other

can only receive. The simplex mode can use the entire capacity of the channel to send data in 1 direction.
Example: Keyboard & traditional monitors. The keyboard can only introduce input, the monitor can only give the output.




- **Half-Duplex Mode**
  In half-duplex mode, each station can both transmit & receive, but not at the same time. When one device is sending, the other can only receive, & vice versa. The half-duplex mode is used in cases where there is no need for communication in both directions at the same time. The entire capacity of the channel can be utilized for each direction.



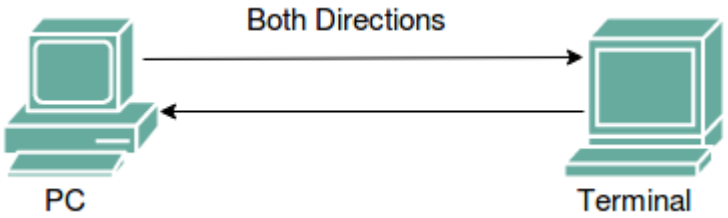

Example: Walkie-talkie in which message is sent one at a time & messages are sent in both directions.
Channel capacity=Bandwidth * Propagation Delay

- **Full-Duplex Mode**
  In full-duplex mode, both stations can transmit & receive simultaneously. In full-duplex mode, signals going in 1 direction

share the capacity of the link with signals going in another direction, this sharing can occur in 2 ways:

- The link must contain 2 physically separate transmission paths, 1 for sending & the other for receiving.
- Or the capacity is divided between signals travelling in both directions.

Full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the 2 directions.




Example: Telephone Network in which there is communication between 2 persons by a telephone line, through which both can talk & listen at the same time.

Channel Capacity=2* Bandwidth*propagation Delay

## 1.6 Physical Topologies

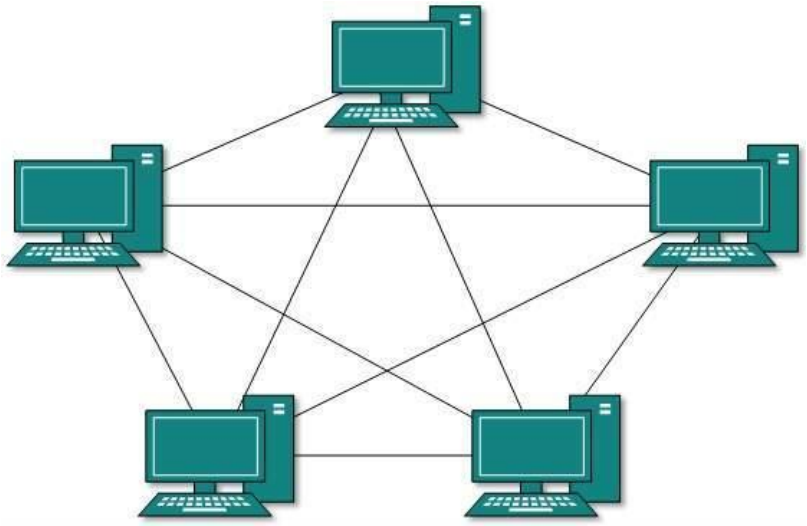
Topology defines the structure of the network of how all the components are interconnected to each other. There are 2 types of topologies: physical & logical topology. **Physical topology** is the geometric representation of all the nodes in a network.

A Network Topology is the arrangement with which computer systems or network devices are connected to each other. Topologies may define both physical & logical aspect of the network. Both logical & physical topologies could be same or different in a same network.

### 1.6.1 Mesh Topology

In this type of topology, a host is connected to one or multiple hosts. This topology has hosts in point-to-point connection with every other host or

may also have hosts which are in point-to-point connection to few hosts only.



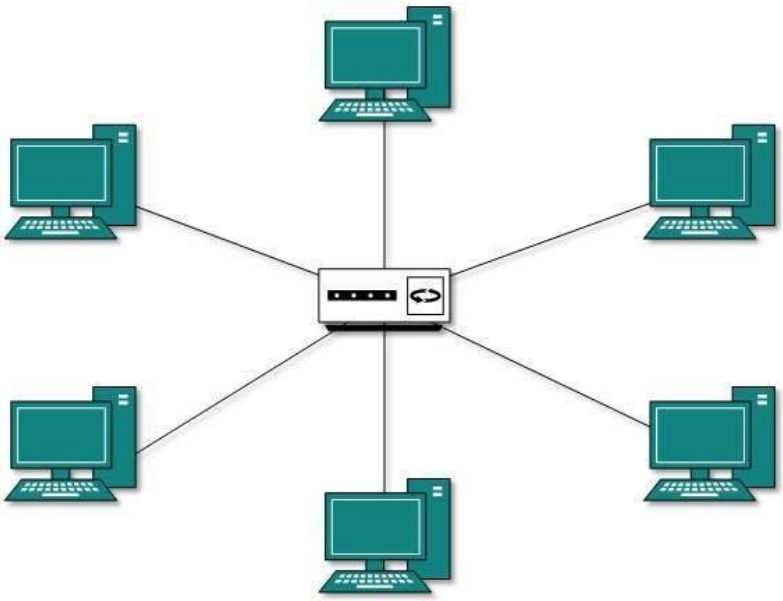
Hosts in mesh topology also work as relay for other hosts which don't have direct point-to-point links. Mesh technology comes into 2 types:

- Full Mesh: All hosts have a point-to-point connection to every other host in the network. Thus, for every new host n(n-1)/2 connections are required. It provides the most reliable network structure among all network topologies.
- Partially Mesh: Not all hosts have point-to-point connection to every other host. Hosts connect to each other in some arbitrarily fashion. This topology exists where we need to provide reliability to some hosts out of all

### 1.6.2 Star Topology

All hosts in star topology are connected to a central device, known as hub device, using a point-to-point connection. That is, there exists a point-to-point connection between hosts & hub. The hub device can be any of the following:

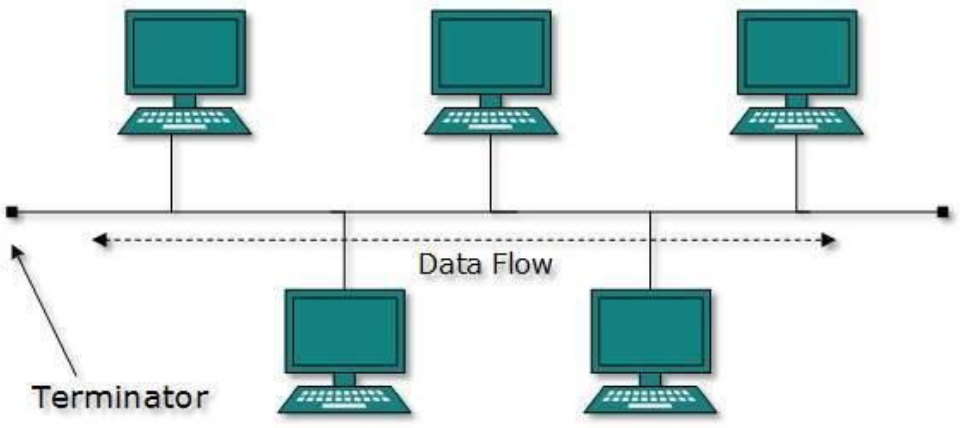
- Layer-1 device such as hub or repeater
- Layer-2 device such as switch or bridge
- Layer-3 device such as router or gateway



As in bus topology, hub acts as single point of failure. If hub fails, connectivity of all hosts to all other hosts fails. Every communication between hosts, takes place through only the hub. Star topology is not expensive as to connect one more host, only one cable is required & configuration is simple.

### 1.6.3 Bus Topology

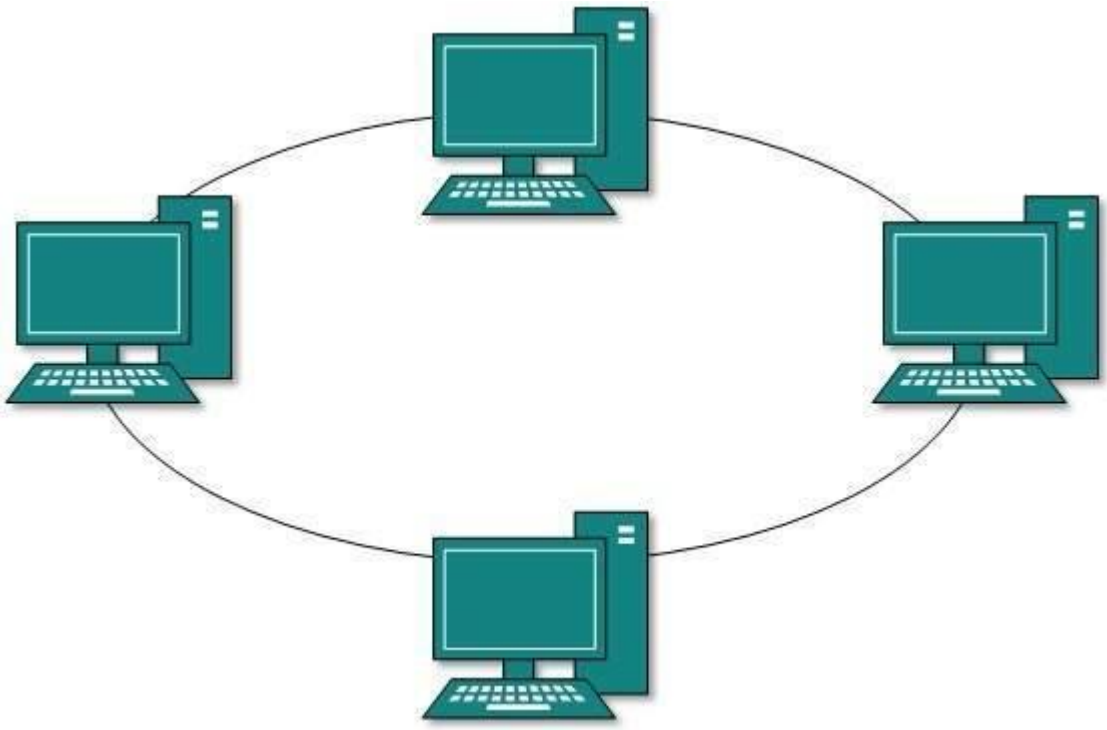
In case of Bus topology, all devices share single communication line or cable. Bus topology may have problem while multiple hosts sending data at the same time. Therefore, Bus topology either uses CSMA/CD technology or recognizes one host as Bus Master to solve the issue. It is one of the simple forms of networking where a failure of a device doesn't affect the other devices. But failure of the shared communication line can make all other devices stop functioning.



Both ends of the shared channel have line terminator. The data is sent in only 1 direction & as soon as it reaches the extreme end, the terminator removes the data from the line.

### 1.6.4 Ring Topology

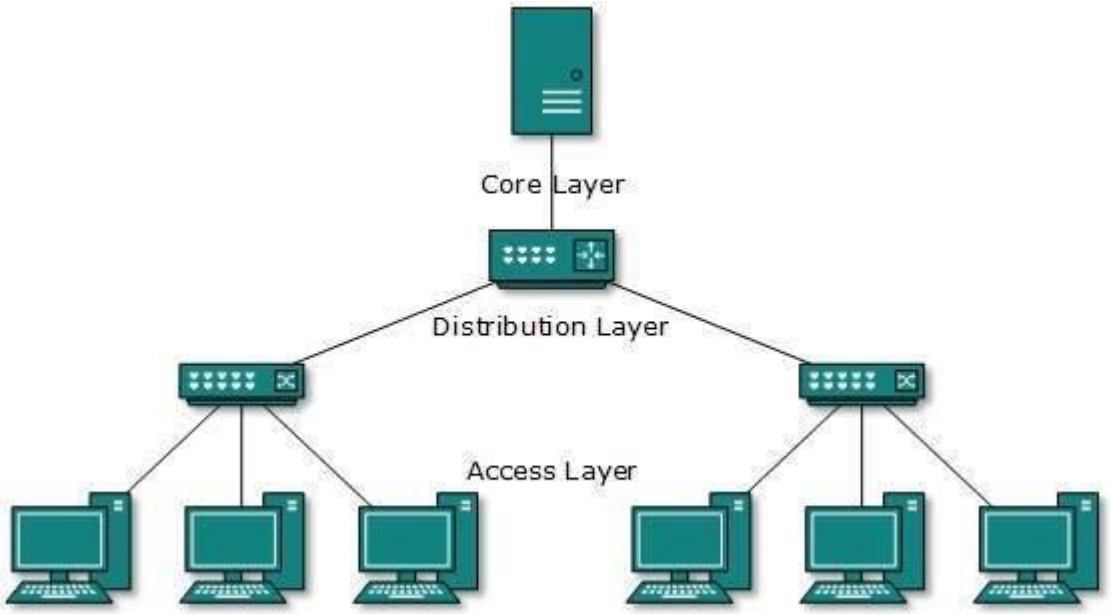
In ring topology, each host machine connects to exactly 2 other machines, creating a circular network structure. When 1 host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts. To connect 1 more host in the existing structure, the administrator may need only 1 more extra cable.



Failure of any host results in failure of the whole ring. Thus, every connection in the ring is a point of failure. There are methods which employ one more backup ring.

### 1.6.5 Tree Topology (not in the syllabus)

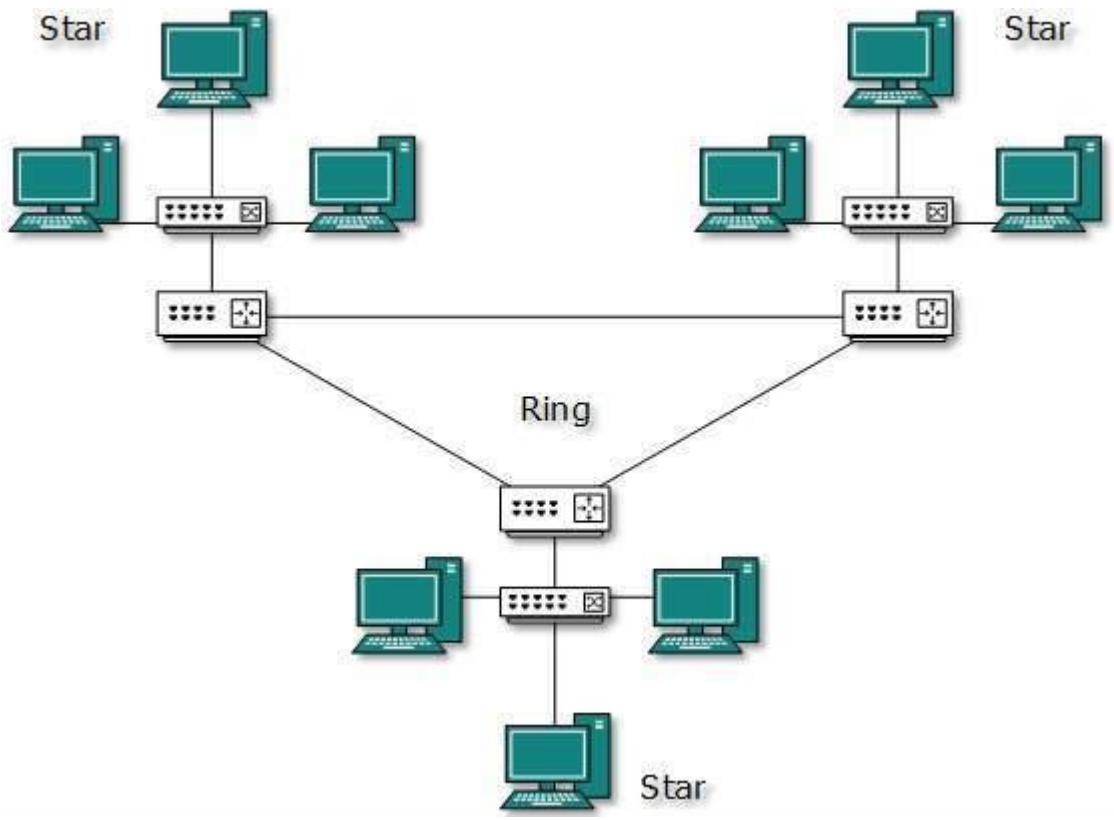
Also known as hierarchical topology, this is the most common form of network topology in use presently. This topology imitates as extended star topology & inherits properties of bus topology.




This topology divides the network into multiple levels/layers of network. Mainly in LANs, a network is bifurcated into 3 types of network devices. The lowermost is access-layer where computers are attached. The middle layer is known as distribution layer, which works as mediator between upper layer & lower layer. The highest layer is known as core layer, & is central point of the network, i.e., root of the tree from which all nodes fork. All neighbouring hosts have point-to-point connection between them. Like the bus topology, if the root goes down, then the entire network suffers even though it is not the single point of failure. Every connection serves as point of failure, failing of which divides the network into unreachable segment.

### 1.6.6 Hybrid Topology

A network structure whose design contains more than 1 topology is said to be hybrid topology. Hybrid topology inherits merits & demerits of all the incorporating topologies.




The above picture represents an arbitrarily hybrid topology. The combining topologies may contain attributes of Star, Ring, & Bus topologies. Most WANs are connected by means of Dual-Ring topology & networks connected to them are mostly star topology networks. Internet is the best example of largest Hybrid topology.

## 1.7 Signal Encoding

**Encoding** is the process of converting the data or a given sequence of characters, symbols, alphabets etc., into a specified format, for the secured transmission of data. **Decoding** is the reverse process of encoding which is to extract the information from the converted format.

Data Encoding is the process of using various patterns of voltage or current levels to represent **1s** & **0s** of the digital signals on the transmission link. The common types of line encoding are Unipolar, Polar, Bipolar, & Manchester.

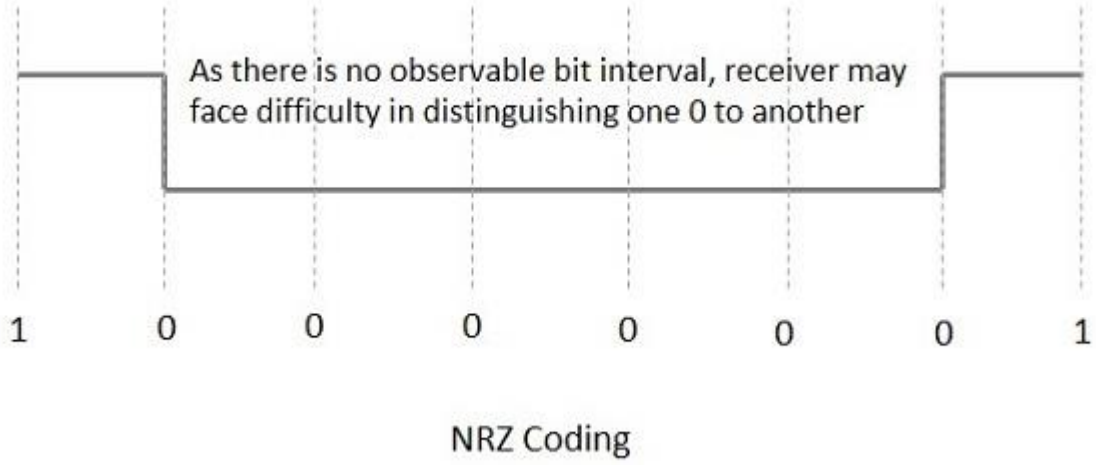
The data encoding technique is divided into the following types, depending upon the type of data conversion.

- **Analog data to Analog signals** − The modulation techniques such as Amplitude Modulation, Frequency Modulation & Phase Modulation of analogue signals, fall under this category.
- **Analog data to Digital signals** − This process can be termed as digitization, which is done by Pulse Code Modulation PCM. Hence, it is nothing but digital modulation. Sampling & quantization are the important factors in this. Delta Modulation gives a better output than PCM.
- **Digital data to Analog signals** − The modulation techniques such as Amplitude Shift Keying ASK, Frequency Shift Keying FSK, Phase Shift Keying PSK, etc., fall under this category.
- **Digital data to Digital signals** − There are several ways to map digital data to digital signals. Some of them are:

**Non-Return to Zero NRZ**

NRZ Codes has **1** for High voltage level & **0** for Low voltage level. The main behaviour of NRZ codes is that the voltage level remains constant during bit interval. The end or start of a bit will not be indicated & it will maintain the same voltage state, if the value of the previous bit & the value of the present bit are same.

The following figure explains the concept of NRZ coding.




NRZ Coding

If the above example is considered, as there is a long sequence of constant voltage level & the clock synchronization may be lost due to the absence

of bit interval, it becomes difficult for the receiver to differentiate between 0 & 1.

There are 2 variations in NRZ namely −

**NRZ - L NRZ–LEVEL**

There is a change in the polarity of the signal, only when the incoming signal changes from 1 to 0 or from 0 to 1. It is the same as NRZ, however, the 1st bit of the input signal should have a change of polarity.

**NRZ - I NRZ–INVERTED**

If a **1** occurs at the incoming signal, then there occurs a transition at the beginning of the bit interval. For a **0** at the incoming signal, there is no transition at the beginning of the bit interval.

NRZ codes has a **disadvantage** that the synchronization of the transmitter clock with the receiver clock gets completely disturbed, when there is a string of **1s** & **0s**. Hence, a separate clock line needs to be provided.

## Bi-phase Encoding

The signal level is checked twice for every bit time, both initially & in the middle. Hence, the clock rate is double the data transfer rate & thus the modulation rate is also doubled. The clock is taken from the signal itself. The bandwidth required for this coding is greater.

There are 2 types of Bi-phase Encoding.

- Bi-phase Manchester
- Differential Manchester

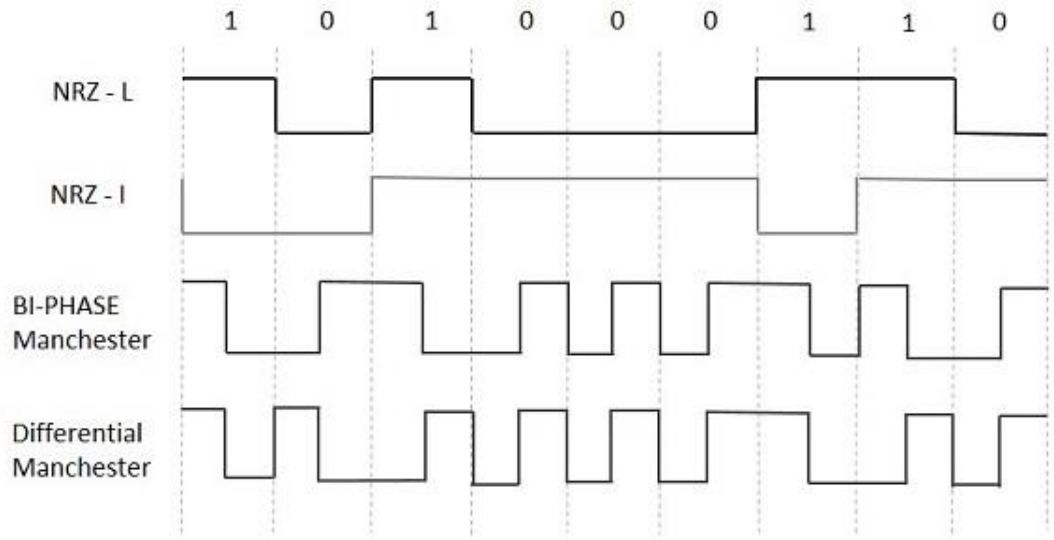
**Bi-phase Manchester** (important)

In this type of coding, the transition is done at the middle of the bit interval. The transition for the resultant pulse is from High to Low

in the middle of the interval, for the input bit 1. While the transition is from Low to High for the input bit **0**.

**Differential Manchester** (important)

In this type of coding, there always occurs a transition in the middle of the bit interval. If there occurs a transition at the beginning of the bit interval, then the input bit is **0**. If no transition occurs at the beginning of the bit interval, then the input bit is **1**.
The following figure illustrates the waveforms of NRZ-L, NRZ-I, Bi-phase Manchester and Differential Manchester coding for different digital inputs.




## Block Coding

Among the types of block coding, the famous ones are 4B/5B encoding & 8B/6T encoding. The number of bits is processed in different manners, in both of these processes.

### 4B/5B Encoding

In Manchester encoding, to send the data, the clock with double speed is required rather than NRZ coding. Here, as the name implies, 4 bits of code is mapped with 5 bits, with a minimum number of **1** bits in the group. The clock synchronization problem in NRZ-I encoding is avoided by assigning an equivalent word of 5 bits in the place of each block of 4 consecutive bits. These 5-bit words are predetermined in a dictionary. The basic idea of selecting a 5-bit code is that it should have **one leading 0** & it should have **no more than two trailing 0s**. Hence, these words are chosen such that 2 transactions take place per block of bits.

**8B/6T Encoding**

We have used 2 voltage levels to send a single bit over a single signal. But if we use more than 3 voltage levels, we can send more bits per signal. For example, if 6 voltage levels are used to represent 8 bits on a single signal, then such encoding is termed as 8B/6T encoding. Hence in this method, we have as many as 729 3636 combinations for signal & 256 2828 combinations for bits.

These are the techniques mostly used for converting digital data into digital signals by compressing or coding them for reliable transmission of data.

## 1.8 Transmission Media Overview

The purpose of the physical layer is to transport bits from 1 machine to another. Various physical media can be used for the actual transmission. Each one has its own niche in terms of bandwidth, delay, cost, & ease of installation & maintenance. Media are roughly grouped into guided media, such as copper wire & fibre optics, & unguided media, such as terrestrial wireless, satellite, & lasers through the air.

### 1.8.1 Guided Media

**Magnetic Media** (not in the syllabus)

One of the most common ways to transport data from 1 computer to another is to write them onto magnetic tape or removable media (e.g., recordable

DVDs), physically transport the tape or disks to the destination machine & read them back in again. Although this method is not as sophisticated as using a geosynchronous communication satellite, it is often more cost effective, especially for applications in which high bandwidth or cost per bit transported is the key factor.

A simple calculation will make this point clear. An industry-standard Ultrium tape can hold 800 gigabytes. A box 60 × 60 × 60 cm can hold about 1000 of these tapes, for a total capacity of 800 terabytes, or 6400 terabits (6.4 petabits). A box of tapes can be delivered anywhere in the United States in 24 hours by Federal Express and other companies. The effective bandwidth of this transmission is 6400 terabits/86,400 sec, or a bit over 70 Gbps. If the destination is only an hour away by road, the bandwidth is increased to over 1700 Gbps. No computer network can even approach this. Of course, networks are getting faster, but tape densities are increasing, too.

If we now look at cost, we get a similar picture. The cost of an Ultrium tape is around $40 when bought in bulk. A tape can be reused at least 10 times, so the tape cost is maybe $4000 per box per usage. Add to this another $1000 for shipping (probably much less), and we have a cost of roughly $5000 to ship 800 TB. This amounts to shipping a gigabyte for a little over half a cent. No network can beat that. The moral of the story is:
*Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.*

## Twisted Pairs

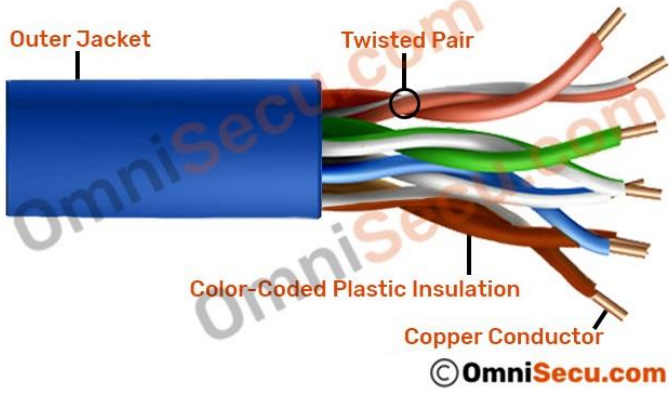
Although the bandwidth characteristics of magnetic tape are excellent, the delay characteristics are poor. Transmission time is measured in minutes or hours, not milliseconds. For many applications an online connection is needed. One of the oldest and still most common transmission media is twisted pair. A twisted pair consists of two insulated copper wires, typically about 1 mm thick. The wires are twisted together in a helical form, just like a DNA molecule. Twisting is done because two parallel wires constitute a fine antenna. When the wires are twisted, the waves from different twists cancel out, so the wire radiates less effectively. A signal is

usually carried as the difference in voltage between the two wires in the pair. This provides better immunity to external noise because the noise tends to affect both wires the same, leaving the differential unchanged.

The most common application of the twisted pair is the telephone system. Nearly all telephones are connected to the telephone company (telco) office by a twisted pair. Both telephone calls and ADSL Internet access run over these lines. Twisted pairs can run several kilometres without amplification, but for longer distances the signal becomes too attenuated and repeaters are needed. When many twisted pairs run in parallel for a substantial distance, such as all the wires coming from an apartment building to the telephone company office, they are bundled together and encased in a protective sheath. The pairs in these bundles would interfere with one another if it were not for the twisting. In parts of the world where telephone lines run on poles above ground, it is common to see bundles several centimetres in diameter.

Twisted pairs can be used for transmitting either analogue or digital information. The bandwidth depends on the thickness of the wire and the distance travelled, but several megabits/sec can be achieved for a few kilometres in many cases. Due to their adequate performance and low cost, twisted pairs are widely used and are likely to remain so for years to come.

Twisted-pair cabling comes in several varieties. The garden variety deployed in many office buildings is called Category 5 cabling, or ‘‘Cat 5.’’ A category 5 twisted pair consists of two insulated wires gently twisted together. Four such pairs are typically grouped in a plastic sheath to protect the wires and keep them together. This arrangement is shown in Fig. 2-3.



Different LAN standards may use the twisted pairs differently. For example, 100-Mbps Ethernet uses two (out of the four) pairs, one pair for each direction. To reach higher speeds, 1-Gbps Ethernet uses all four pairs in both directions simultaneously; this requires the receiver to factor out the signal that is transmitted locally.

Some general terminology is now in order. Links that can be used in both directions at the same time, like a two-lane road, are called full-duplex links. In contrast, links that can be used in either direction, but only one way at a time, like a single-track railroad line. are called half-duplex links. A third category consists of links that allow traffic in only one direction, like a one-way street. They are called simplex links.

Returning to twisted pair, Cat 5 replaced earlier Category 3 cables with a similar cable that uses the same connector but has more twists per meter. More twists result in less crosstalk and a better-quality signal over longer distances, making the cables more suitable for high-speed computer communication, especially 100-Mbps and 1-Gbps Ethernet LANs.

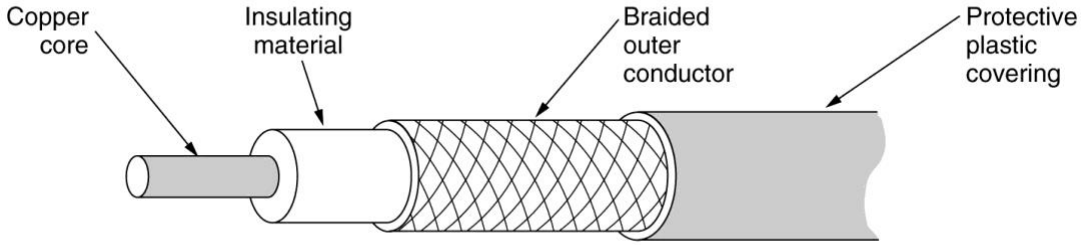
New wiring is more likely to be Category 6 or even Category 7. These categories have more stringent specifications to handle signals with greater bandwidths. Some cables in Category 6 and above are rated for signals of 500 MHz and can support the 10-Gbps links that will soon be deployed.

Through Category 6, these wiring types are referred to as UTP (Unshielded Twisted Pair) as they consist simply of wires and insulators. In contrast to these, Category 7 cables have shielding on the individual twisted pairs, as well as around the entire cable (but inside the plastic protective sheath). Shielding reduces the susceptibility to external interference and crosstalk with other nearby cables to meet demanding performance specifications. The cables are reminiscent of the high-quality, but bulky and expensive shielded twisted pair cables that IBM introduced in the early 1980s, but which did not prove popular outside of IBM installations. Evidently, it is time to try again.

## Coaxial Cable

Another common transmission medium is the coaxial cable (known to its many friends as just ‘‘coax’’ and pronounced ‘‘co-ax’’). It has better shielding and greater bandwidth than unshielded twisted pairs, so it can span longer distances at higher speeds. Two kinds of coaxial cable are widely used. One kind, 50-ohm cable, is commonly used when it is intended for digital transmission from the start. The other kind, 75-ohm cable, is commonly used for analogue transmission and cable television. This distinction is based on historical, rather than technical, factors (e.g., early dipole antennas had an impedance of 300 ohms, and it was easy to use existing 4:1 impedance-matching transformers). Starting in the mid1990s, cable TV operators began to provide Internet access over cable, which has made 75-ohm cable more important for data communication.

A coaxial cable consists of a stiff copper wire as the core, surrounded by an insulating material. The insulator is encased by a cylindrical conductor, often as a closely woven braided mesh. The outer conductor is covered in a protective plastic sheath. A cutaway view of a coaxial cable is shown in Fig. 2-4.



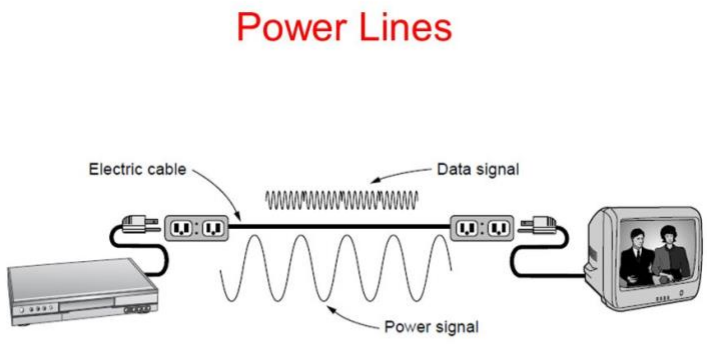

The construction and shielding of the coaxial cable give it a good combination of high bandwidth and excellent noise immunity. The bandwidth possible depends on the cable quality and length. Modern cables have a bandwidth of up to a few GHz. Coaxial cables used to be widely used within the telephone system for long-distance lines but have now largely been replaced by fibre optics on long haul routes. Coax is still widely used for cable television and metropolitan area networks, however.

**Power Lines** (not in the syllabus)

The telephone and cable television networks are not the only sources of wiring that can be reused for data communication. There is a yet more common kind of wiring: electrical power lines. Power lines deliver electrical power to houses, and electrical wiring within houses distributes the power to electrical outlets.

The use of power lines for data communication is an old idea. Power lines have been used by electricity companies for low-rate communication such as remote metering for many years, as well in the home to control devices (e.g., the X10 standard). In recent years there has been renewed interest in high-rate communication over these lines, both inside the home as a LAN and outside the home for broadband Internet access. We will concentrate on the most common scenario: using electrical wires inside the home.

The convenience of using power lines for networking should be clear. Simply plug a TV and a receiver into the wall, which you must do anyway because they need power, and they can send and receive movies over the electrical wiring. This configuration is shown in Fig. 2-5. There is no other plug or radio. The data signal is superimposed on the low-frequency power signal (on the active or ‘‘hot’’ wire) as both signals use the wiring at the same time.




A network that uses household electrical wiring.

The difficulty with using household electrical wiring for a network is that it was designed to distribute power signals. This task is quite different than distributing data signals, at which household wiring does a horrible job. Electrical signals are sent at 50–60 Hz and the wiring attenuates the much higher frequency (MHz) signals needed for high-rate data communication. The electrical properties of the wiring vary from one house to the next and change as appliances are turned on and off, which causes data signals to bounce around the wiring. Transient currents when appliances switch on and off create electrical noise over a wide range of frequencies. And without the careful twisting of twisted pairs, electrical wiring acts as a fine antenna, picking up external signals and radiating signals of its own. This behaviour means that to meet regulatory requirements, the data signal must exclude licensed frequencies such as the amateur radio bands.

Despite these difficulties, it is practical to send at least 100 Mbps over typical household electrical wiring by using communication schemes that resist impaired frequencies and bursts of errors. Many products use various proprietary standards for power-line networking, so international standards are actively under development.

## Fibre Optics

Many people in the computer industry take enormous pride in how fast computer technology is improving as it follows Moore's law, which predicts a doubling of the number of transistors per chip roughly every two years (Schaller, 1997). The original (1981) IBM PC ran at a clock speed of 4.77 MHz. 28 years later, PCs could run a four-core CPU at 3 GHz. This increase is a gain of a factor of around 2500, or 16 per decade. Impressive.

In the same period, wide area communication links went from 45 Mbps (a T3 line in the telephone system) to 100 Gbps (a modern long-distance line). This gain is similarly impressive, more than a factor of 2000 and close to 16 per decade, while at the same time the error rate went from $10^{-5}$ per bit to almost zero. Furthermore, single CPUs are beginning to approach physical limits, which is why it is now the number of CPUs that is being increased per chip. In contrast, the achievable bandwidth with fibre

technology is in excess of 50,000 Gbps (50 Tbps) and we are nowhere near reaching these limits. The current practical limit of around 100 Gbps is due to our inability to convert between electrical and optical signals any faster. To build higher-capacity links, many channels are simply carried in parallel over a single fibre.

In this section we will study fibre optics to learn how that transmission technology works. In the ongoing race between computing and communication, communication may yet win because of fibre optic networks. The implication of this would be essentially infinite bandwidth and a new conventional wisdom that computers are hopelessly slow so that networks should try to avoid computation at all costs, no matter how much bandwidth that wastes. This change will take a while to sink into a generation of computer scientists and engineers taught to think in terms of the low Shannon limits imposed by copper.

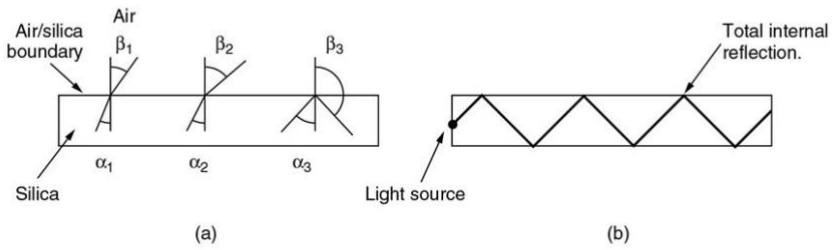
Of course, this scenario does not tell the whole story because it does not include cost. The cost to install fibre over the last mile to reach consumers and bypass the low bandwidth of wires and limited availability of spectrum is tremendous. It also costs more energy to move bits than to compute. We may always have islands of inequities where either computation or communication is essentially free. For example, at the edge of the Internet we throw computation and storage at the problem of compressing and caching content, all to make better use of Internet access links. Within the Internet, we may do the reverse, with companies such as Google moving huge amounts of data across the network to where it is cheaper to store or compute on it.

Fibre optics are used for long-haul transmission in network backbones, highspeed LANs (although so far, copper has always managed catch up eventually), and high-speed Internet access such as FTTH (Fibre to the Home). An optical transmission system has three key components: the light source, the transmission medium, and the detector. Conventionally, a pulse of light indicates a 1 bit, and the absence of light indicates a 0 bit. The transmission medium is an ultra-thin fibre of glass. The detector generates an electrical pulse when light falls on it. By attaching a light source to one end of an optical fibre and a detector to the other, we have a unidirectional

data transmission system that accepts an electrical signal, converts and transmits it by light pulses, and then reconverts the output to an electrical signal at the receiving end.

This transmission system would leak light and be useless in practice were it not for an interesting principle of physics. When a light ray passes from one medium to another—for example, from fused silica to air—the ray is refracted (bent) at the silica/air boundary, as shown in Fig. 2-6(a). Here we see a light ray incident on the boundary at an angle α1 emerging at an angle β1. The amount of refraction depends on the properties of the two media (in particular, their indices of refraction). For angles of incidence above a certain critical value, the light is refracted back into the silica; none of it escapes into the air. Thus, a light ray incident at or above the critical angle is trapped inside the fibre, as shown in Fig. 2-6(b), and can propagate for many kilometres with virtually no loss.

## Fiber Optics




(a) Three examples of a light ray from inside a silica fiber impinging on the air/silica boundary at different angles.
(b) Light trapped by total internal reflection.

The sketch of Fig. 2-6(b) shows only one trapped ray, but since any light ray incident on the boundary above the critical angle will be reflected internally, many different rays will be bouncing around at different angles. Each ray is said to have a different mode, so a fibre having this property is called a multimode fibre.

However, if the fibre's diameter is reduced to a few wavelengths of light the fibre acts like a wave guide and the light can propagate only in a straight line, without bouncing, yielding a single-mode fibre. Single mode fibres are more expensive but are widely used for longer distances. Currently available single mode fibres can transmit data at 100 Gbps for 100 km without amplification. Even higher data rates have been achieved in the laboratory for shorter distances.

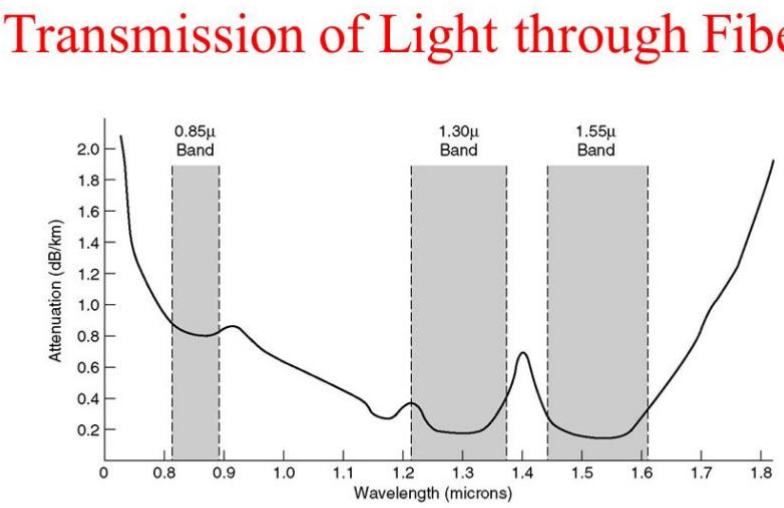
## Transmission of Light Through Fibre

Optical fibres are made of glass, which, in turn, is made from sand, an inexpensive raw material available in unlimited amounts. Glassmaking was known to the ancient Egyptians, but their glass had to be no more than 1 mm thick, or the light could not shine through. Glass transparent enough to be useful for windows was developed during the Renaissance. The glass used for modern optical fibres is so transparent that if the oceans were full of it instead of water, the seabed would be as visible from the surface as the ground is from an airplane on a clear day.

The attenuation of light through glass depends on the wavelength of the light (as well as on some physical properties of the glass). It is defined as the ratio of input to output signal power. For the kind of glass used in fibres, the attenuation is shown in Fig. 2-7 in units of decibels per linear kilometre of fibre. For example, a factor of two loss of signal power gives an attenuation of 10 log10 2 = 3 dB. The figure shows the near-infrared part of the spectrum, which is what is used in practice. Visible light has slightly shorter wavelengths, from 0.4 to 0.7 microns. (1 micron is $10^{-6}$ meters.) The true metric purist would refer to these wavelengths as 400 nm to 700 nm, but we will stick with traditional usage.

Three wavelength bands are most used at present for optical communication. They are centred at 0.85, 1.30, and 1.55 microns, respectively. All three bands are 25,000 to 30,000 GHz wide. The 0.85-micron band was used first. It has higher attenuation and so is used for shorter distances, but at that wavelength the lasers and electronics could be made from the same material (gallium arsenide). The last two bands have

good attenuation properties (less than 5% loss per kilometre). The 1.55-micron band is now widely used with erbium-doped amplifiers that work directly in the optical domain.



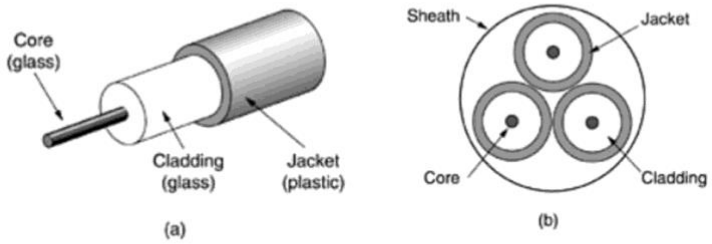

Attenuation of light through fiber in the infrared region.

Light pulses sent down a fibre spread out in length as they propagate. This spreading is called chromatic dispersion. The amount of it is wavelength dependent. One way to keep these spread-out pulses from overlapping is to increase the distance between them, but this can be done only by reducing the signalling rate. Fortunately, it has been discovered that making the pulses in a special shape related to the reciprocal of the hyperbolic cosine causes nearly all the dispersion effects cancel out, so it is possible to send pulses for thousands of kilometres without appreciable shape distortion. These pulses are called solitons. A considerable amount of research is going on to take solitons out of the lab and into the field.

## Fibre Cables

Fibre optic cables are like coax, except without the braid. Figure 2-8(a) shows a single fibre viewed from the side. At the centre is the glass core through which the light propagates. In multimode fibres, the core is typically 50 microns in diameter, about the thickness of a human hair. In single mode fibres, the core is 8 to 10 microns.

**Fiber optic cable**




**(a) Side view of a single fiber. (b) End view of a sheath with three fibers.**

The core is surrounded by a glass cladding with a lower index of refraction than the core, to keep all the light in the core. Next comes a thin plastic jacket to protect the cladding. fibres are typically grouped in bundles, protected by an outer sheath. Figure 2-8(b) shows a sheath with three fibres.

Terrestrial fibre sheaths are normally laid in the ground within a meter of the surface, where they are occasionally subject to attacks by backhoes or gophers. Near the shore, transoceanic fibre sheaths are buried in trenches by a kind of seaplow. In deep water, they just lie on the bottom, where they can be snagged by fishing trawlers or attacked by giant squid.

fibres can be connected in three different ways. First, they can terminate in connectors and be plugged into fibre sockets. Connectors lose about 10 to 20% of the light, but they make it easy to reconfigure systems.

Second, they can be spliced mechanically. Mechanical splices just lay the two carefully cut ends next to each other in a special sleeve and clamp them in place. Alignment can be improved by passing light through the junction and then making small adjustments to maximize the signal. Mechanical splices take trained personnel about 5 minutes and result in a 10% light loss.

Third, two pieces of fibre can be fused (melted) to form a solid connection. A fusion splice is almost as good as a single drawn fibre, but even here, a small amount of attenuation occurs.

For all three kinds of splices, reflections can occur at the point of the splice, and the reflected energy can interfere with the signal.

Two kinds of light sources are typically used to do the signalling. These are LEDs (Light Emitting Diodes) and semiconductor lasers. They have different properties, as shown in Fig. 2-9. They can be tuned in wavelength by inserting Fabry-Perot or Mach-Zehnder interferometers between the source and the fibre. Fabry-Perot interferometers are simple resonant cavities consisting of two parallel mirrors. The light is incident perpendicular to the mirrors. The length of the cavity selects out those wavelengths that fit inside an integral number of times. Mach-Zehnder interferometers separate the light into two beams. The two beams travel slightly different distances. They are recombined at the end and are in phase for only certain wavelengths.

## Fiber Cables (2)

| Item | LED | Semiconductor laser |
|---|---|---|
| Data rate | Low | High |
| Fiber type | Multimode | Multimode or single mode |
| Distance | Short | Long |
| Lifetime | Long life | Short life |
| Temperature sensitivity | Minor | Substantial |
| Cost | Low cost | Expensive |

A comparison of semiconductor diodes and LEDs as light sources.

The receiving end of an optical fibre consists of a photodiode, which gives off an electrical pulse when struck by light. The response time of photodiodes, which convert the signal from the optical to the electrical domain, limits data rates to about 100 Gbps. Thermal noise is also an issue,

so a pulse of light must carry enough energy to be detected. By making the pulses powerful enough, the error rate can be made arbitrarily small.

**Comparison of Fibre Optics & Copper Wire**

It is instructive to compare fibre to copper. Fibre has many advantages. To start with, it can handle much higher bandwidths than copper. This alone would require its use in high-end networks. Due to the low attenuation, repeaters are needed only about every 50 km on long lines, versus about every 5 km for copper, resulting in a big cost saving. Fibre also has the advantage of not being affected by power surges, electromagnetic interference, or power failures. Nor is it affected by corrosive chemicals in the air, important for harsh factory environments.

Oddly enough, telephone companies like fibre for a different reason: it is thin and lightweight. Many existing cable ducts are completely full, so there is no room to add new capacity. Removing all the copper and replacing it with fibre empties the ducts, and the copper has excellent resale value to copper refiners who see it as very high-grade ore. Also, fibre is much lighter than copper. One thousand twisted pairs 1 km long weigh 8000 kg. Two fibres have more capacity and weigh only 100 kg, which reduces the need for expensive mechanical support systems that must be maintained. For new routes, fibre wins hands down due to its much lower installation cost. Finally, fibres do not leak light and are difficult to tap. These properties give fibre good security against potential wiretappers.

On the downside, fibre is a less familiar technology requiring skills not all engineers have, and fibres can be damaged easily by being bent too much. Since optical transmission is inherently unidirectional, two-way communication requires either two fibres or two frequency bands on one fibre. Finally, fibre interfaces cost more than electrical interfaces. Nevertheless, the future of all fixed data communication over more than short distances is clearly with fibre. For a discussion of all aspects of fibre optics and their networks, see Hecht (2005).

### 1.8.2 Unguided/ Wireless Media

Our age has given rise to information junkies: people who need to be online all the time. For these mobile users, twisted pair, coax, and fibre optics are of no use. They need to get their ‘‘hits’’ of data for their laptop, notebook, shirt pocket, palmtop, or wristwatch computers without being tethered to the terrestrial communication infrastructure. For these users, wireless communication is the answer.

In the following sections, we will look at wireless communication in general. It has many other important applications besides providing connectivity to users who want to surf the Web from the beach. Wireless has advantages for even fixed devices in some circumstances. For example, if running a fibre to a building is difficult due to the terrain (mountains, jungles, swamps, etc.), wireless may be better. It is noteworthy that modern wireless digital communication began in the Hawaiian Islands, where large chunks of Pacific Ocean separated the users from their computer centre and the telephone system was inadequate.

**The Electromagnetic Spectrum** (not in the syllabus)

When electrons move, they create electromagnetic waves that can propagate through space (even in a vacuum). These waves were predicted by the British physicist James Clerk Maxwell in 1865 and first observed by the German physicist Heinrich Hertz in 1887. The number of oscillations per second of a wave is called its frequency, f, and is measured in Hz (in honour of Heinrich Hertz). The distance between two consecutive maxima (or minima) is called the wavelength, which is universally designated by the Greek letter λ (lambda).

When an antenna of the appropriate size is attached to an electrical circuit, the electromagnetic waves can be broadcast efficiently and received by a receiver some distance away. All wireless communication is based on this principle.

In a vacuum, all electromagnetic waves travel at the same speed, no matter what their frequency. This speed, usually called the speed of light, c, is approximately 3 × 108 m/sec, or about 1 foot (30 cm) per nanosecond. (A case could be made for redefining the foot as the distance light travels in a

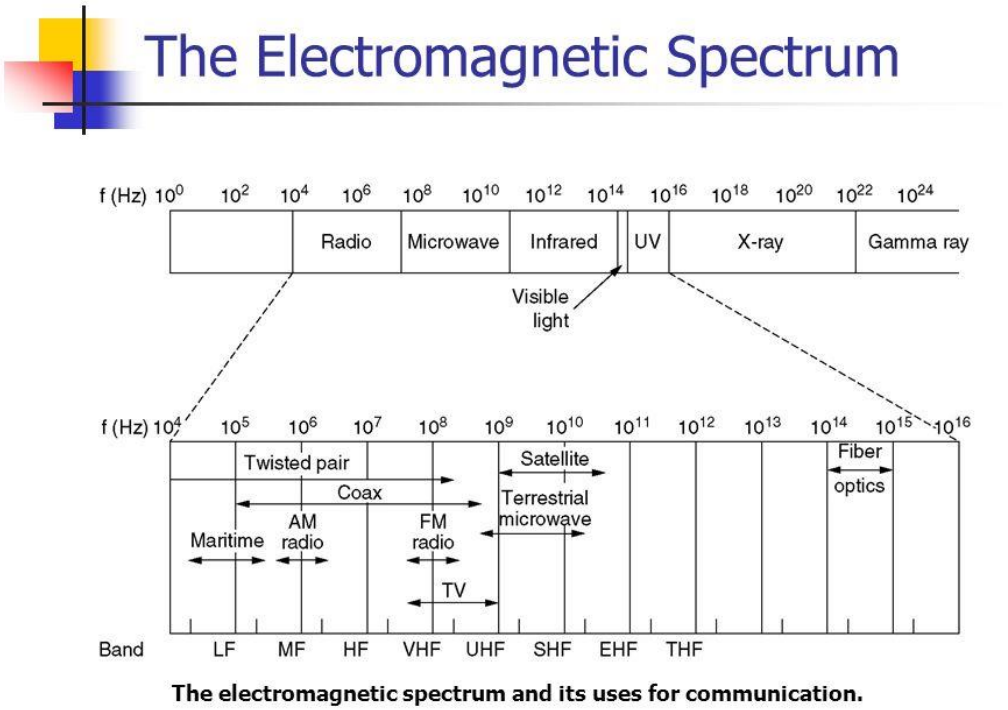
vacuum in 1 nsec rather than basing it on the shoe size of some long-dead king.) In copper or fibre the speed slows to about 2/3 of this value and becomes slightly frequency dependent. The speed of light is the ultimate speed limit. No object or signal can ever move faster than it.

The fundamental relation between f, λ, and c (in a vacuum) is $\lambda f = c$ (2-4) Since c is a constant, if we know f, we can find λ, and vice versa. As a rule of thumb, when λ is in meters and f is in MHz, $\lambda f \sim\sim 300$. For example, 100-MHz waves are about 3 meters long, 1000-MHz waves are 0.3 meters long, and 0.1- meter waves have a frequency of 3000 MHz.

The electromagnetic spectrum is shown in Fig. 2-10. The radio, microwave, infrared, and visible light portions of the spectrum can all be used for transmitting information by modulating the amplitude, frequency, or phase of the waves. Ultraviolet light, X-rays, and gamma rays would be even better, due to their higher frequencies, but they are hard to produce and modulate, do not propagate well through buildings, and are dangerous to living things. The bands listed at the bottom of Fig. 2-10 are the official ITU (International Telecommunication Union) names and are based on the wavelengths, so the LF band goes from 1 km to 10 km (approximately 30 kHz to 300 kHz). The terms LF, MF, and HF refer to Low, Medium, and High Frequency, respectively. Clearly, when the names were assigned, nobody expected to go above 10 MHz, so the higher bands were later named the Very, Ultra, Super, Extremely, and Tremendously High Frequency bands. Beyond that there are no names, but Incredibly, Astonishingly, and Prodigiously High Frequency (IHF, AHF, and PHF) would sound nice.

We know from Shannon [Eq. (2-3)] that the amount of information that a signal such as an electromagnetic wave can carry depends on the received power and is proportional to its bandwidth. From Fig. 2-10 it should now be obvious why networking people like fibre optics so much. Many GHz of bandwidth are available to tap for data transmission in the microwave band, and even more in fibre because it is further to the right in our logarithmic scale. As an example, consider the 1.30-micron band of Fig. 2-7, which has a width of 0.17 microns. If we use Eq. (2-4) to find the start and end frequencies from the start and end wavelengths, we find the

frequency range to be about 30,000 GHz. With a reasonable signal to-noise ratio of 10 dB, this is 300 Tbps.




**The electromagnetic spectrum and its uses for communication.**

Most transmissions use a relatively narrow frequency band (i.e., $\Delta f/f << 1$). They concentrate their signals in this narrow band to use the spectrum efficiently and obtain reasonable data rates by transmitting with enough power. However, in some cases, a wider band is used, with three variations. In frequency hopping spread spectrum, the transmitter hops from frequency-to-frequency hundreds of times per second. It is popular for military communication because it makes transmissions hard to detect and next to impossible to jam. It also offers good resistance to multipath fading and narrowband interference because the receiver will not be stuck on an impaired frequency for long enough to shut down communication. This robustness makes it useful for crowded parts of the spectrum, such as the ISM bands we will describe shortly. This technique is used commercially, for example, in Bluetooth and older versions of 802.11.

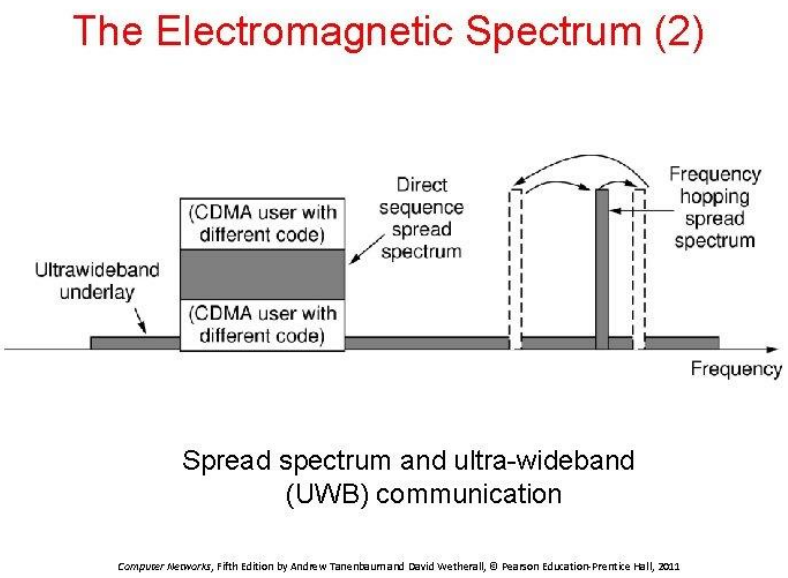
As a curious footnote, the technique was coinvented by the Austrian-born sex goddess Hedy Lamarr, the first woman to appear nude in a motion

picture (the 1933 Czech film Extase). Her first husband was an armaments manufacturer who told her how easy it was to block the radio signals then used to control torpedoes. When she discovered that he was selling weapons to Hitler, she was horrified, disguised herself as a maid to escape him, and fled to Hollywood to continue her career as a movie actress. In her spare time, she invented frequency hopping to help the Allied war effort. Her scheme used 88 frequencies, the number of keys (and frequencies) on the piano. For their invention, she and her friend, the musical composer George Antheil, received U.S. patent 2,292,387. However, they were unable to convince the U.S. Navy that their invention had any practical use and never received any royalties. Only years after the patent expired did it become popular.

A second form of spread spectrum, direct sequence spread spectrum, uses a code sequence to spread the data signal over a wider frequency band. It is widely used commercially as a spectrally efficient way to let multiple signals share the same frequency band. These signals can be given different codes, a method called CDMA (Code Division Multiple Access) that we will return to later in this chapter. This method is shown in contrast with frequency hopping in Fig. 2-11. It forms the basis of 3G mobile phone networks and is also used in GPS (Global Positioning System). Even without different codes, direct sequence spread spectrum, like frequency hopping spread spectrum, can tolerate narrowband interference and multipath fading because only a fraction of the desired signal is lost. It is used in this role in older 802.11b wireless LANs. For a fascinating and detailed history of spread spectrum communication, see Scholtz (1982).

A third method of communication with a wider band is UWB (Ultra-wideband) communication. UWB sends a series of rapid pulses, varying their positions to communicate information. The rapid transitions lead to a signal that is spread thinly over a very wide frequency band. UWB is defined as signals that have a bandwidth of at least 500 MHz or at least 20% of the centre frequency of their frequency band. UWB is also shown in Fig. 2-11. With this much bandwidth, UWB has the potential to communicate at high rates. Because it is spread across a wide band of frequencies, it can tolerate a substantial amount of relatively strong interference from other narrowband signals. Just as importantly, since

UWB has very little energy at any given frequency when used for short-range transmission, it does not cause harmful interference to those other narrowband radio signals. It is said to underlay the other signals. This peaceful coexistence has led to its application in wireless PANs that run at up to 1 Gbps, although commercial success has been mixed. It can also be used for imaging through solid objects (ground, walls, and bodies) or as part of precise location systems.




We will now discuss how the various parts of the electromagnetic spectrum of Fig. 2-11 are used, starting with radio. We will assume that all transmissions use a narrow frequency band unless otherwise stated.

## Radio Transmission

Radio frequency (RF) waves are easy to generate, can travel long distances, and can penetrate buildings easily, so they are widely used for communication, both indoors and outdoors. Radio waves also are omnidirectional, meaning that they travel in all directions from the source, so the transmitter and receiver do not have to be carefully aligned physically.

Sometimes omnidirectional radio is good, but sometimes it is bad. In the 1970s, General Motors decided to equip all its new Cadillacs with computer-controlled antilock brakes. When the driver stepped on the brake

pedal, the computer pulsed the brakes on and off instead of locking them on hard. One fine day an Ohio Highway Patrolman began using his new mobile radio to call headquarters, and suddenly the Cadillac next to him began behaving like a bucking bronco. When the officer pulled the car over, the driver claimed that he had done nothing and that the car had gone crazy.

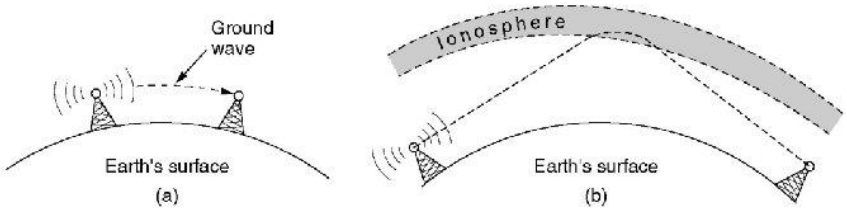
Eventually, a pattern began to emerge: Cadillacs would sometimes go berserk, but only on major highways in Ohio and then only when the Highway Patrol was watching. For a long, long time General Motors could not understand why Cadillacs worked fine in all the other states and also on minor roads in Ohio. Only after much searching did they discover that the Cadillac's wiring made a fine antenna for the frequency used by the Ohio Highway Patrol's new radio system.

The properties of radio waves are frequency dependent. At low frequencies, radio waves pass through obstacles well, but the power falls off sharply with distance from the source—at least as fast as 1/r 2 in air—as the signal energy is spread more thinly over a larger surface. This attenuation is called path loss. At high frequencies, radio waves tend to travel in straight lines and bounce off obstacles. Path loss still reduces power, though the received signal can depend strongly on reflections as well. High-frequency radio waves are also absorbed by rain and other obstacles to a larger extent than are low-frequency ones. At all frequencies, radio waves are subject to interference from motors and other electrical equipment.

It is interesting to compare the attenuation of radio waves to that of signals in guided media. With fibre, coax and twisted pair, the signal drops by the same fraction per unit distance, for example 20 dB per 100m for twisted pair. With radio, the signal drops by the same fraction as the distance doubles, for example 6 dB per doubling in free space. This behaviour means that radio waves can travel long distances, and interference between users is a problem. For this reason, all governments tightly regulate the use of radio transmitters, with few notable exceptions, which are discussed later in this chapter.

In the VLF, LF, and MF bands, radio waves follow the ground, as illustrated in Fig. 2-12(a). These waves can be detected for perhaps 1000 km at the lower frequencies, less at the higher ones. AM radio broadcasting uses the MF band, which is why the ground waves from Boston AM radio stations cannot be heard easily in New York. Radio waves in these bands pass through buildings easily, which is why portable radios work indoors. The main problem with using these bands for data communication is their low bandwidth [see Eq. (2-4)].

## Radio Transmission (3)




(a) In the VLF, LF, and MF bands, radio waves follow the curvature of the earth.
(b) In the HF band, they bounce off the ionosphere.

39

In the HF and VHF bands, the ground waves tend to be absorbed by the earth. However, the waves that reach the ionosphere, a layer of charged particles circling the earth at a height of 100 to 500 km, are refracted by it and sent back to earth, as shown in Fig. 2-12(b). Under certain atmospheric conditions, the signals can bounce several times. Amateur radio operators (hams) use these bands to talk long distance. The military also communicate in the HF and VHF bands.

**Microwave Transmission**

Above 100 MHz, the waves travel in nearly straight lines and can therefore be narrowly focused. Concentrating all the energy into a small beam by means of a parabolic antenna (like the familiar satellite TV dish) gives a

much higher signal to-noise ratio, but the transmitting and receiving antennas must be accurately aligned with each other. In addition, this directionality allows multiple transmitters lined up in a row to communicate with multiple receivers in a row without interference, provided some minimum spacing rules are observed. Before fibre optics, for decades these microwaves formed the heart of the long-distance telephone transmission system. In fact, MCI, one of AT&T's first competitors after it was deregulated, built its entire system with microwave communications passing between towers tens of kilometres apart. Even the company's name reflected this (MCI stood for Microwave Communications, Inc.). MCI has since gone over to fibre and through a long series of corporate mergers and bankruptcies in the telecommunications shuffle has become part of Verizon.

Microwaves travel in a straight line, so if the towers are too far apart, the earth will get in the way (think about a Seattle-to-Amsterdam link). Thus, repeaters are needed periodically. The higher the towers are, the farther apart they can be. The distance between repeaters goes up very roughly with the square root of the tower height. For 100-meter-high towers, repeaters can be 80 km apart.

Unlike radio waves at lower frequencies, microwaves do not pass through buildings well. In addition, even though the beam may be well focused at the transmitter, there is still some divergence in space. Some waves may be refracted off low-lying atmospheric layers and may take slightly longer to arrive than the direct waves. The delayed waves may arrive out of phase with the direct wave and thus cancel the signal. This effect is called multipath fading and is often a serious problem. It is weather and frequency dependent. Some operators keep 10% of their channels idle as spares to switch on when multipath fading temporarily wipes out some frequency band.

The demand for more and more spectrum drives operators to yet higher frequencies. Bands up to 10 GHz are now in routine use, but at about 4 GHz a new problem sets in: absorption by water. These waves are only a few centimetres long and are absorbed by rain. This effect would be fine if one were planning to build a huge outdoor microwave oven for roasting

passing birds, but for communication it is a severe problem. As with multipath fading, the only solution is to shut off links that are being rained on and route around them.

In summary, microwave communication is so widely used for long-distance telephone communication, mobile phones, television distribution, and other purposes that a severe shortage of spectrum has developed. It has several key advantages over fibre. The main one is that no right of way is needed to lay down cables. By buying a small plot of ground every 50 km and putting a microwave tower on it, one can bypass the telephone system entirely. This is how MCI managed to get started as a new long-distance telephone company so quickly. (Sprint, another early competitor to the deregulated AT&T, went a completely different route: it was formed by the Southern Pacific Railroad, which already owned a large amount of right of way and just buried fibre next to the tracks.)

Microwave is also relatively inexpensive. Putting up two simple towers (which can be just big poles with four guy wires) and putting antennas on each one may be cheaper than burying 50 km of fibre through a congested urban area or up over a mountain, and it may also be cheaper than leasing the telephone company's fibre, especially if the telephone company has not yet even fully paid for the copper it ripped out when it put in the fibre.

## The Politics of the Electromagnetic Spectrum

To prevent total chaos, there are national and international agreements about who gets to use which frequencies. Since everyone wants a higher data rate, everyone wants more spectrum. National governments allocate spectrum for AM and FM radio, television, and mobile phones, as well as for telephone companies, police, maritime, navigation, military, government, and many other competing users. Worldwide, an agency of ITU-R (WRC) tries to coordinate this allocation so devices that work in multiple countries can be manufactured. However, countries are not bound by ITU-R's recommendations, and the FCC (Federal Communication Commission), which does the allocation for the United States, has occasionally rejected ITU-R's recommendations (usually because they

required some politically powerful group to give up some piece of the spectrum).

Even when a piece of spectrum has been allocated to some use, such as mobile phones, there is the additional issue of which carrier is allowed to use which frequencies. Three algorithms were widely used in the past. The oldest algorithm, often called the beauty contest, requires each carrier to explain why its proposal serves the public interest best. Government officials then decide which of the nice stories they enjoy most. Having some government official award property worth billions of dollars to his favourite company often leads to bribery, corruption, nepotism, and worse. Furthermore, even a scrupulously honest government official who thought that a foreign company could do a better job than any of the national companies would have a lot of explaining to do.

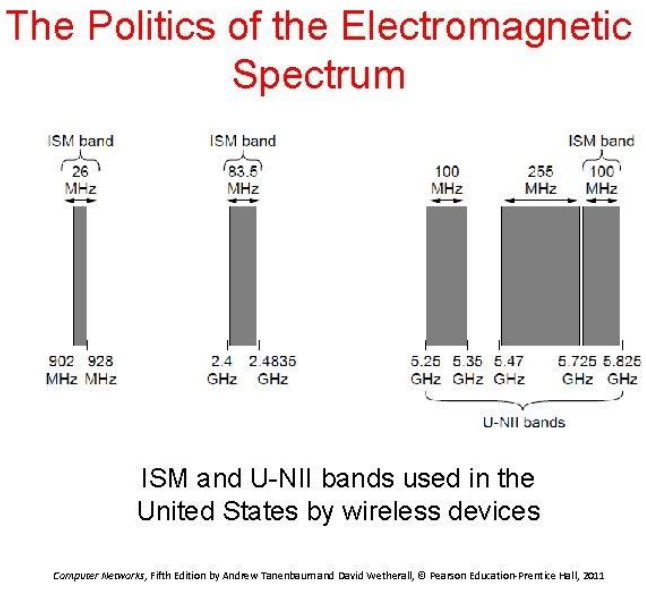
This observation led to algorithm 2, holding a lottery among the interested companies. The problem with that idea is that companies with no interest in using the spectrum can enter the lottery. If, say, a fast-food restaurant or shoe store chain wins, it can resell the spectrum to a carrier at a huge profit and with no risk.

Bestowing huge windfalls on alert but otherwise random companies has been severely criticized by many, which led to algorithm 3: auction off the bandwidth to the highest bidder. When the British government auctioned off the frequencies needed for third-generation mobile systems in 2000, it expected to get about $4 billion. It received about $40 billion because the carriers got into a feeding frenzy, scared to death of missing the mobile boat. This event switched on nearby governments' greedy bits and inspired them to hold their own auctions. It worked, but it also left some of the carriers with so much debt that they are close to bankruptcy. Even in the best cases, it will take many years to recoup the licensing fee.

A completely different approach to allocating frequencies is to not allocate them at all. Instead, let everyone transmit at will, but regulate the power used so that stations have such a short range that they do not interfere with each other. Accordingly, most governments have set aside some frequency bands, called the ISM (Industrial, Scientific, Medical) bands for unlicensed

usage. Garage door openers, cordless phones, radio-controlled toys, wireless mice, and numerous other wireless household devices use the ISM bands. To minimize interference between these uncoordinated devices, the FCC mandates that all devices in the ISM bands limit their transmit power (e.g., to 1 watt) and use other techniques to spread their signals over a range of frequencies. Devices may also need to take care to avoid interference with radar installations.

The location of these bands varies somewhat from country to country. In the United States, for example, the bands that networking devices use in practice without requiring a FCC license are shown in Fig. 2-13. The 900-MHz band was used for early versions of 802.11, but it is crowded. The 2.4-GHz band is available in most countries and widely used for 802.11b/g and Bluetooth, though it is subject to interference from microwave ovens and radar installations. The 5-GHz part of the spectrum includes U-NII (Unlicensed National Information Infrastructure) bands. The 5-GHz bands are relatively undeveloped but, since they have the most bandwidth and are used by 802.11a, they are quickly gaining in popularity.




The unlicensed bands have been a roaring success over the past decade. The ability to use the spectrum freely has unleashed a huge amount of innovation in wireless LANs and PANs, evidenced by the widespread deployment of technologies such as 802.11 and Bluetooth. To continue this

innovation, more spectrum is needed. One exciting development in the U.S. is the FCC decision in 2009 to allow unlicensed use of white spaces around 700 MHz. White spaces are frequency bands that have been allocated but are not being used locally. The transition from analogue to all-digital television broadcasts in the U.S. in 2010 freed up white spaces around 700 MHz. The only difficulty is that, to use the white spaces, unlicensed devices must be able to detect any nearby licensed transmitters, including wireless microphones, that have first rights to use the frequency band.

Another flurry of activity is happening around the 60-GHz band. The FCC opened 57 GHz to 64 GHz for unlicensed operation in 2001. This range is an enormous portion of spectrum, more than all the other ISM bands combined, so it can support the kind of high-speed networks that would be needed to stream high-definition TV through the air across your living room. At 60 GHz, radio waves are absorbed by oxygen. This means that signals do not propagate far, making them well suited to short-range networks. The high frequencies (60 GHz is in the Extremely High Frequency or ‘‘millimetre’’ band, just below infrared radiation) posed an initial challenge for equipment makers, but products are now on the market.

## Infrared Transmission

Unguided infrared waves are widely used for short-range communication. The remote controls used for televisions, VCRs, and stereos all use infrared communication. They are relatively directional, cheap, and easy to build but have a major drawback: they do not pass-through solid objects. (Try standing between your remote control and your television and see if it still works.) In general, as we go from long-wave radio toward visible light, the waves behave more and more like light and less and less like radio.

On the other hand, the fact that infrared waves do not pass through solid walls well is also a plus. It means that an infrared system in one room of a building will not interfere with a similar system in adjacent rooms or buildings: you cannot control your neighbour’s television with your remote control. Furthermore, security of infrared systems against eavesdropping is better than that of radio systems precisely for this reason. Therefore, no

government license is needed to operate an infrared system, in contrast to radio systems, which must be licensed outside the ISM bands. Infrared communication has a limited use on the desktop, for example, to connect notebook computers and printers with the IrDA (Infrared Data Association) standard, but it is not a major player in the communication game.

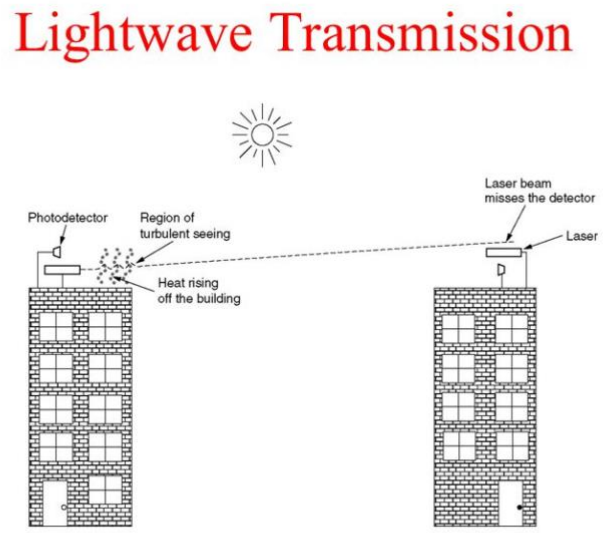
**Light Transmission** (not in the syllabus)

Unguided optical signalling or free-space optics has been in use for centuries. Paul Revere used binary optical signalling from the Old North Church just prior to his famous ride. A more modern application is to connect the LANs in two buildings via lasers mounted on their rooftops. Optical signalling using lasers is inherently unidirectional, so each end needs its own laser and its own photodetector. This scheme offers very high bandwidth at very low cost and is relatively secure because it is difficult to tap a narrow laser beam. It is also relatively easy to install and, unlike microwave transmission, does not require an FCC license.

The laser's strength, a very narrow beam, is also its weakness here. Aiming a laser beam 1 mm wide at a target the size of a pin head 500 meters away requires the marksmanship of a latter-day Annie Oakley. Usually, lenses are put into the system to defocus the beam slightly. To add to the difficulty, wind and temperature changes can distort the beam and laser beams also cannot penetrate rain or thick fog, although they normally work well on sunny days. However, many of these factors are not an issue when the use is to connect two spacecraft.

One of the authors (AST) once attended a conference at a modern hotel in Europe at which the conference organizers thoughtfully provided a room full of terminals to allow the attendees to read their email during boring presentations. Since the local PTT was unwilling to install many telephone lines for just 3 days, the organizers put a laser on the roof and aimed it at their university's computer science building a few kilometres away. They tested it the night before the conference, and it worked perfectly. At 9 A.M. on a bright, sunny day, the link failed completely and stayed down all day. The pattern repeated itself the next two days. It was not until after the

conference that the organizers discovered the problem: heat from the sun during the daytime caused convection currents to rise from the roof of the building, as shown in Fig. 2-14. This turbulent air diverted the beam and made it dance around the detector, much like a shimmering road on a hot day. The lesson here is that to work well in difficult conditions as well as good conditions, unguided optical links need to be engineered with a sufficient margin of error.




Convection currents can interfere with laser communication systems.
cn ch2A bidirectional system with two lasers is pictured here. 19

Unguided optical communication may seem like an exotic networking technology today, but it might soon become much more prevalent. We are surrounded by cameras (that sense light) and displays (that emit light using LEDs and other technology). Data communication can be layered on top of these displays by encoding information in the pattern at which LEDs turn on and off that is below the threshold of human perception. Communicating with visible light in this way is inherently safe and creates a low-speed network in the immediate vicinity of the display. This could enable all sorts of fanciful ubiquitous computing scenarios. The flashing lights on emergency vehicles might alert nearby traffic lights and vehicles to help clear a path. Informational signs might broadcast maps. Even festive lights might broadcast songs that are synchronized with their display.

**Communication Satellites** (not in the syllabus)

In the 1950s and early 1960s, people tried to set up communication systems by bouncing signals off metallized weather balloons. Unfortunately, the received signals were too weak to be of any practical use. Then the U.S. Navy noticed a kind of permanent weather balloon in the sky—the moon—and built an operational system for ship-to-shore communication by bouncing signals off it.

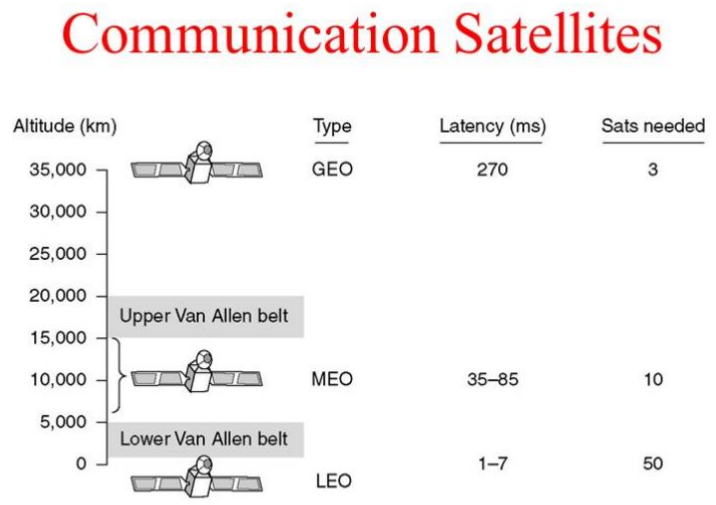
Further progress in the celestial communication field had to wait until the first communication satellite was launched. The key difference between an artificial satellite and a real one is that the artificial one can amplify the signals before sending them back, turning a strange curiosity into a powerful communication system.

Communication satellites have some interesting properties that make them attractive for many applications. In its simplest form, a communication satellite can be thought of as a big microwave repeater in the sky. It contains several transponders, each of which listens to some portion of the spectrum, amplifies the incoming signal, and then rebroadcasts it at another frequency to avoid interference with the incoming signal. This mode of operation is known as a bent pipe. Digital processing can be added to separately manipulate or redirect data streams in the overall band, or digital information can even be received by the satellite and rebroadcast. Regenerating signals in this way improves performance compared to a bent pipe because the satellite does not amplify noise in the upward signal. The downward beams can be broad, covering a substantial fraction of the earth's surface, or narrow, covering an area only hundreds of kilometres in diameter.

According to Kepler's law, the orbital period of a satellite varies as the radius of the orbit to the 3/2 power. The higher the satellite, the longer the period. Near the surface of the earth, the period is about 90 minutes. Consequently, low-orbit satellites pass out of view quickly, so many of them are needed to provide continuous coverage and ground antennas must track them. At an altitude of about 35,800 km, the period is 24 hours. At

an altitude of 384,000 km, the period is about one month, as anyone who has observed the moon regularly can testify.

A satellite's period is important, but it is not the only issue in determining where to place it. Another issue is the presence of the Van Allen belts, layers of highly charged particles trapped by the earth's magnetic field. Any satellite flying within them would be destroyed quickly by the particles. These factors lead to three regions in which satellites can be placed safely. These regions and some of their properties are illustrated in Fig. 2-15. Below we will briefly describe the satellites that inhabit each of these regions.




Communication satellites and some of their properties, including altitude above the earth, round-trip delay time and number of satellites needed for global coverage.

## Geostationary Satellites

In 1945, the science fiction writer Arthur C. Clarke calculated that a satellite at an altitude of 35,800 km in a circular equatorial orbit would appear to remain motionless in the sky, so it would not need to be tracked (Clarke, 1945). He went on to describe a complete communication system that used these (manned) geostationary satellites, including the orbits, solar panels, radio frequencies, and launch procedures. Unfortunately, he concluded that satellites were impractical due to the impossibility of putting power-hungry, fragile vacuum tube amplifiers into orbit, so he

never pursued this idea further, although he wrote some science fiction stories about it.

The invention of the transistor changed all that, and the first artificial communication satellite, Telstar, was launched in July 1962. Since then, communication satellites have become a multibillion-dollar business and the only aspect of outer space that has become highly profitable. These high-flying satellites are often called GEO (Geostationary Earth Orbit) satellites.

With current technology, it is unwise to have geostationary satellites spaced much closer than 2 degrees in the 360-degree equatorial plane, to avoid interference. With a spacing of 2 degrees, there can only be 360/2 = 180 of these satellites in the sky at once. However, each transponder can use multiple frequencies and polarizations to increase the available bandwidth. To prevent total chaos in the sky, orbit slot allocation is done by ITU. This process is highly political, with countries barely out of the stone age demanding ‘‘their’’ orbit slots (for the purpose of leasing them to the highest bidder). Other countries, however, maintain that national property rights do not extend up to the moon and that no country has a legal right to the orbit slots above its territory. To add to the fight, commercial telecommunication is not the only application. Television broadcasters, governments, and the military also want a piece of the orbiting pie.

Modern satellites can be quite large, weighing over 5000 kg and consuming several kilowatts of electric power produced by the solar panels. The effects of solar, lunar, and planetary gravity tend to move them away from their assigned orbit slots and orientations, an effect countered by on-board rocket motors. This fine-tuning activity is called station keeping. However, when the fuel for the motors has been exhausted (typically after about 10 years) the satellite drifts and tumbles helplessly, so it must be turned off. Eventually, the orbit decays and the satellite re-enters the atmosphere and burns up (or very rarely crashes to earth).

Orbit slots are not the only bone of contention. Frequencies are an issue, too, because the downlink transmissions interfere with existing microwave users. Consequently, ITU has allocated certain frequency bands to satellite

users. The main ones are listed in Fig. 2-16. The C band was the first to be designated for commercial satellite traffic. Two frequency ranges are assigned in it, the lower one for downlink traffic (from the satellite) and the upper one for uplink traffic (to the satellite). To allow traffic to go both ways at the same time, two channels are required. These channels are already overcrowded because they are also used by the common carriers for terrestrial microwave links. The L and S bands were added by international agreement in 2000. However, they are narrow and crowded.

## Communication Satellites (2)

The principal satellite bands.

| Band | Downlink | Uplink | Bandwidth | Problems |
|---|---|---|---|---|
| L | 1.5 GHz | 1.6 GHz | 15 MHz | Low bandwidth; crowded |
| S | 1.9 GHz | 2.2 GHz | 70 MHz | Low bandwidth; crowded |
| C | 4.0 GHz | 6.0 GHz | 500 MHz | Terrestrial interference |
| Ku | 11 GHz | 14 GHz | 500 MHz | Rain |
| Ka | 20 GHz | 30 GHz | 3500 MHz | Rain, equipment cost |

The next-highest band available to commercial telecommunication carriers is the Ku (K under) band. This band is not (yet) congested, and at its higher frequencies, satellites can be spaced as close as 1 degree. However, another problem exists: rain. Water absorbs these short microwaves well. Fortunately, heavy storms are usually localized, so using several widely separated ground stations instead of just one circumvents the problem, but at the price of extra antennas, extra cables, and extra electronics to enable rapid switching between stations. Bandwidth has also been allocated in the Ka (K above) band for commercial satellite traffic, but the equipment needed to use it is expensive. In addition to these commercial bands, many governments and military bands also exist.

A modern satellite has around 40 transponders, most often with a 36-MHz bandwidth. Usually, each transponder operates as a bent pipe, but recent satellites have some on-board processing capacity, allowing more sophisticated operation. In the earliest satellites, the division of the transponders into channels was static: the bandwidth was simply split up into fixed frequency bands. Nowadays, each transponder beam is divided into time slots, with various users taking turns. We will study these two techniques (frequency division multiplexing and time division multiplexing) in detail later in this chapter.

The first geostationary satellites had a single spatial beam that illuminated about 1/3 of the earth's surface, called its footprint. With the enormous decline in the price, size, and power requirements of microelectronics, a much more sophisticated broadcasting strategy has become possible. Each satellite is equipped with multiple antennas and multiple transponders. Each downward beam can be focused on a small geographical area, so multiple upward and downward transmissions can take place simultaneously. Typically, these so-called spot beams are elliptically shaped, and can be as small as a few hundred km in diameter. A communication satellite for the United States typically has one wide beam for the contiguous 48 states, plus spot beams for Alaska and Hawaii.

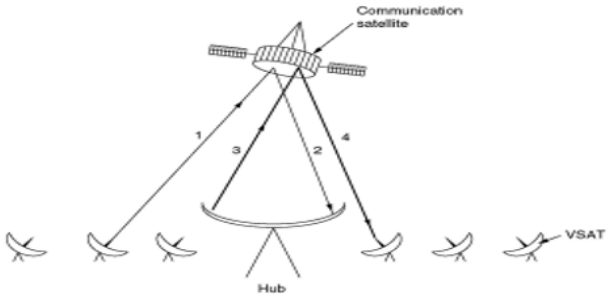
A recent development in the communication satellite world is the development of low-cost micro stations, sometimes called VSATs (Very Small Aperture Terminals) (Abramson, 2000). These tiny terminals have 1-meter or smaller antennas (versus 10 m for a standard GEO antenna) and can put out about 1 watt of power. The uplink is generally good for up to 1 Mbps, but the downlink is often up to several megabits/sec. Direct broadcast satellite television uses this technology for one-way transmission.

In many VSAT systems, the micro stations do not have enough power to communicate directly with one another (via the satellite, of course). Instead, a special ground station, the hub, with a large, high-gain antenna is needed to relay traffic between VSATs, as shown in Fig. 2-17. In this mode of operation, either the sender or the receiver has a large antenna and

a powerful amplifier. The trade-off is a longer delay in return for having cheaper end-user stations.

2- How long will it take to transmit a 1-GB file from one VSAT to another using a hub as shown in Figure below? Assume that the uplink is 1 Mbps, the downlink is 7 Mbps, and circuit switching is used with 1.2 sec circuit setup time.

## Communication Satellites (3)




VSATs using a hub.

VSATs have great potential in rural areas. It is not widely appreciated, but over half the world's population lives more than hour's walk from the nearest telephone. Stringing telephone wires to thousands of small villages is far beyond the budgets of most Third World governments but installing 1-meter VSAT dishes powered by solar cells is often feasible. VSATs provide the technology that will wire the world.

Communication satellites have several properties that are radically different from terrestrial point-to-point links. To begin with, even though signals to and from a satellite travel at the speed of light (nearly 300,000 km/sec), the long round-trip distance introduces a substantial delay for GEO satellites. Depending on the distance between the user and the ground station and the elevation of the satellite above the horizon, the end-to-end transit time is between 250 and 300 msec. A typical value is 270 msec (540 msec for a VSAT system with a hub).

For comparison purposes, terrestrial microwave links have a propagation delay of roughly 3 μsec/km, and coaxial cable or fibre optic links have a

delay of approximately 5 μsec/km. The latter are slower than the former because electromagnetic signals travel faster in air than in solid materials.

Another important property of satellites is that they are inherently broadcast media. It does not cost more to send a message to thousands of stations within a transponder's footprint than it does to send to one. For some applications, this property is very useful. For example, one could imagine a satellite broadcasting popular Web pages to the caches of a large number of computers spread over a wide area. Even when broadcasting can be simulated with point-to-point lines, satellite broadcasting may be much cheaper. On the other hand, from a privacy point of view, satellites are a complete disaster: everybody can hear everything. Encryption is essential when security is required.

Satellites also have the property that the cost of transmitting a message is independent of the distance traversed. A call across the ocean costs no more to service than a call across the street. Satellites also have excellent error rates and can be deployed almost instantly, a major consideration for disaster response and military communication.

## Medium-Earth Orbit Satellites

At much lower altitudes, between the two Van Allen belts, we find the MEO (Medium-Earth Orbit) satellites. As viewed from the earth, these drift slowly in longitude, taking something like 6 hours to circle the earth. Accordingly, they must be tracked as they move through the sky. Because they are lower than the GEOs, they have a smaller footprint on the ground and require less powerful transmitters to reach them. Currently they are used for navigation systems rather than telecommunications, so we will not examine them further here. The constellation of roughly 30 GPS (Global Positioning System) satellites orbiting at about 20,200 km are examples of MEO satellites.

## Low-Earth Orbit Satellites

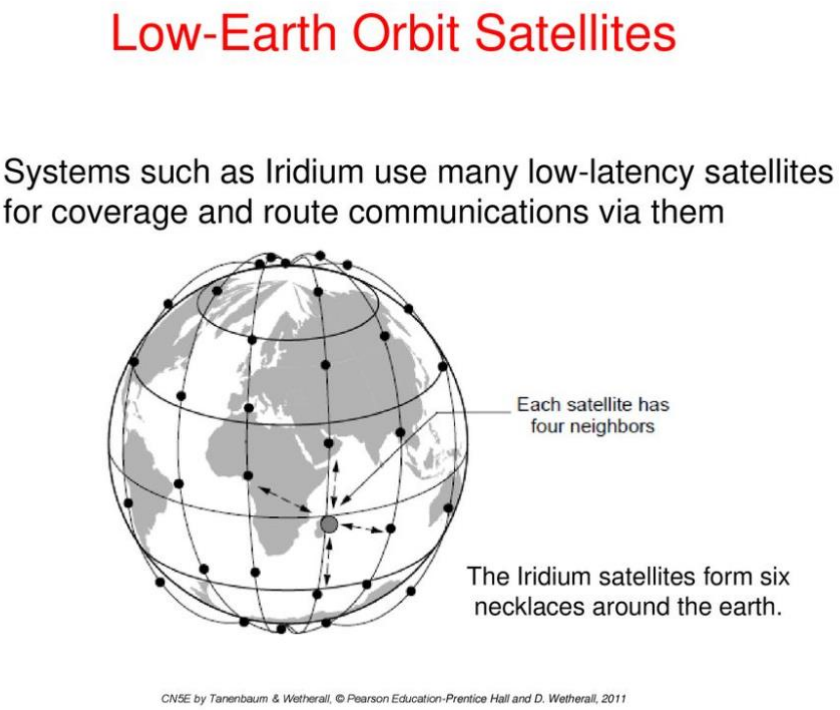
Moving down in altitude, we come to the LEO (Low-Earth Orbit) satellites. Due to their rapid motion, large numbers of them are needed for a complete system. On the other hand, because the satellites are so close to the earth, the ground stations do not need much power, and the round-trip delay is only a few milliseconds. The launch cost is substantially cheaper too. In this section we will examine two examples of satellite constellations for voice service, Iridium and Globalstar.

For the first 30 years of the satellite era, low-orbit satellites were rarely used because they zip into and out of view so quickly. In 1990, Motorola broke new ground by filing an application with the FCC asking for permission to launch 77 low-orbit satellites for the Iridium project (element 77 is iridium). The plan was later revised to use only 66 satellites, so the project should have been renamed Dysprosium (element 66), but that probably sounded too much like a disease. The idea was that as soon as one satellite went out of view, another would replace it. This proposal set off a feeding frenzy among other communication companies. All of a sudden, everyone wanted to launch a chain of low-orbit satellites.

After seven years of cobbling together partners and financing, communication service began in November 1998. Unfortunately, the commercial demand for large, heavy satellite telephones was negligible because the mobile phone network had grown in a spectacular way since 1990. Consequently, Iridium was not profitable and was forced into bankruptcy in August 1999 in one of the most spectacular corporate fiascos in history. The satellites and other assets (worth $5 billion) were later purchased by an investor for $25 million at a kind of extra-terrestrial garage sale. Other satellite business ventures promptly followed suit.

The Iridium service restarted in March 2001 and has been growing ever since. It provides voice, data, paging, fax, and navigation service everywhere on land, air, and sea, via hand-held devices that communicate directly with the Iridium satellites. Customers include the maritime, aviation, and oil exploration industries, as well as people traveling in parts of the world lacking a telecom infrastructure (e.g., deserts, mountains, the South Pole, and some Third World countries).

The Iridium satellites are positioned at an altitude of 750 km, in circular polar orbits. They are arranged in north-south necklaces, with one satellite every 32 degrees of latitude, as shown in Fig. 2-18. Each satellite has a maximum of 48 cells (spot beams) and a capacity of 3840 channels, some of which are used for paging and navigation, while others are used for data and voice.

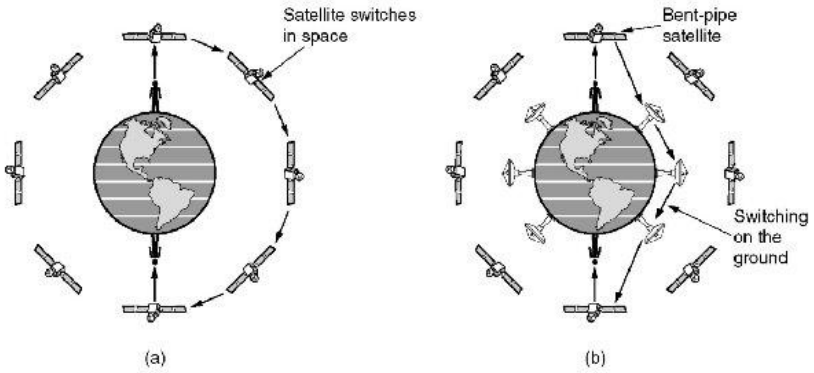



With six satellite necklaces the entire earth is covered, as suggested by Fig. 2-18. An interesting property of Iridium is that communication between distant customers takes place in space, as shown in Fig. 2-19(a). Here we see a caller at the North Pole contacting a satellite directly overhead. Each satellite has four neighbours with which it can communicate, two in the same necklace (shown) and two in adjacent necklaces (not shown). The satellites relay the call across this grid until it is finally sent down to the callee at the South Pole.

An alternative design to Iridium is Globalstar. It is based on 48 LEO satellites but uses a different switching scheme than that of Iridium. Whereas Iridium relays calls from satellite to satellite, which requires sophisticated switching equipment in the satellites, Globalstar uses a traditional bent-pipe design. The call originating at the North Pole in Fig.

2-19(b) is sent back to earth and picked up by the large ground station at Santa's Workshop. The call is then routed via a terrestrial network to the ground station nearest the callee and delivered by a bent-pipe connection as shown. The advantage of this scheme is that it puts much of the complexity on the ground, where it is easier to manage. Also, the use of large ground station antennas that can put out a powerful signal and receive a weak one means that lower-powered telephones can be used. After all, the telephone puts out only a few milliwatts of power, so the signal that gets back to the ground station is weak, even after having been amplified by the satellite.

## Globalstar




(a) Relaying in space.
(b) Relaying on the ground.

51

Satellites continue to be launched at a rate of around 20 per year, including ever-larger satellites that now weigh over 5000 kilograms. But there are also very small satellites for the more budget-conscious organization. To make space research more accessible, academics from Cal Poly and Stanford got together in 1999 to define a standard for miniature satellites and an associated launcher that would greatly lower launch costs (Nugent et al., 2008). CubeSats are satellites in units of 10 cm × 10 cm × 10 cm cubes, each weighing no more than 1 kilogram, that can be launched for as little as $40,000 each. The launcher flies as a secondary payload on commercial space missions. It is basically a tube that takes up to three units

of cubesats and uses springs to release them into orbit. Roughly 20 cubesats have launched so far, with many more in the works. Most of them communicate with ground stations on the UHF and VHF bands.

## Satellites Versus Fibre

A comparison between satellite communication and terrestrial communication is instructive. As recently as 25 years ago, a case could be made that the future of communication lay with communication satellites. After all, the telephone system had changed little in the previous 100 years and showed no signs of changing in the next 100 years. This glacial movement was caused in no small part by the regulatory environment in which the telephone companies were expected to provide good voice service at reasonable prices (which they did), and in return got a guaranteed profit on their investment. For people with data to transmit, 1200-bps modems were available. That was pretty much all there was.

The introduction of competition in 1984 in the United States and somewhat later in Europe changed all that radically. Telephone companies began replacing their long-haul networks with fibre and introduced high-bandwidth services like ADSL (Asymmetric Digital Subscriber Line). They also stopped their long-time practice of charging artificially high prices to long-distance users to subsidize local service. Suddenly, terrestrial fibre connections looked like the winner.

Nevertheless, communication satellites have some major niche markets that fibre does not (and, sometimes, cannot) address. First, when rapid deployment is critical, satellites win easily. A quick response is useful for military communication systems in times of war and disaster response in times of peace. Following the massive December 2004 Sumatra earthquake and subsequent tsunami, for example, communications satellites were able to restore communications to first responders within 24 hours. This rapid response was possible because there is a developed satellite service provider market in which large players, such as Intelsat with over 50 satellites, can rent out capacity pretty much anywhere it is needed. For customers served by existing satellite networks, a VSAT can be set up easily and quickly to provide a megabit/sec link to elsewhere in the world.

A second niche is for communication in places where the terrestrial infrastructure is poorly developed. Many people nowadays want to communicate everywhere they go. Mobile phone networks cover those locations with good population density, but do not do an adequate job in other places (e.g., at sea or in the desert). Conversely, Iridium provides voice service everywhere on Earth, even at the South Pole. Terrestrial infrastructure can also be expensive to install, depending on the terrain and necessary rights of way. Indonesia, for example, has its own satellite for domestic telephone traffic. Launching one satellite was cheaper than stringing thousands of undersea cables among the 13,677 islands in the archipelago.

A third niche is when broadcasting is essential. A message sent by satellite can be received by thousands of ground stations at once. Satellites are used to distribute much network TV programming to local stations for this reason. There is now a large market for satellite broadcasts of digital TV and radio directly to end users with satellite receivers in their homes and cars. All sorts of other content can be broadcast too. For example, an organization transmitting a stream of stock, bond, or commodity prices to thousands of dealers might find a satellite system to be much cheaper than simulating broadcasting on the ground.

In short, it looks like the mainstream communication of the future will be terrestrial fibre optics combined with cellular radio, but for some specialized uses, satellites are better. However, there is one caveat that applies to all of this: economics. Although fibre offers more bandwidth, it is conceivable that terrestrial and satellite communication could compete aggressively on price. If advances in technology radically cut the cost of deploying a satellite (e.g., if some future space vehicle can toss out dozens of satellites on one launch) or low-orbit satellites catch on in a big way, it is not certain that fibre will win all markets.

## 1.9 Performance Indicators

Up to this point, we have focused primarily on the functional aspects of networks. Like any computer system, however, computer networks are also

expected to perform well. This is because the effectiveness of computations distributed over the network often depends directly on the efficiency with which the network delivers the computation's data. While the old programming adage "1st get it right & then make it fast" remains true, in networking it is often necessary to "design for performance." It is therefore important to understand the various factors that impact network performance.

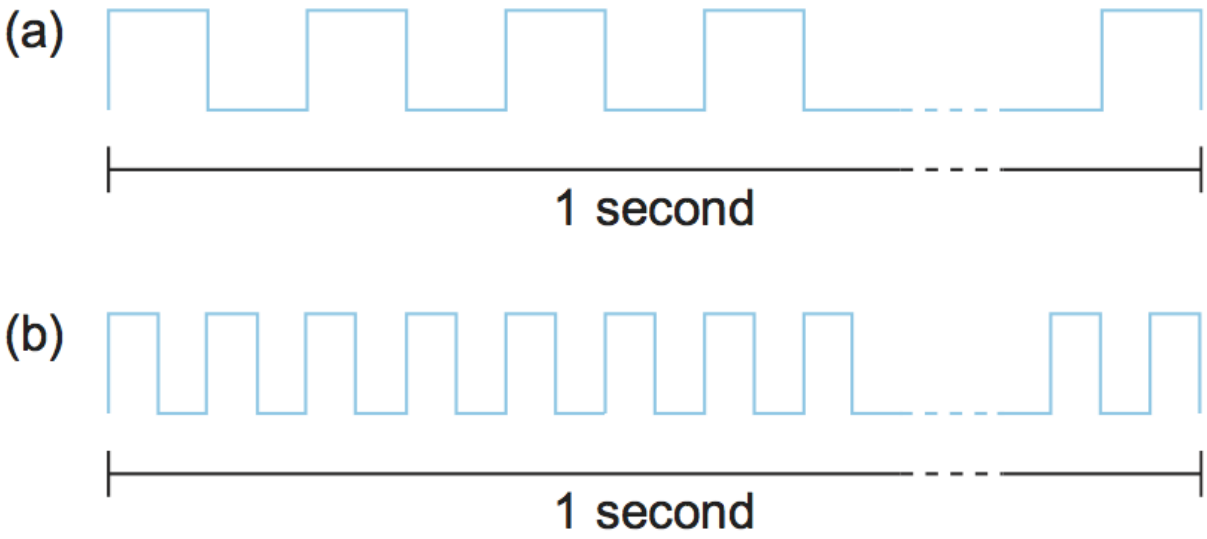
### 1.9.1 Bandwidth

Network performance is measured in 2 fundamental ways: *bandwidth* (also called *throughput*) & *latency* (also called *delay*). The bandwidth of a network is given by the number of bits that can be transmitted over the network in a certain period. For example, a network might have a bandwidth of 10 million bits/second (Mbps), meaning that it is able to deliver 10 million bits every second. It is sometimes useful to think of bandwidth in terms of how long it takes to transmit each bit of data. On a 10-Mbps network, for example, it takes 0.1 microsecond (μs) to transmit each bit.

Bandwidth & throughput are subtly different terms. First, bandwidth is literally a measure of the width of a frequency band. For example, legacy voice-grade telephone lines supported a frequency band ranging from 300 to 3300 Hz; it was said to have a bandwidth of 3300 Hz - 300 Hz = 3000 Hz. If you see the word *bandwidth* used in a situation in which it is being measured in hertz, then it probably refers to the range of signals that can be accommodated.

When we talk about the bandwidth of a communication link, we normally refer to the number of bits per second that can be transmitted on the link. This is also sometimes called the *data rate*. We might say that the bandwidth of an Ethernet link is 10 Mbps. A useful distinction can also be made, however, between the maximum data rate that is available on the link & the number of bits per second that we can transmit over the link in practice.

While you can talk about the bandwidth of the network, sometimes you want to be more precise, focusing, for example, on the bandwidth of a single physical link or of a logical process-to-process channel. At the physical level, bandwidth is constantly improving, with no end in sight. Intuitively, if you think of a second of time as a distance you could measure with a ruler & bandwidth as how many bits fit in that distance, then you can think of each bit as a pulse of some width. For example, each bit on a 1-Mbps link is 1 μs wide, while each bit on a 2-Mbps link is 0.5 μs wide, as illustrated in the figure. The more sophisticated the transmitting & receiving technology, the narrower each bit can become &, thus, the higher the bandwidth. For logical process-to-process channels, bandwidth is also influenced by other factors, including how many times the software that implements the channel must handle, & possibly transform, each bit of data.




In terms of Hertz, **bandwidth** is the width of a frequency band; The width is the highest frequency minus the lowest frequency. In the hearing example, the bandwidth of a person's ears is about 20,000 Hz - 20 Hz = 19,980 Hz.

### 1.9.2 Throughput

We tend to use the word *throughput* to refer to the *measured performance* of a system. Thus, because of various inefficiencies of implementation, a pair of nodes connected by a link with a bandwidth of

10 Mbps might achieve a throughput of only 2 Mbps. This would mean that an application on one host could send data to the other host at 2 Mbps. Finally, we often talk about the bandwidth *requirements* of an application. This is the number of bits per second that it needs to transmit over the network to perform acceptably. For some applications, this might be "whatever I can get"; for others, it might be some fixed number (preferably not more than the available link bandwidth); & for others, it might be a number that varies with time.

### 1.9.3 Latency

Next performance metric, latency, corresponds to how long it takes a message to travel from 1 end of a network to the other. (As with bandwidth, we could be focused on the latency of a single link or an end-to-end channel.) Latency is measured strictly in terms of time. For example, a transcontinental network might have a latency of 24 milliseconds (ms); that is, it takes a message 24 ms to travel from one coast of North America to the other. There are many situations in which it is more important to know how long it takes to send a message from one end of a network to the other & back, rather than the one-way latency. We call this the *round-trip time* (RTT) of the network.

We often think of latency as having 3 components.1st, there is the speed-of-light propagation delay. This delay occurs because nothing, including a bit on a wire, can travel faster than the speed of light. If you know the distance between 2 points, you can calculate the speed-of-light latency, although you have to be careful because light travels across different media at different speeds: It travels at $3.0 \times 10^8$ m/s in a vacuum, $2.3 \times 10^8$ m/s in a copper cable, & $2.0 \times 10^8$ m/s in an optical fibre. 2nd, there is the amount of time it takes to transmit a unit of data. This is a function of the network bandwidth & the size of the packet in which the data is carried. 3rd, there may be queuing delays inside the network, since packet switches generally need to store packets for some time before forwarding them on an outbound link. So, we could define the total latency as

Latency = Propagation + Transmit + Queue
Propagation = Distance/Speed of Light

Transmit = Size/Bandwidth

where Distance is the length of the wire over which the data will travel, Speed of Light is the effective speed of light over that wire, Size is the size of the packet, & Bandwidth is the bandwidth at which the packet is transmitted. Note that if the message contains only one bit & we are talking about a single link (as opposed to a whole network), then the Transmit & Queue terms are not relevant, & latency corresponds to the propagation delay only.

Bandwidth & latency combine to define the performance characteristics of a given link or channel. Their relative importance, however, depends on the application. For some applications, latency dominates bandwidth. For example, a client that sends a 1-byte message to a server and receives a 1-byte message in return is latency bound. Assuming that no serious computation is involved in preparing the response, the application will perform much differently on a transcontinental channel with a 100-ms RTT than it will on an across-the-room channel with a 1-ms RTT. Whether the channel is 1 Mbps or 100 Mbps is relatively insignificant, however, since the former implies that the time to transmit a byte (Transmit) is 8 μs and the latter implies Transmit = 0.08 μs.

In contrast, consider a digital library program that is being asked to fetch a 25-megabyte (MB) image—the more bandwidth that is available, the faster it will be able to return the image to the user. Here, the bandwidth of the channel dominates performance. To see this, suppose that the channel has a bandwidth of 10 Mbps. It will take 20 seconds to transmit the image ($25 \times 10^6 \times$ 8-bits / ($10 \times 10^6$ Mbps = 20 seconds), making it relatively unimportant if the image is on the other side of a 1-ms channel or a 100-ms channel; the difference between a 20.001-second response time and a 20.1-second response time is negligible.

### 1.9.4 Queuing Time

In telecommunication & computer engineering, the **queuing delay** or **queueing delay** is the time a job waits in a queue until it can be executed. It is a key component of network delay. In a switched network,

queuing delay is the time between the completion of signaling by the call originator & the arrival of a ringing signal at the call receiver. Queuing delay may be caused by delays at the originating switch, intermediate switches, or the call receiver servicing switch. In a data network, queuing delay is the sum of the delays between the request for service and the establishment of a circuit to the called data terminal equipment (DTE). In a packet-switched network, queuing delay is the sum of the delays encountered by a packet between the time of insertion into the network and the time of delivery to the address. [1]

This term is most often used in reference to routers. When packets arrive at a router, they have to be processed and transmitted. A router can only process one packet at a time. If packets arrive faster than the router can process them (such as in a burst transmission) the router puts them into the queue (also called the buffer) until it can get around to transmitting them. Delay can also vary from packet to packet so averages and statistics are usually generated when measuring and evaluating queuing delay. [2]

As a queue begins to fill up due to traffic arriving faster than it can be processed, the amount of delay a packet experiences going through the queue increases. The speed at which the contents of a queue can be processed is a function of the transmission rate of the facility. This leads to the classic delay curve. The average delay any given packet is likely to experience is given by the formula $1/(\mu-\lambda)$ where $\mu$ is the number of packets per second the facility can sustain and $\lambda$ is the average rate at which packets are arriving to be serviced. [3] This formula can be used when no packets are dropped from the queue.

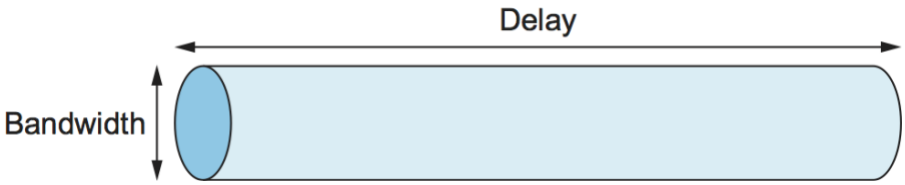
The maximum queuing delay is proportional to buffer size. The longer the line of packets waiting to be transmitted, the longer the average waiting time is. The router queue of packets waiting to be sent also introduces a potential cause of packet loss. Since the router has a finite amount of buffer memory to hold the queue, a router which receives packets at too high a rate may experience a full queue. In this case, the router has no other option than to simply discard excess packets.

When the transmission protocol uses the dropped-packets symptom of filled buffers to regulate its transmit rate, as the Internet's TCP does, bandwidth is fairly shared at near theoretical capacity with minimal network congestion delays. Absent this feedback mechanism the delays become both unpredictable and rise sharply, a symptom also seen

as freeways approach capacity; metered onramps are the most effective solution there, just as TCP's self-regulation is the most effective solution when the traffic is packets instead of cars). This result is both hard to model mathematically and quite counterintuitive to people who lack experience with mathematics or real networks. Failing to drop packets, choosing instead to buffer an ever-increasing number of them, produces bufferbloat. In Kendall's notation, the M/M/1/K queuing model, where K is the size of the buffer, may be used to analyze the queuing delay in a specific system. Kendall's notation should be used to calculate the queuing delay when packets are dropped from the queue. The M/M/1/K queuing model is the most basic and important queuing model for network analysis

### 1.9.5 Bandwidth – Delay product

It is also useful to talk about the product of these 2 metrics, often called the *delay × bandwidth product*. Intuitively, if we think of a channel between a pair of processes as a hollow pipe, where the latency corresponds to the length of the pipe & the bandwidth gives the diameter of the pipe, then the delay × bandwidth product gives the volume of the pipe—the maximum number of bits that could be in transit through the pipe at any given instant. Said another way, if latency (measured in time) corresponds to the length of the pipe, then given the width of each bit (also measured in time) you can calculate how many bits fit in the pipe. For example, a transcontinental channel with a one-way latency of 50 ms and a bandwidth of 45 Mbps can hold $\mathbf{50 \times 10^{-3} \times 45 \times 10^{6}}$ ***bits/sec*** $\mathbf{= 2.25 \times 10^{6}}$ ***bits*** or approximately 280 KB of data. In other words, this example channel (pipe) holds as many bytes as the memory of a personal computer from the early 1980s could hold.




The delay × bandwidth product is important to know when constructing high-performance networks because it corresponds to how many bits the

sender must transmit before the first bit arrives at the receiver. If the sender is expecting the receiver to somehow signal that bit are starting to arrive, & it takes another channel latency for this signal to propagate back to the sender, then the sender can send up one *RTT* × *bandwidth* worth of data before hearing from the receiver that all is well. The bits in the pipe are said to be "in flight," which means that if the receiver tells the sender to stop transmitting it might receive up to one RTT × bandwidth's worth of data before the sender manages to respond. In our example above, that amount corresponds to $5.5 \times 10^6$ bits (671 KB) of data. On the other hand, if the sender does not fill the pipe—i.e., doesn't send a whole RTT × bandwidth product's worth of data before it stops to wait for a signal—the sender will not fully utilize the network.

Most of the time we are interested in the RTT scenario, which we simply refer to as the delay × bandwidth product, without explicitly saying that "delay" is the RTT (i.e., multiply the one-way delay by two). Usually, whether the "delay" in delay × bandwidth means one-way latency or RTT is made clear by the context. The below table shows some examples of RTT × bandwidth products for some typical network links.

## High-Speed Networks

| Link Type | Bandwidth | Distance | RTT | Delay x BW |
|---|---|---|---|---|
| Dial-up | 56 kbps | 10 km | 87 µs | 5 bits |
| Wireless LAN | 54 Mbps | 50 m | 0.33 µs | 18 bits |
| Satellite link | 45 Mbps | 35,000 km | 230 ms | 10 Mb |
| Cross-country fiber | 10 Gbps | 4,000 km | 40 ms | 400 Mb |

Infinite bandwidth

- Propagation delay dominates
- Throughput = Transfer size/Transfer time
- Transfer time = RTT + Transfer size/Bandwidth
- 1MB file across 1Gbps line with 100ms RTT, Throughput is 74.1 Mbps

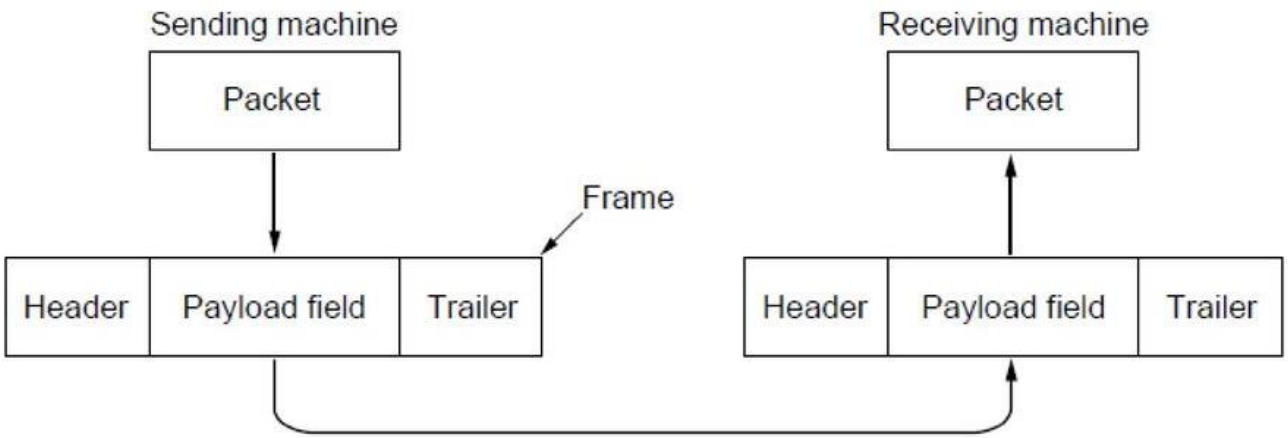
# Module - 2 (Data Link Layer)

Data link layer - Data link layer design issues, Error detection and correction, Sliding window protocols, High-Level Data Link Control (HDLC)protocol. Medium Access Control (MAC) sublayer –Channel allocation problem, Multiple access protocols, Ethernet, Wireless LANs - 802.11, Bridges & switches - Bridges from 802.x to 802.y, Repeaters, Hubs, Bridges, Switches, Routers and Gateways.

# Data Link layer

- $2^{nd}$ layer after physical layer
- Responsible for maintaining the data link between 2 hosts or nodes
- Protocol layer that transfers data between nodes on a network segment across the physical layer
- Provides functional & procedural means to transfer data between network entities
- Provide means to detect & possibly correct errors that can occur in the physical layer
- Data link layer uses the services of the physical layer to send & receive bits over communication channels
- Has several functions, including
  - Providing a well-defined service interface to the network layer
  - Dealing with transmission errors
  - Regulating the flow of data so that slow receivers are not swamped by fast senders

- To accomplish these goals, the layer takes the packets it gets from the network layer & encapsulates them into frames for transmission




Relationship between packets and frames.

- Each frame contains (Refer the above figure for **Frame Structure**)
  - a frame header
  - a payload field for holding the packet
  - a frame trailer
- Frame management forms the heart of what the data link layer does

# Data Link Layer Design Issues

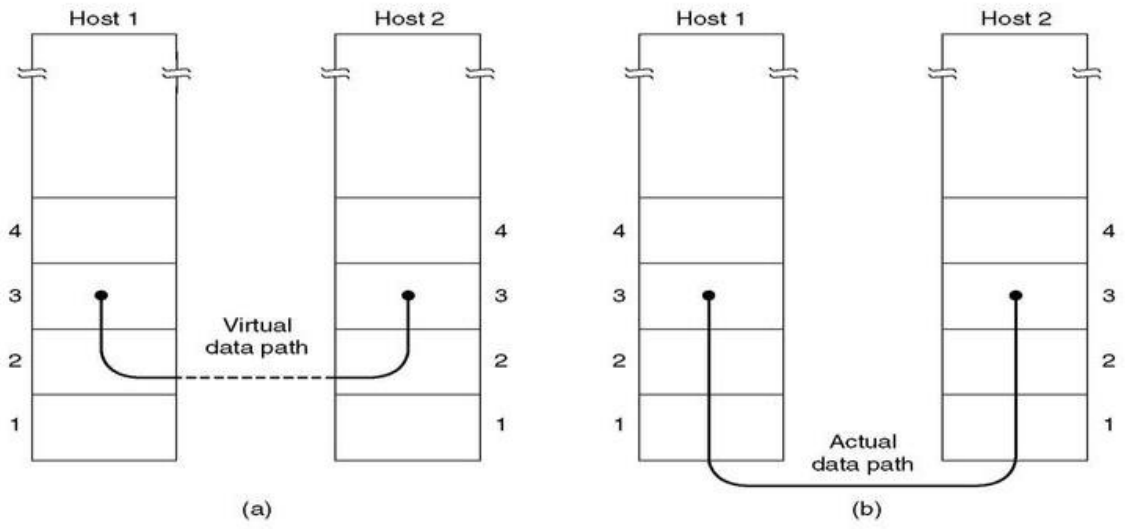
- Design issues with data link layer are:
  - **Services provided to the network layer**
    - Act as a service interface to the network layer
    - Transfer data from network layer on sending machine to the network layer on destination machine
    - This transfer also takes place via DLL (Dynamic Link Library)
  - **Framing**
    - Source machine sends data in the form of blocks called frames to the destination machine
    - Starting & ending of each frame should be identified so that the frame can be recognized by the destination machine.
  - **Error control**

- Done to prevent duplication of frames
- Errors introduced during transmission from source to destination machines must be detected & corrected at the destination machine

- **Flow control**
  - Done to prevent the flow of data frame at the receiver end
  - Source machine must not send data frames at a rate faster than the capacity of destination machine to accept them

## Services Provided to the Network Layer

- The principal service is to transfer data from the network layer on the source machine to the network layer on the destination machine
- Source machine has an entity (a process), in the network layer that hands some bits to the data link layer for transmission to destination
- The job of data link layer is to transmit these bits to the destination machine so they can be handed over to the network layer there
- The actual transmission follows the path shown in the figure, but it's easier to think in terms of 2 data link layer processes communicating using a data link protocol




(a) Virtual communication. (b) Actual communication.

- Data link layer can be designed to offer various services • The actual services that are offered vary from protocol to protocol.

- 3 reasonable possibilities (in terms of reliability) that we will consider in turn are:

1. Unacknowledged connectionless service
   - Source machine sends independent frames to the destination machine without having the destination machine acknowledge them
   - Ethernet is a good example
   - No logical connection is established beforehand or released afterward
   - No attempt is made to detect the loss or recover it, if a frame is lost due to noise on the line
   - Appropriate when error rate is very low, since, recovery is left to higher layers
   - Appropriate for real-time traffic, like voice, in which late data are worse than bad data.
2. Acknowledged connectionless service
   - No logical connections used
   - Each frame sent is individually acknowledged
   - Sender knows whether a frame has arrived correctly or been lost
   - Can be sent again if not arrived within a specified time interval
   - Useful over unreliable channels, like wireless systems
   - 802.11 (Wi-Fi) is a good example
   - Lost acknowledgements could cause a frame to be sent & received several times, wasting bandwidth
3. Acknowledged Connection – Oriented Service
   - The most sophisticated service the data link layer can provide to the network layer
   - Source & destination machines establish a connection before any data are transferred
   - Each frame sent over the connection is numbered & also guarantees that each frame sent is received
   - Guarantees that each frame is received exactly once & that all frames are received in the right order
   - Provides the network layer processes with the equivalent of a reliable bit stream
   - Appropriate over long, unreliable links
   - Satellite channel or a long-distance telephone circuit are good examples
   - Transfers go through 3 distinct phases:

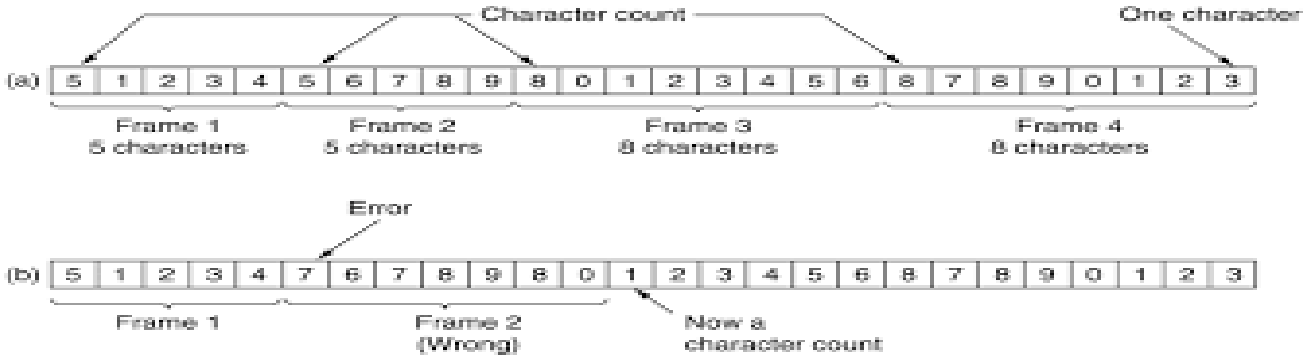
- Connection is established by having both sides initialize variables & counters needed to keep track of which frames have been received & which ones have not.
- 1 or more frames are transmitted
- Connection is released, freeing up the variables, buffers & other resources used to maintain the connection

## Framing

- The bit stream received by the data link layer is not guaranteed to be error free
- Some bits may have different values & the number of bits received may be the number of bits transmitted
- The usual approach to detect & correct errors is to break up the bit stream into discrete frames, compute a short token called a checksum for each frame, & include the checksum in the frame when it is transmitted.
- When a frame arrives at the destination, the checksum is recomputed
- If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred & takes steps to deal with it by discarding the bad frame or sending back an error report
- 4 methods of breaking up the bit stream into frames include:

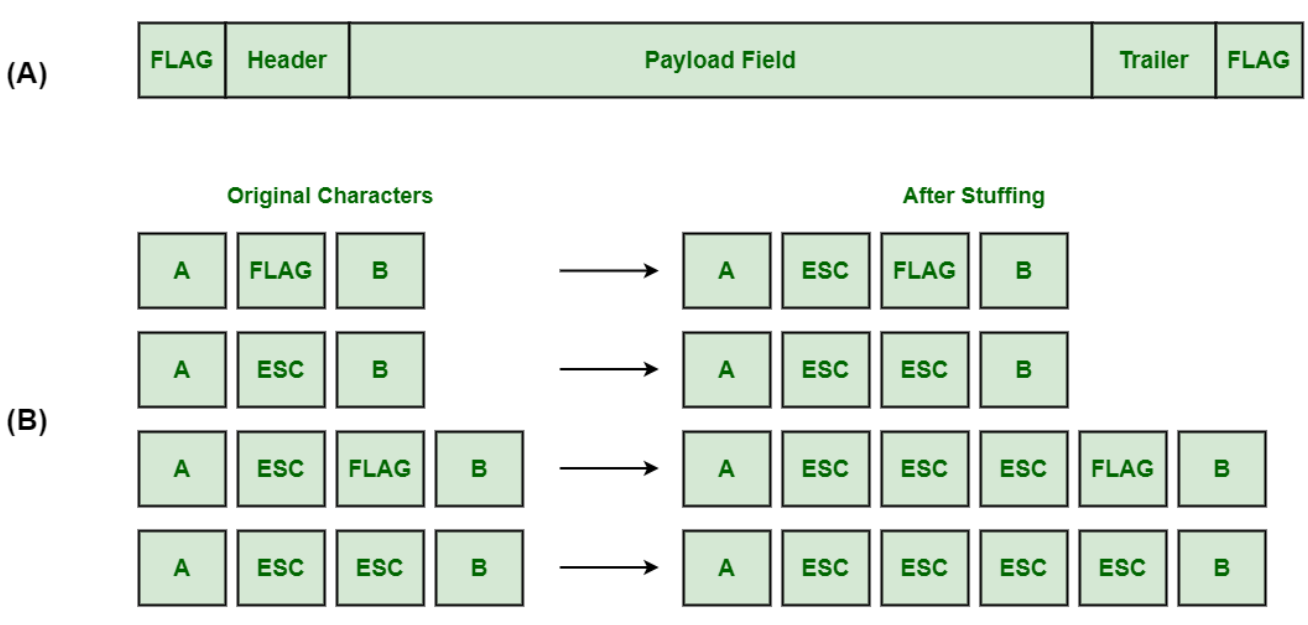
### 1) Byte count

- Uses a field in the header to specify the number of bytes in the frame
- When destination layer data link layer sees the byte count, it knows how many bytes follow & hence know where the end of the frame is
- 4 small example frames of sizes 5, 5, 8, & 8 bytes, respectively are shown in the figure
- The count can be garbled by a transmission error
- If the byte count of 5 in the second frame becomes a 7 due to a single bit flip, the destination will get out of synchronization
- It will then be unable to locate the correct start of the next frame
    - The byte count method is rarely used by itself



2) **Flag bytes with byte stuffing**

- Gets around with the problem of resynchronization after an error by having each frame start & end with special bytes
- Same byte, called a flag byte, is used as both the starting & ending delimiter
- 2 consecutive flag bytes indicate the end of 1 frame & the start of the next
- If receiver loses synchronization, it can just search for 2 flag bytes to find the end of the current frame & the start of the next frame
- But it may happen that the flag byte occurs in the data, especially when binary data such as photographs or songs are being transmitted & this situation would interfere with the framing
- To solve this problem, we can insert a special escape byte (ESC) just before each "accidental" flag byte in the data in the sender's data link layer & so, a framing flag byte can be distinguished from the one in the data by the absence or presence of an escape byte before it.
- Data link layer on the receiving end removes the escape bytes before giving the data to the network layer & this technique is called byte stuffing
- If an escape byte occurs in the middle of the data, it is also stuffed with an escape byte
- At the receiver, the 1st escape byte is removed, leaving the data byte that follows it




3) **Flag Bits with Bit Stuffing**

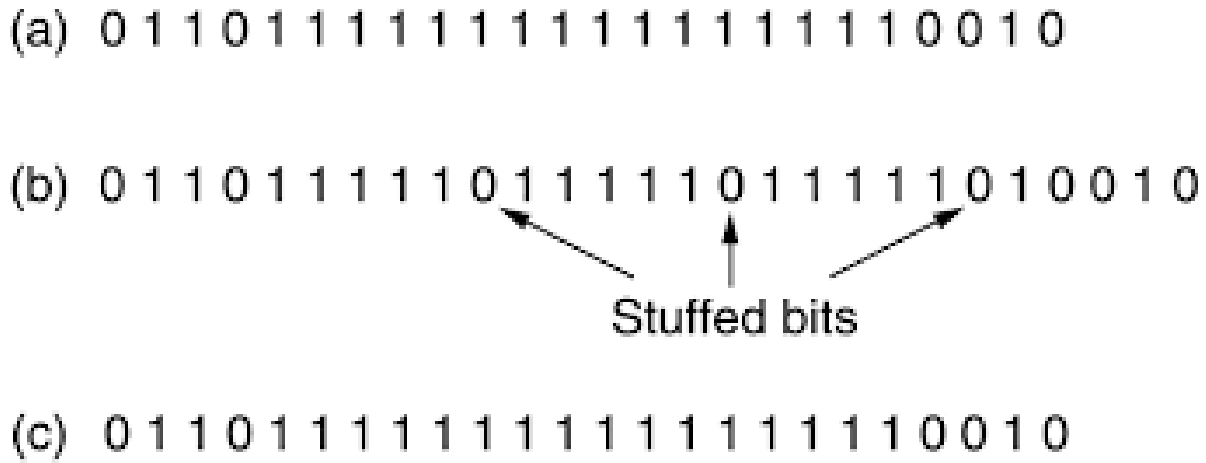
- Gets around the disadvantage of byte stuffing, i.e., it is tied to the use of 8-bit bytes
- Framing can be done at the bit level, so frames can contain an arbitrary number of bits made up of units of any size
- Developed for the once very popular HDLC (High-level Data Link Control) protocol
- Each frame begins & ends with a special bit pattern, 01111110 or 0x7E in hexadecimal. This pattern is a flag byte
- Whenever the sender's data link layer encounters 5 consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data
- If particular data of flag comes in the data field, we can't distinguish between flag & user information. To prevent this, after 5 consecutive 1's an additional 0 is inserted.

E.g.:

Flag: 01111110
User Information: 01111110
Updated User Information: 011111010
So, a redundant 0 is inserted in between, & is called zero stuffing, or bit stuffing. When it comes to receiver side, that particular 0 is deleted.

- Also ensures a minimum density of transitions that help the physical layer maintain synchronization. USB (Universal Serial Bus) uses bit stuffing for this reason
- When the receiver sees 5 consecutive incoming 1 bit, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.
- Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing.
- If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.The given figure gives an example of bit stuffing.
- With bit stuffing, the boundary between 2 frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it must do is scan the input for flag sequences, since they can only occur at frame boundaries & never within the data.

With both bit & byte stuffing, a side effect is that the length of a frame now depends on the contents of the data it carries. For instance, if there are no flag bytes in the data, 100 bytes might be carried in a frame of roughly 100 bytes. If, however, the data consists solely of flag bytes, each flag byte will be escaped, & the frame will become roughly 200 bytes long. With bit stuffing, the increase would be roughly 12.5% as 1 bit is added to every bite.




4) **<u>Physical layer coding violations</u>**
   - Last method of framing
   - Use a shortcut from the physical layer
   - The encoding of bits as signals often includes redundancy to help the receiver
   - This redundancy means that some signals won't occur in regular data
   - For example, in the 4B/5B line code 4 data bits are mapped to 5 signal bits to ensure sufficient bit transitions. This means that 16 out of the 32 signal possibilities are not used.
   - We can use some reserved signals to indicate the start & end of frames.
   - In effect, we use "coding violations" to delimit frames.
   - The beauty of this scheme is that, because they are reserved signals, it is easy to find the start & end of frames and there is no need to stuff the data.

Many data link protocols use a combination of these methods for safety. A common pattern used for Ethernet & 802.11 is to have a frame begin with a well-defined pattern called a preamble. This pattern might be quite long (72 bits is typical for 802.11) to allow the receiver to prepare for an incoming packet. The

preamble is then followed by a length (i.e., count) field in the header that is used to locate the end of the frame.

## Error Control

- Make sure all frames are eventually delivered to the network layer at the destination & in the proper order
- Consider a case where the receiver can tell whether a frame that it receives contains correct or faulty information
- For unacknowledged connectionless service, it might be fine if the sender just kept outputting frames without regard to whether they were arriving properly
- For reliable, connection-oriented service it wouldn't be fine at all
- A usual way is to provide the sender with some feedback about what is happening at the other end of the line.
    - The protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames.
    - If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely
    - A negative acknowledgement means that something has gone wrong, & the frame must be transmitted again
    - Complication comes from the possibility that hardware troubles may cause a frame to vanish completely (e.g., in a noise burst). In this case, the receiver will not react at all, since it has no reason to react.
    - Also, if the acknowledgement frame is lost, the sender will not know how to proceed. It should be clear that a protocol in which the sender transmits a frame & then waits for an acknowledgement, positive or negative, will hang forever if a frame is ever lost due to, for example, malfunctioning hardware or a faulty communication channel.
- The above issue is dealt by introducing timers into the data link layer.
    - When the sender transmits a frame, it also starts a timer
    - The timer is set to expire after an interval long enough for the frame to reach the destination, be processed there, & have the acknowledgement propagate back to the sender. Normally, the frame will be correctly received, & the acknowledgement will get back before the timer runs out, in which case the timer will be cancelled.

    - But, if either the frame or the acknowledgement is lost, the timer will go off, alerting the sender to a potential problem
- The solution for above issue is to just transmit the frame again
    - But when frames may be transmitted multiple times there is a danger that the receiver will accept the same frame 2 or more times & pass it to the network layer more than once
- To prevent this from happening, it is generally necessary to assign sequence numbers to outgoing frames, so that the receiver can distinguish retransmissions from originals
- The whole issue of managing the timers & sequence numbers to ensure that each frame is ultimately passed to the network layer at the destination exactly once, no more & no less, is an important part of the duties of the data link layer (& higher layers).

## Flow Control

- Deals with what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them
- This situation can occur when the sender is running on a fast, powerful computer & the receiver is running on a slow, low-end machine
- A common situation is when a smart phone requests a Web page from a far more powerful server, which then turns on the fire hose & blasts the data at the poor helpless phone until it is completely swamped
- Even if the transmission is error free, the receiver may be unable to handle the frames as fast as they arrive & will lose some.
- Something must be done to prevent this situation
- 2 approaches are commonly used:
    1. **Feedback-based flow control**
        - The receiver sends back information to the sender giving it permission to send more data;
        - Or tell the sender how the receiver is doing o Seen at both the link layer & higher layers
    2. **Rate-based flow control**
        - A built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver
        - Only seen as part of the transport layer
        - More common these days, in which case the link layer hardware is designed to run fast enough that it doesn't cause loss.

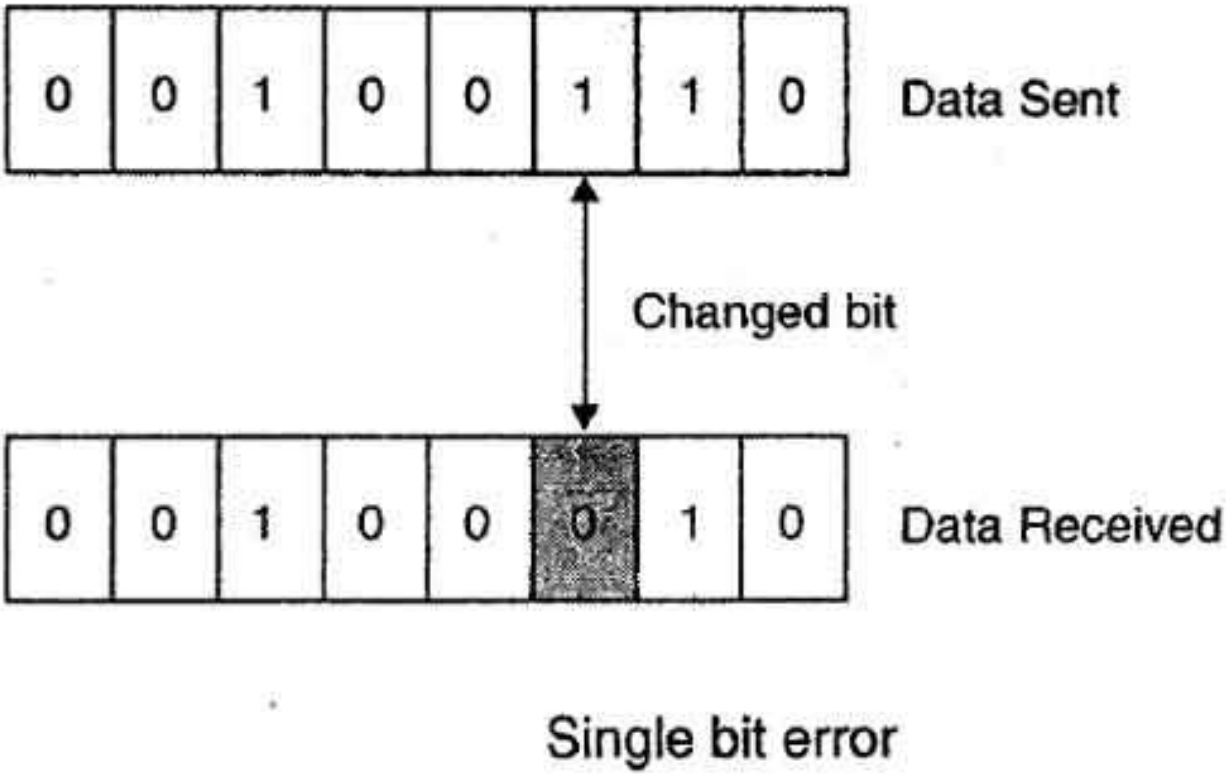
# Error detection & correction

- Channels like optical fibre have tiny error rates so that transmission errors are a rare occurrence
- But other channels, especially wireless links & aging local loops, have error rates that are orders of magnitude larger, that can't be avoided at a reasonable expense or cost in terms of performance
- There are mainly 2 basic strategies to deal with errors. Both add redundant information to the data that is sent.

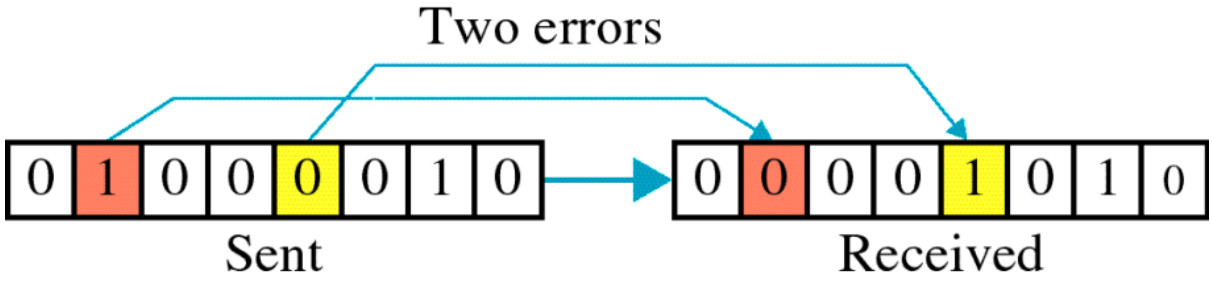
1. **Error Correcting Codes**
   - Include enough redundant information to enable the receiver to deduce what the transmitted data must have been
   - Referred to as FEC (Forward Error Correction)
   - On channels such as wireless links that make many errors, it is better to add redundancy to each block so that the receiver can figure out what the originally transmitted block was
   - Used on noisy channels because retransmissions are just as likely to be in error as the 1st transmission o Seen in physical layer, particularly for noisy channels
   - Seen in higher layers, for real-time media & content distribution
2. **Error Detecting Codes**
   - Include only enough redundancy to allow the receiver to deduce that an error has occurred (but not which error) & have it request a retransmission
   - On channels that are highly reliable, such as fibre, it is cheaper to use an error-detecting code & just retransmit the occasional block found to be faulty.
   - Commonly used in link, network, & transport layers
   - Neither error-correcting codes nor error-detecting codes can handle all possible errors since the redundant bits that offer protection are as likely to be received in error as the data bits (which can compromise their protection)
   - To avoid undetected errors the code must be strong enough to handle the expected errors.
   - There may be 3 types of errors:

   1. **Single bit error**
      - In a frame, there is only 1 bit, anywhere though, which is corrupt

- Caused by extreme values of thermal noise

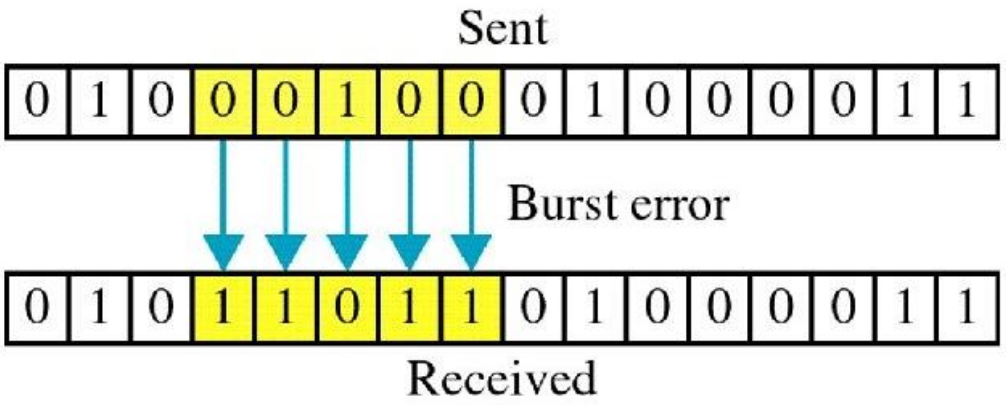



2. **Multiple bits error**
   - Frame is received with more than 1 bits in corrupted state.




3. **Burst error**
   - Frame contain more than 1 consecutive bits corrupted.
   - Caused from the physical processes that generate them, like a deep fade on a wireless channel or transient electrical interference on a wired channel
   - Merit of burst error over single bit error is that computer data are always sent in blocks of bits; suppose that the block size was 1000 bits, & the error rate was 0.001 per bit; If the errors came in bursts of 100, only 1 block in 100 would be affected, on average.

- But it is much harder to correct burst errors than isolated single bit errors




**Erasure Channel**

Sometimes, the location of an error will be known, perhaps because the physical layer received an analogue signal that was far from the expected value for a 0 or 1 & declared the bit to be lost. This situation is called an erasure channel.

Easier to correct errors in erasure channels than in channels that flip bits because even if the value of the bit has been lost, at least we know which bit is in error

- Error codes are applied mathematics

# Error correcting Codes

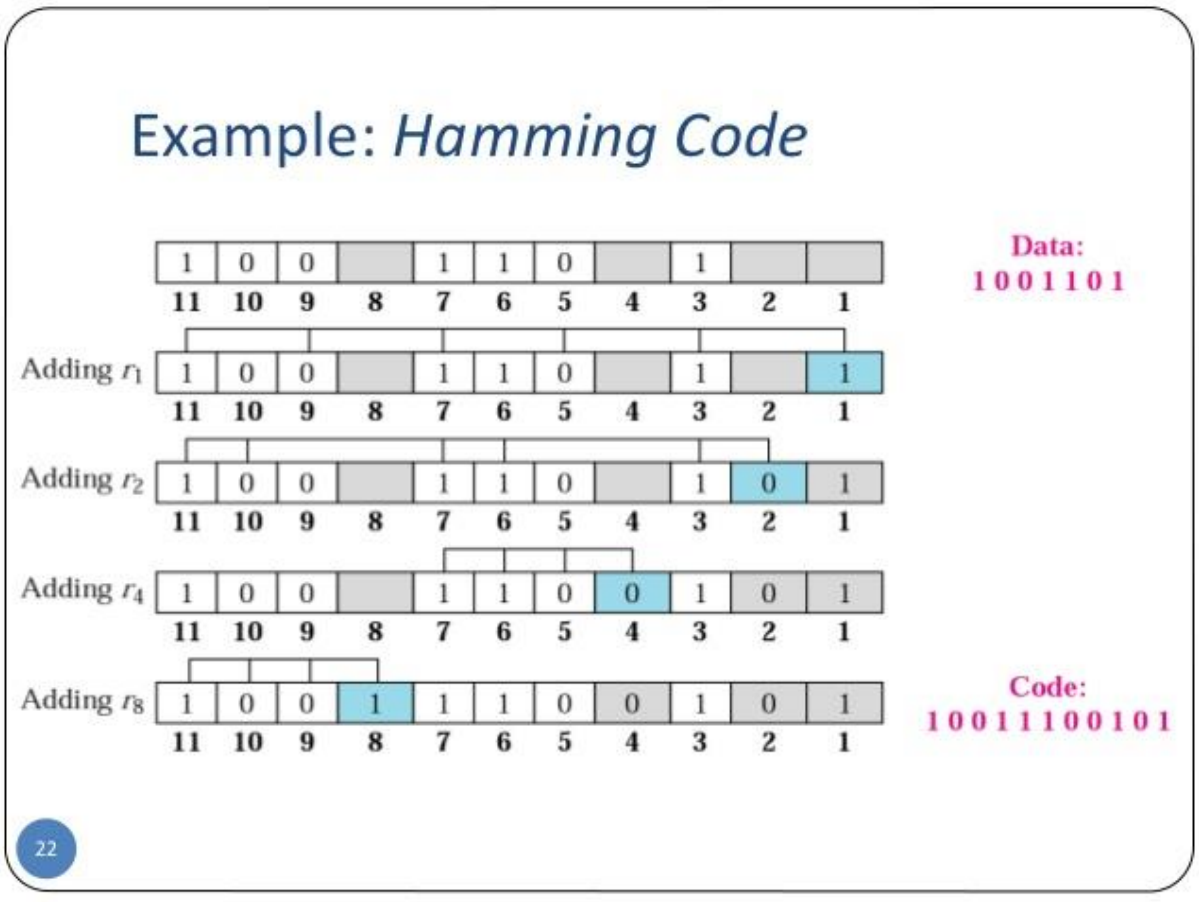
- We will examine 4 different error-correcting codes:
  1. Hamming codes
  2. Binary convolutional codes
  3. Reed-Solomon codes
  4. Low-Density Parity Check codes
- These codes add redundancy to the information that is sent.
- A frame consists of m data (i.e., message) bits & r redundant (i.e., check) bits.
- In a block code, the r check bits are computed solely as a function of the m data bits with which they are associated, as though the m bits were looked up in a large table to find their corresponding r check bits.
- In a systematic code, the m data bits are sent directly, along with the check bits, rather than being encoded themselves before they are sent.

- In a linear code, the r check bits are computed as a linear function of the m data bits. Exclusive OR (XOR) or modulo 2 addition is a popular choice. This means that encoding can be done with operations such as matrix multiplications or simple logic circuits.
- Let the total length of a block be n (i.e., n = m + r). We will describe this as an (n, m) code. An n-bit unit containing data & check bits is referred to as a n bit codeword.
- Code rate (rate) - the fraction of the codeword that carries information that is not redundant, or m/n.
    - 1/2 for a noisy channel, in which case half of the received information is redundant
    - close to 1 for a high-quality channel, with only a small number of check bits added to a large message.
- Given any 2 codewords that may be transmitted or received—say, 10001001 & 10110001—it is possible to determine how many corresponding bits differ. In this case, 3 bits differ. To determine how many bits differ, just XOR the two codewords & count the number of 1 bits in the result.
- For example: 10001001 XOR 10110001 = 00111000. The number of bit positions in which 2 codewords differ is called the Hamming distance

## Hamming Codes

- Bits of codeword are numbered consecutively, starting with bit 1 at the left end, bit 2 to its immediate right, & so on
- The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits
- The rest (3, 5, 6, 7, 9, etc.) are filled up with the m data bits. This pattern is shown for an (11,7) Hamming code with 7 data bits & 4 check bits in the figure
- Each check bit forces the modulo 2 sums, or parity, of some collection of bits, including itself, to be even (or odd)
- To see which check bits the data bit in position k contributes to, rewrite k as a sum of powers of 2
- For example, 11 = 1 + 2 + 8 & 29 = 1 + 4 + 8 + 16
- A bit is checked by just those check bits occurring in its expansion (e.g., bit 11 is checked by bits 1, 2, & 8)
- In the example, the check bits are computed for even parity sums for a message that is the ASCII letter “A”.
- This construction gives a code with a Hamming distance of 3, which means that it can correct single errors (or detect double errors).

- When a codeword arrives, the receiver redoes the check bit computations including the values of the received check bits. These are the check results.
- If the check bits are correct, for even parity sums, each check result should be 0. In this case, the codeword is accepted as valid.
- The set of check results forms the error syndrome that is used to pinpoint & correct the error
- Here, a single-bit error occurred on the channel, so the check results are 0, 1, 0, & 1 for k = 8, 4, 2, & 1, respectively. This gives a syndrome of 0101 or 4 + 1=5.
- By the design of the scheme, this means that the 5th bit is in error
- Flipping the incorrect bit (which might be a check bit or a data bit) & discarding the check bits gives the correct message of an ASCII "A".
- Hamming distances are valuable for understanding block codes
- Hamming codes are used in error-correcting memory

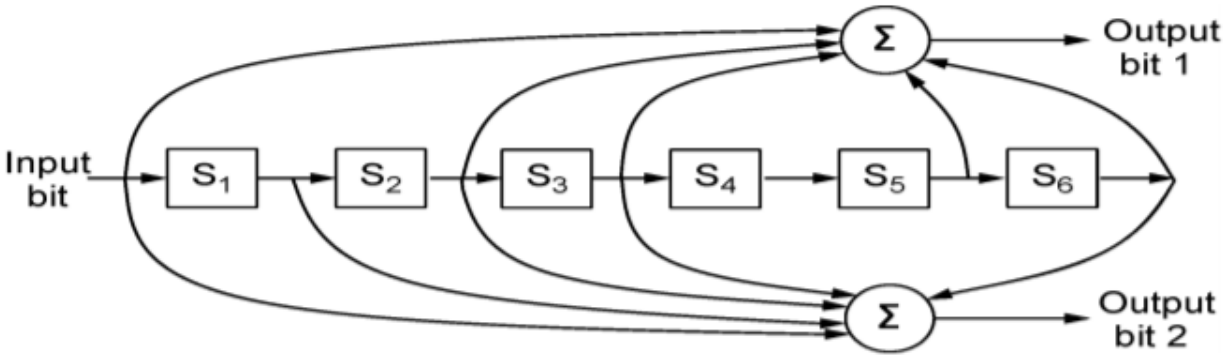



## Binary Convolutional Codes

- Encoder processes a sequence of input bits & generates a sequence of output bits

- No natural message size or encoding boundary as in a block code
- Output depends on the current & previous input bits, i.e., the encoder has memory
- The no. of previous bits on which the output depends is called the constraint length of the code
- Convolutional codes are specified in terms of their rate & constraint length
- Widely used in deployed networks, as part of the GSM cell phone system, in satellite communications, & in 802.11
- A popular convolutional code is shown in the figure. This code is known as the NASA convolutional code of r = 1/2 & k = 7, since it was 1st used for the Voyager space missions starting in 1977. Since then, it has been liberally reused, for example, as part of 802.11.
- In the previous figure, each input bit on the left-hand side produces 2 output bits on the right-hand side that are XOR sums of the input & internal state
- Since it deals with bits & performs linear operations, this is a binary, linear convolutional code
- Since 1 input bit produces 2 output bits, the code rate is ½
- Not systematic since none of the output bits is simply the input bit
- The internal state is kept in 6 memory registers. Each time another bit is input, the values in the registers are shifted to the right
- If 111 is input & the initial state is all 0s, the internal state, written left to right, will become 100000, 110000, & 111000 after the 1st, 2nd, & 3rd bits have been input
- The output bits will be 11, followed by 10, & then 01
- It takes 7 shifts to flush an input completely so that it doesn't affect the output. The constraint length of this code is thus k = 7.
- A convolutional code is decoded by finding the sequence of input bits that is most likely to have produced the observed sequence of output bits (which includes any errors)
- For small values of k, this is done with a widely used algorithm developed by Viterbi
- The algorithm walks the observed sequence, keeping for each step & for each possible internal state the input sequence that would have produced the observed sequence with the fewest errors
- The input sequence requiring the fewest errors at the end is the most likely message
- Convolutional codes have been popular in practice because it is easy to factor the uncertainty of a bit being a 0 or a 1 into the decoding

- Suppose −1V is the logical 0 level & +1V is the logical 1 level, we might receive 0.9V & −0.1V for 2 bits • Instead of mapping these signals to 1 & 0 right away, we would like to treat 0.9V as ‘‘very likely a 1’’ & −0.1V as ‘‘maybe a 0’’ & correct the sequence
- Extensions of the Viterbi algorithm can work with these uncertainties to provide stronger error correction
- This approach of working with the uncertainty of a bit is called soft-decision decoding. Conversely, deciding whether each bit is a 0 or a 1 before subsequent error correction is called hard-decision decoding.



The NASA binary convolutional code used in 802.11 (r = ½, k=7).

## Reed – Solomon Code

- Linear & systematic block codes
- Operates on m bit symbols
- Every n degree polynomial is uniquely determined by n + 1 points
- For example, a line having the form ax + b is determined by 2 points
- Extra points on the same line are redundant, which is helpful for error correction
- Suppose we have 2 data points that represent a line, & we send those 2 data points plus two check points chosen to lie on the same line
- If 1 of the points is received in error, we can still recover the data points by fitting a line to the received points
- 3 of the points will lie on the line, & 1 point, the 1 in error, will not
- By finding the line we have corrected the error
- Codes are defined as polynomials that operate over finite fields, but they work in a similar manner
- For m bit symbols, the codewords are 2m−1 symbols long
- A popular choice is to make m = 8 so that symbols are bytes
- A codeword is then 255 bytes long

- The (255, 233) code is widely used; it adds 32 redundant symbols to 233 data symbols
- Decoding with error correction is done with an algorithm developed by Berlekamp & Massey that can efficiently perform the fitting task for moderate-length codes
- Widely used in practice because of their strong error-correction properties, particularly for burst errors
- Used for DSL, data over cable, satellite communications, & perhaps most ubiquitously on CDs, DVDs, & Blu-ray discs
- Because they are based on m bit symbols, a single-bit error & an m-bit burst error is both treated simply as 1 symbol error. When 2t redundant symbols are added, a Reed-Solomon code can correct up to t errors in any of the transmitted symbols
- This means, for example, that the (255, 233) code, which has 32 redundant symbols, can correct up to 16 symbol errors
- Since the symbols maybe consecutive & they are each 8 bits, an error burst of up to 128 bits can be corrected
- If the error model is one of erasures (e.g., a scratch on a CD that obliterates some symbols), up to 2t errors can be corrected
- Used in combination with other codes such as a convolutional code
- Convolutional codes are effective at handling isolated bit errors, but they will fail, likely with a burst of errors, if there are too many errors in the received bit stream
- By adding a Reed-Solomon code within the convolutional code, the Reed-Solomon decoding can mop up the error bursts, a task at which it is very good
- The overall code then provides good protection against both single & burst errors

## Low-Density Parity Check Code

- Linear block codes
- Invented by Robert Gallagher in his doctoral thesis
- Each output bit is formed from only a fraction of the input bits
- Leads to a matrix representation of the code that has a low density of 1s, hence the name for the code

- The received codewords are decoded with an approximation algorithm that iteratively improves on a best fit of the received data to a legal codeword & this corrects errors
- Practical for large block sizes & have excellent error-correction abilities that outperform many other codes in practice
- So, they are rapidly being included in new protocols
- They are part of the standard for digital video broadcasting, 10 Gbps Ethernet, power-line networks, & the latest version of 802.11

# Error detecting codes

- Over fibre or high-quality copper, the error rate is much lower, so error detection & retransmission is usually more efficient there for dealing with the occasional error
- There are 3 different error-detecting codes
- They are all linear, systematic block codes:
    1. Parity
    2. Checksums
    3. Cyclic Redundancy Checks (CRCs)

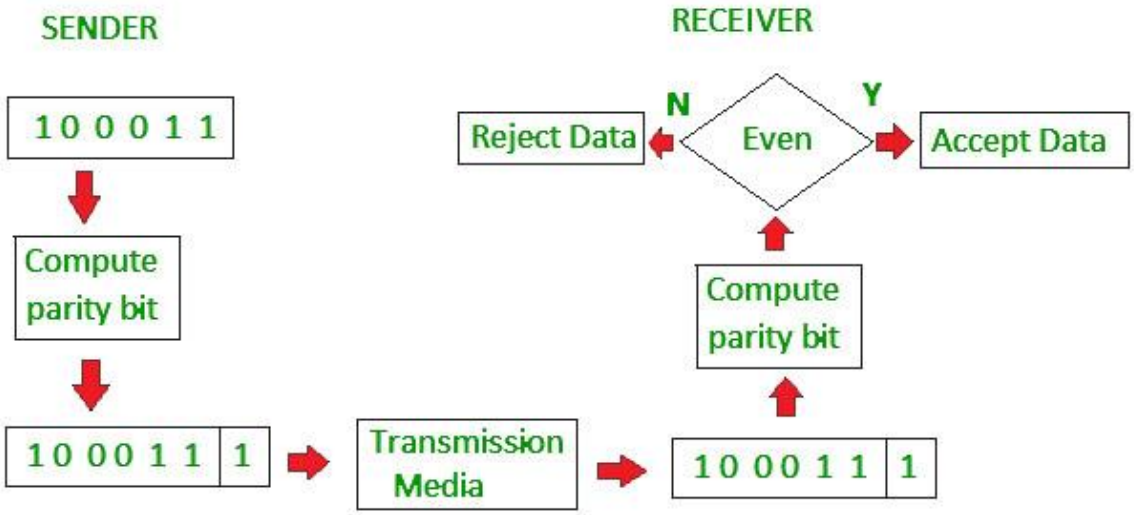
## Parity

1. Simple Parity check

- Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of:
    - 1 is added to the block if it contains odd number of 1's, &
    - 0 is added if it contains even number of 1's

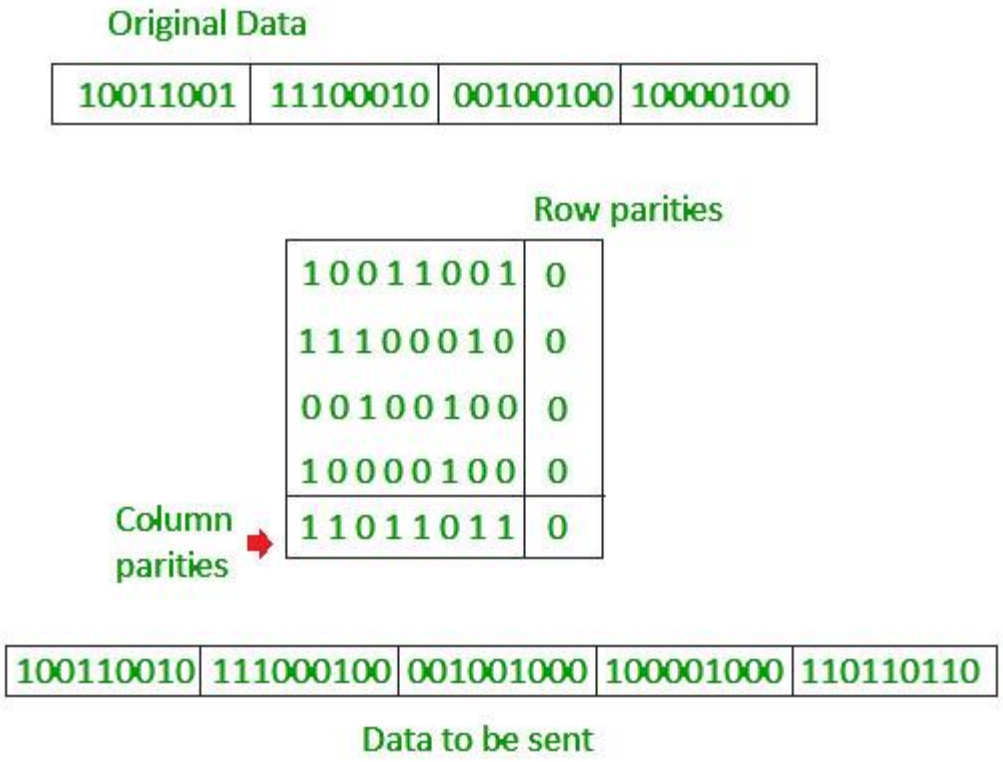
This scheme makes the total number of 1's even, that is why it is called even parity checking (shown in figure)

- Also,
    - 1 is added to the block if it contains even number of 1's, &
    - 0 is added if it contains odd number of 1's
- This scheme makes the total number of 1's odd, that is why it is called odd parity checking



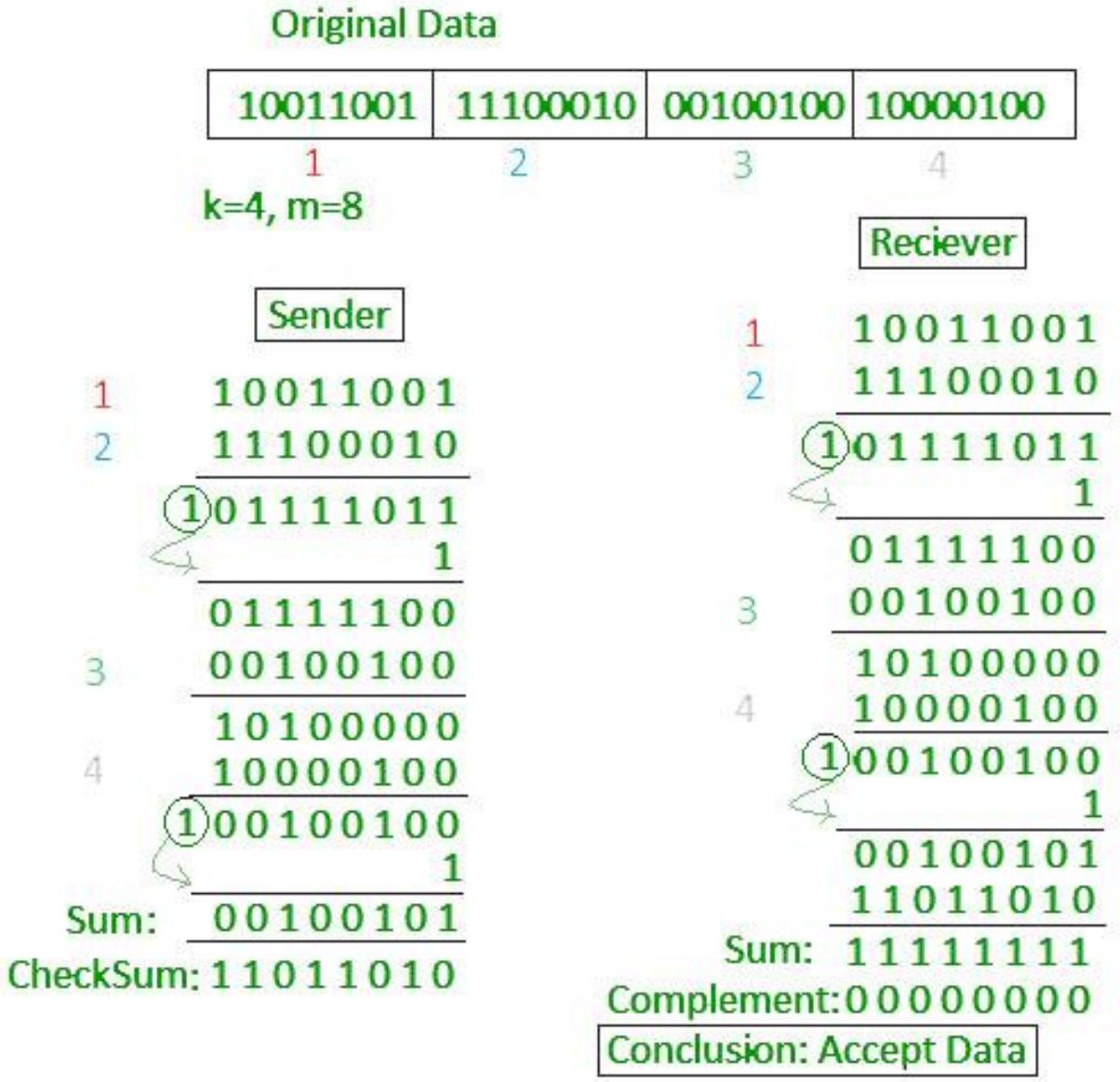

2. Two-dimensional Parity check

- Parity check bits are calculated for each row, which is equivalent to a simple parity check bit
- Parity check bits are also calculated for all columns, then both are sent along with the data
- At the receiving end these are compared with the parity bits calculated on the received data



## Checksums

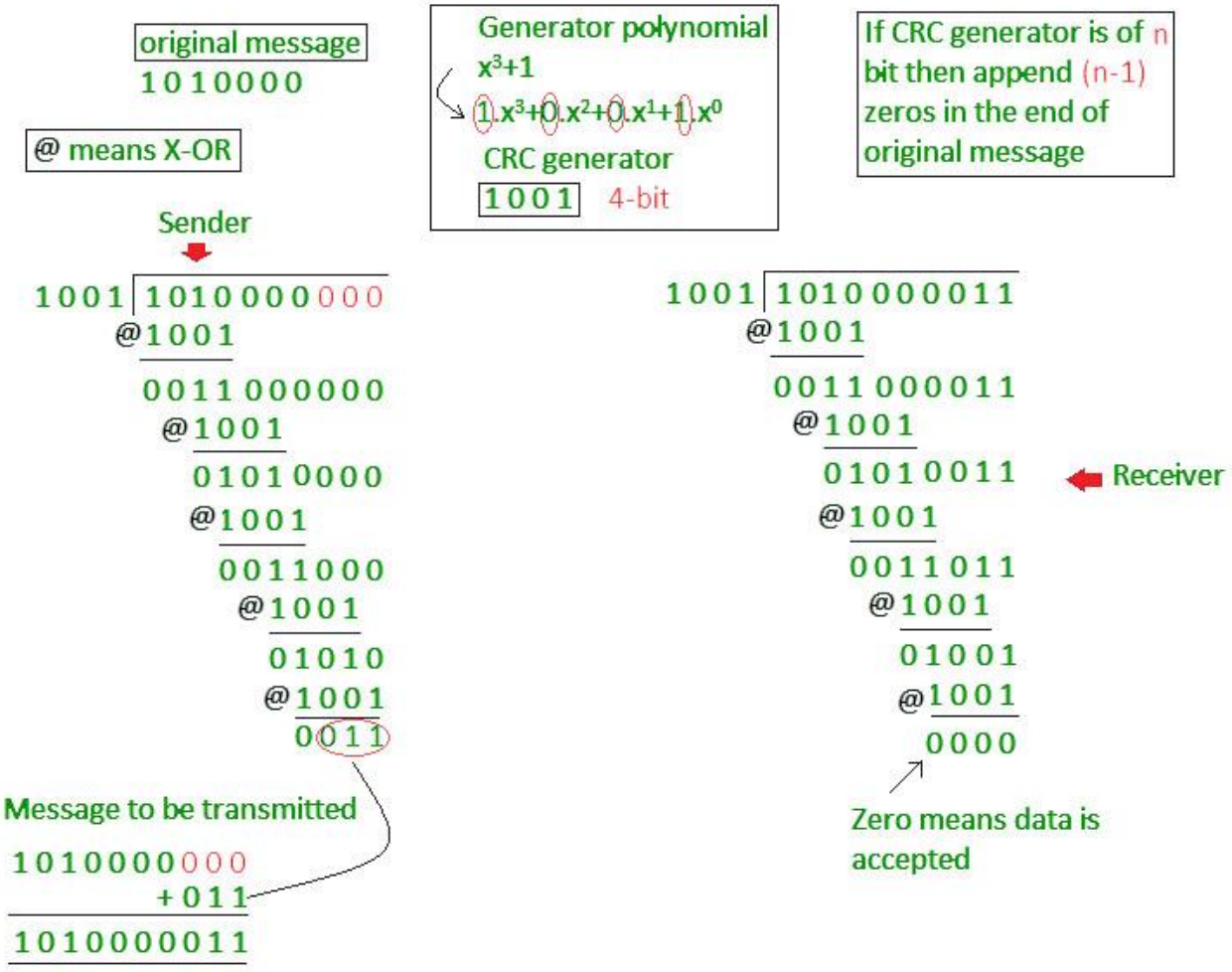
- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is 0, the received data is accepted; otherwise discarded.




## Cyclic Redundancy Checks

- Based on binary division.

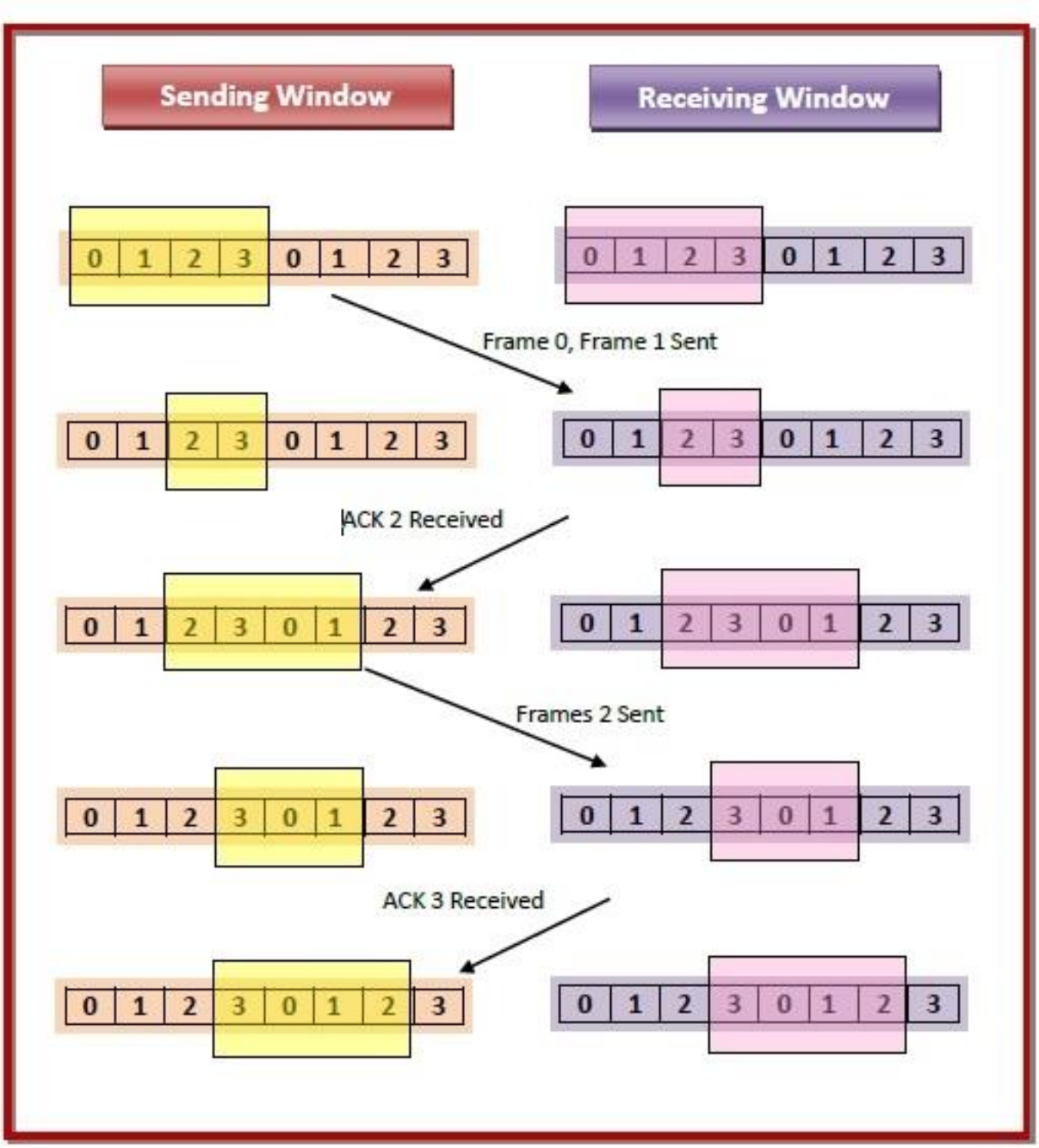
- A sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a 2nd, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct & is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.




# Sliding window protocols

- Data link layer protocols for reliable & sequential delivery of data frames
- Also used in Transmission Control Protocol
- Multiple frames can be sent by a sender at a time before receiving an acknowledgment from the receiver
- The term sliding window refers to the imaginary boxes to hold frames

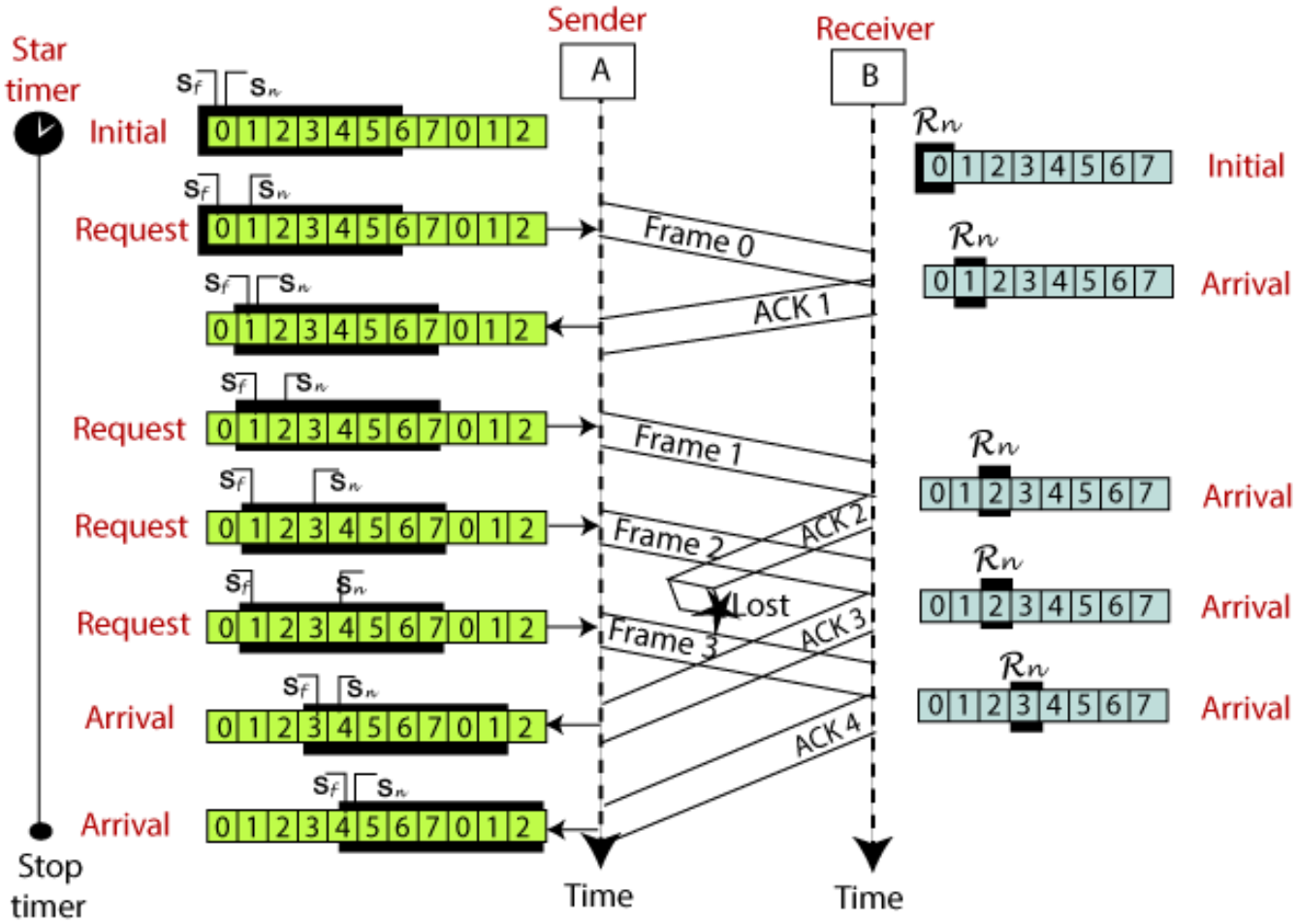
- Also known as windowing
- The sender has a buffer called the sending window & the receiver has buffer called the receiving window
- The size of the sending window determines the sequence number of the outbound frames. If the sequence number of the frames is an n-bit field, then the range of sequence numbers that can be assigned is 0 to $2^{n-1}$.
- The size of the sending window is $2^{n-1}$
- In order to accommodate a sending window size of $2^{n-1}$, a n-bit sequence number is chosen.
- The sequence numbers are numbered as modulo-n. For example, if the sending window size is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, & so on. The number of bits in the sequence number is 2 to generate the binary sequence 00, 01, 10, 11.
- The size of the receiving window is the maximum number of frames that the receiver can accept at a time
- It determines the maximum number of frames that the sender can send before receiving acknowledgment.
- Suppose that we have sender window & receiver window each of size 4




## Go – Back – N ARQ

- A category of sliding window protocol
- Provides for sending multiple frames before receiving the acknowledgment for the 1st frame
- The frames are sequentially numbered, & a finite number of frames are sent
- If the acknowledgment of a frame is not received within the time period, all frames starting from that frame are retransmitted
- Also known as Go-Back-N Automatic Repeat Request

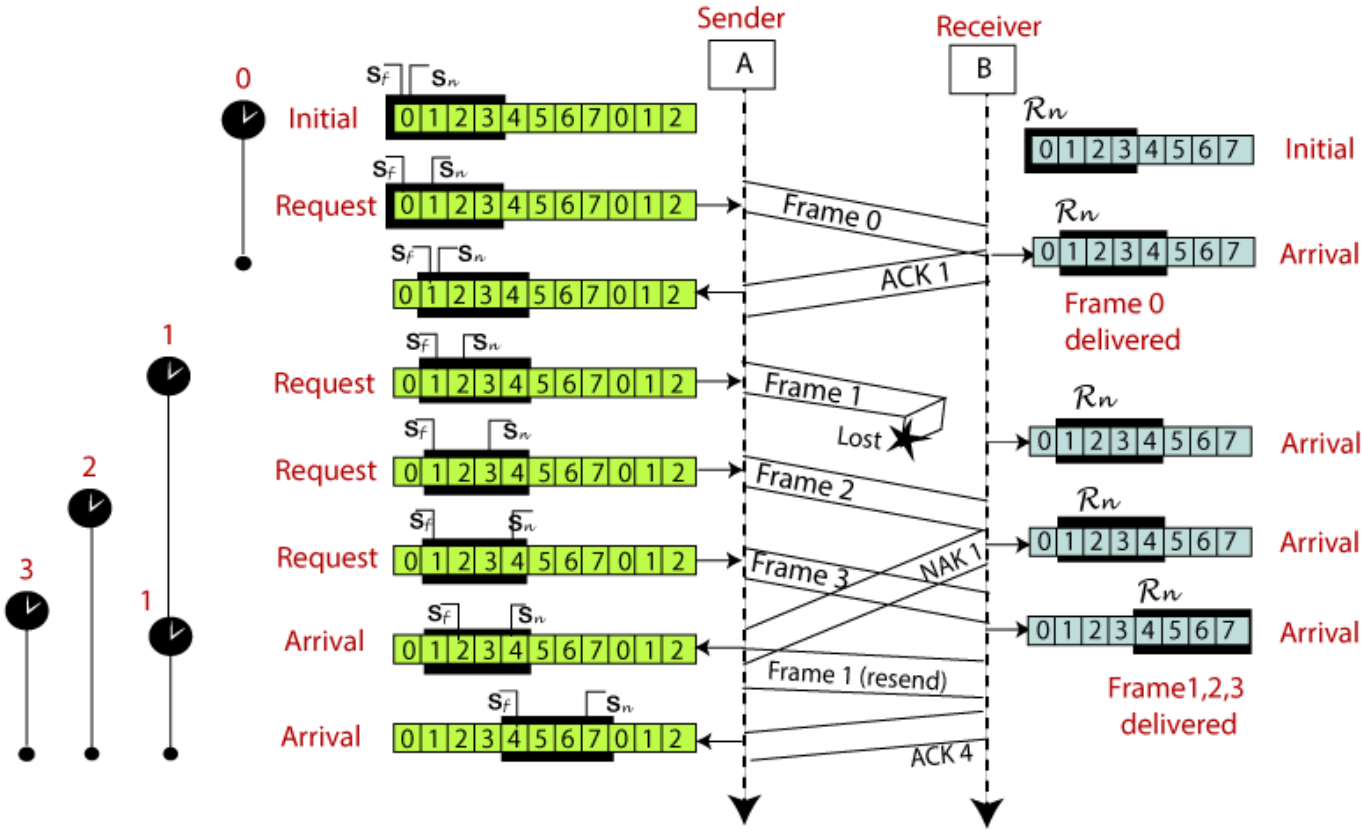
- If any frame is corrupted or lost, all subsequent frames must be sent again • The size of the sender window is N in this protocol
- For example, Go-Back-8, the size of the sender window, will be 8
- The receiver window size is always 1
- If the receiver receives a corrupted frame, it cancels it
- The receiver does not accept a corrupted frame
- When the timer expires, the sender sends the correct frame again
- The example of the Go-Back-N ARQ protocol is shown here:




## Selective Repeat ARQ

- Another category of sliding window protocol
- Provides for sending multiple frames before receiving the acknowledgment for the 1st frame
- But only the erroneous or lost frames are retransmitted, while the good frames are received & buffered
- Also known as the Selective Repeat Automatic Repeat Request
- Go-back-N ARQ protocol works well if it has fewer errors

- But if there is a lot of error in the frame, lots of bandwidth loss in sending the frames again
- So, we use the Selective Repeat ARQ protocol
- The size of the sender window is always equal to the size of the receiver window
- The size of the sliding window is always greater than 1
- If the receiver receives a corrupt frame, it doesn't directly discard it
- It sends a negative acknowledgment to the sender
- The sender sends that frame again as soon as on the receiving negative acknowledgment
- There is no waiting for any time-out to send that frame
- The example of the Selective Repeat ARQ protocol is shown:



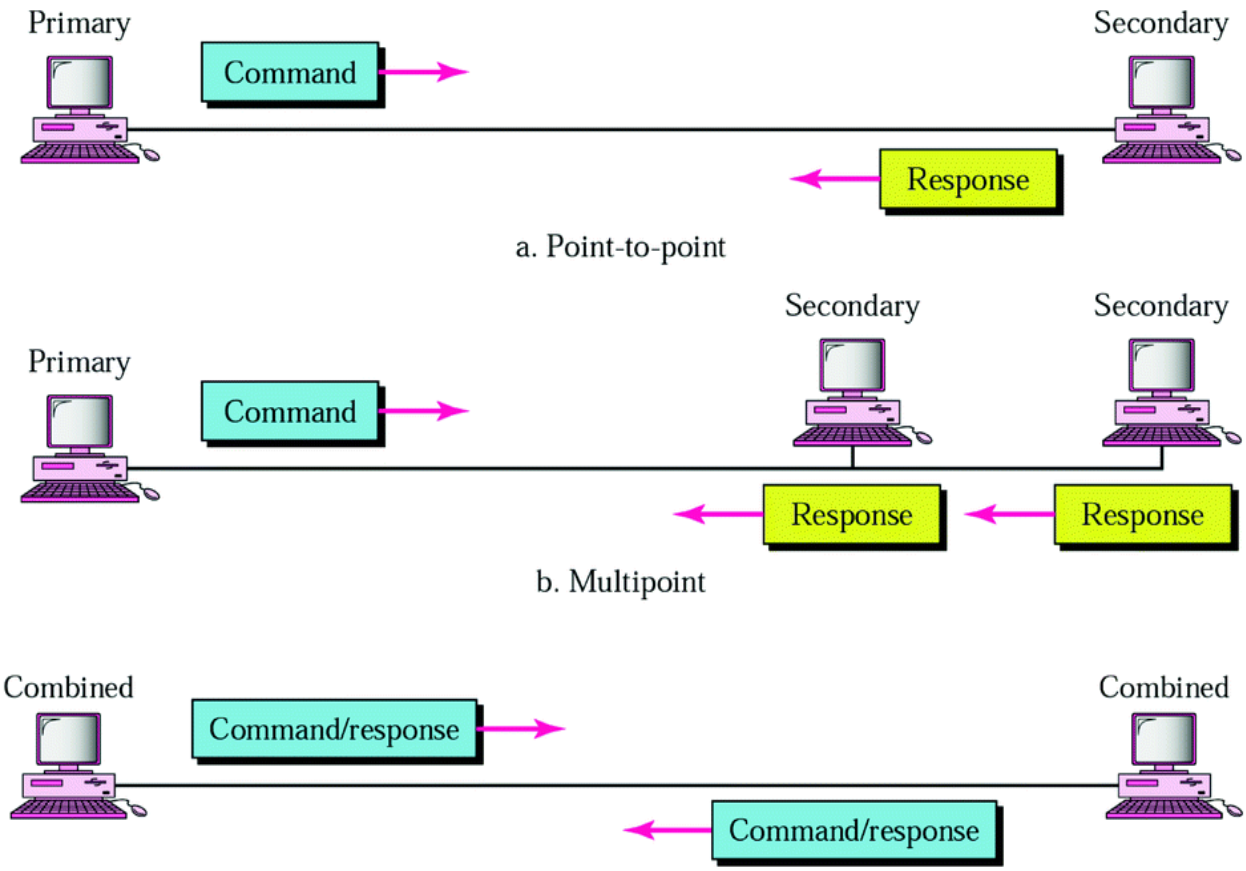

| Go-Back-N ARQ | Selective Repeat ARQ |
|---|---|
| If a frame is corrupted or lost in it, all subsequent frames have to be sent again. | In this, only the frame is sent again, which is corrupted or lost. |
| If it has a high error rate, it wastes a lot of bandwidth. | There is a loss of low bandwidth. |

| | |
|---|---|
| It is less complex. | It is more complex because it has to do sorting and searching as well. And it also requires more storage. |
| It doesn't require sorting. | In this, sorting is done to get the frames in the correct order. |
| It doesn't require searching. | The search operation is performed in it. |
| It is used more. | It is used less because it is more complex |

# High-Level Data Link Control (HDLC) protocol

- HDLC is bit oriented protocol for communication over point to point & multipoint links.
- It is developed by ISO.
- It offers a high level of flexibility, adaptability, reliability & efficient of operation.
- It falls under the ISO standards ISO 3309 & ISO 4335.
- It specifies a packetization standard for serial links.
- It has found itself being used throughout the world.
- It has been so widely implemented because it supports both half-duplex & full-duplex communication lines, point to-point (peer to peer) & multi-point networks & switched or non-switched channels.
- HDLC supports several modes of operation, including a simple sliding-window mode for reliable delivery. Since Internet provides retransmission at higher levels (i.e., TCP), most Internet applications use HDLC's unreliable delivery mode, **Unnumbered Information**.
- Other benefits of HDLC are that the control information is always in the same position, & specific bit patterns used for control differ dramatically from those in representing data, which reduces the chance of errors.
- It has also led to many subsets. Two subsets widely in use are **Synchronous Data Link Control (SDLC) & Link Access Procedure Balanced (LAP-B).**
- **Three types of stations have been defined in HDLC**:
    - Primary station: The Primary station has a responsibility of connecting & disconnecting the data link. The frame send by a primary station are called commands.
    - Secondary Station: It operates under the control of a primary station. The frame sent by the secondary station are called Response.

- Combined Station: A combined station can act as a primary as well as secondary station. It can issue both command and response.

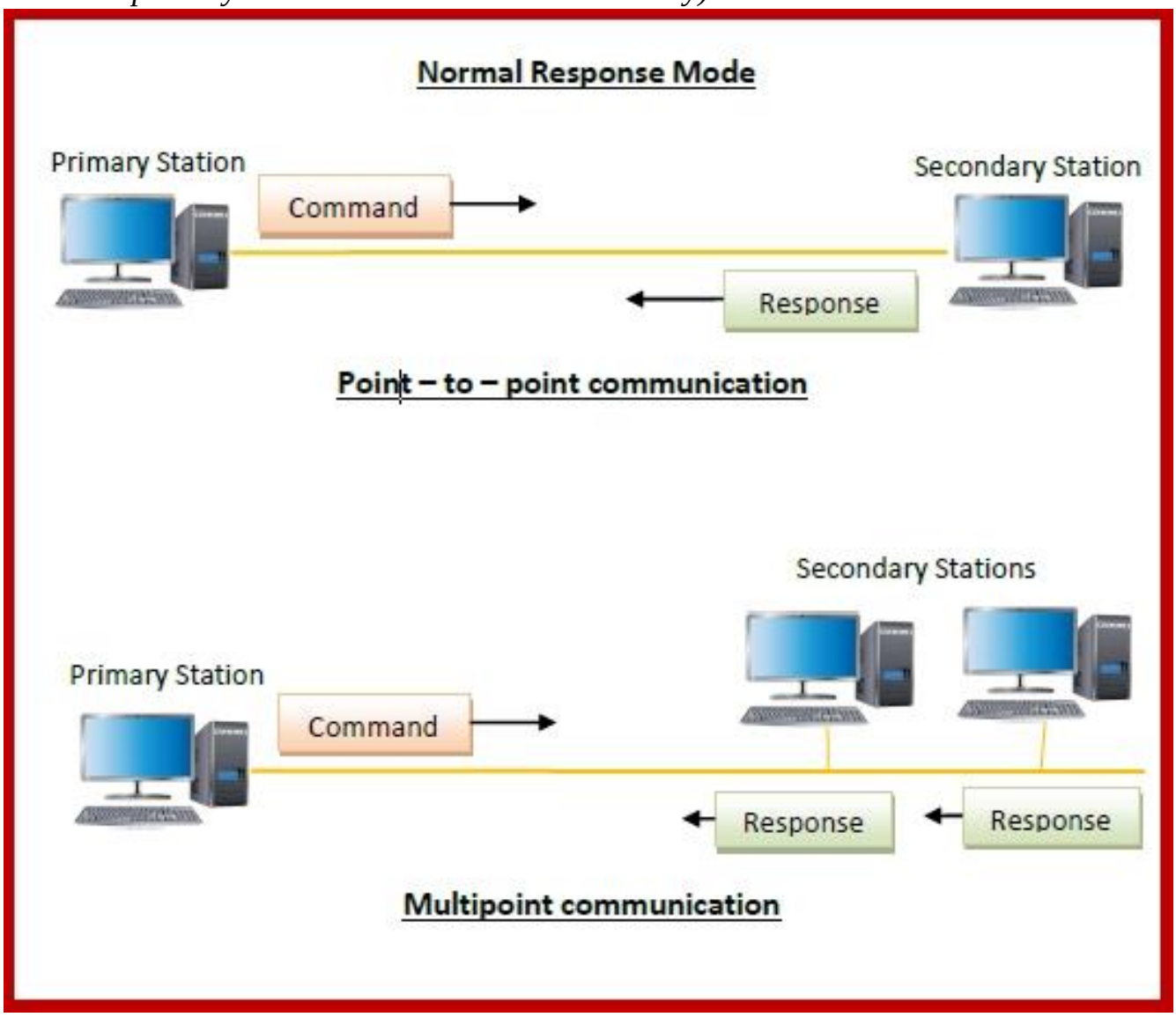



## HDLC Operational Modes

- A mode in HDLC is the relationship between two devices involved in an exchange
- The mode describes who controls the link
- Exchanges over unbalanced configurations are always conducted in normal response mode.
- Exchanges over symmetric (*consists of exactly two nodes i.e., two independent point-to-point unbalanced station configurations*) or balanced (*consists of two combined stations*) configurations can be set to specific mode using a frame design to deliver the command.
- HDLC offers **three different modes of operation**. These three modes of operations are:
  - Normal Response Mode (NRM)
  - Asynchronous Response Mode (ARM)
  - Asynchronous Balanced Mode (ABM)

## Normal Response Mode

- This is the mode in which the primary station initiates transfers to the secondary station.
- The secondary station can only transmit a response when, & only when, it is instructed to do so by the primary station. In other words, the secondary station must receive explicit permission from the primary station to transfer a response.
- After receiving permission from the primary station, the secondary station initiates its transmission. This transmission from the secondary station to the primary station may be much more than just an acknowledgment of a frame.
- It may in fact be more than one information frame. Once the last frame is transmitted by the secondary station, it must wait once again from explicit permission to transfer anything, from the primary station.
- Normal Response Mode is only used within an unbalanced configuration (*one station is primary & all other stations are secondary).*

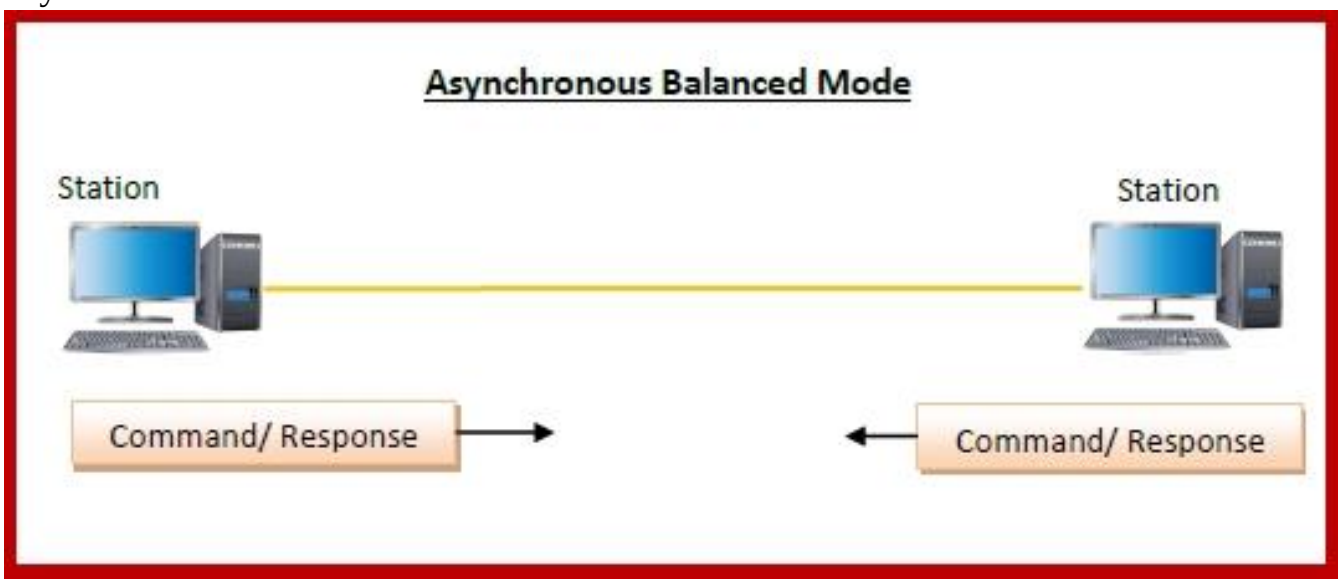


## Asynchronous Response Mode

- In this mode, the primary station doesn't initiate transfers to the secondary station. In fact, the secondary station doesn't have to wait to receive explicit permission from the primary station to transfer any frames.
- The frames may be more than just acknowledgment frames. They may contain data, or control information regarding the status of the secondary station.
- This mode can reduce overhead on the link, as no frames need to be transferred in order to give the secondary station permission to initiate a transfer. But some limitations do exist.
- Due to the fact that this mode is asynchronous, the secondary station must wait until it detects & idle channel before it can transfer any frames. This is when the ARM link is operating at half-duplex.
- If the ARM link is operating at full duplex, the secondary station can transmit at any time.
- In this mode, the primary station still retains responsibility for error recovery, link setup, & link disconnection.

## Asynchronous Balanced Mode

- This mode is used in case of combined stations.
- There is no need for permission on the part of any station in this mode. This is because combined stations don't require any sort of instructions to perform any task on the link.

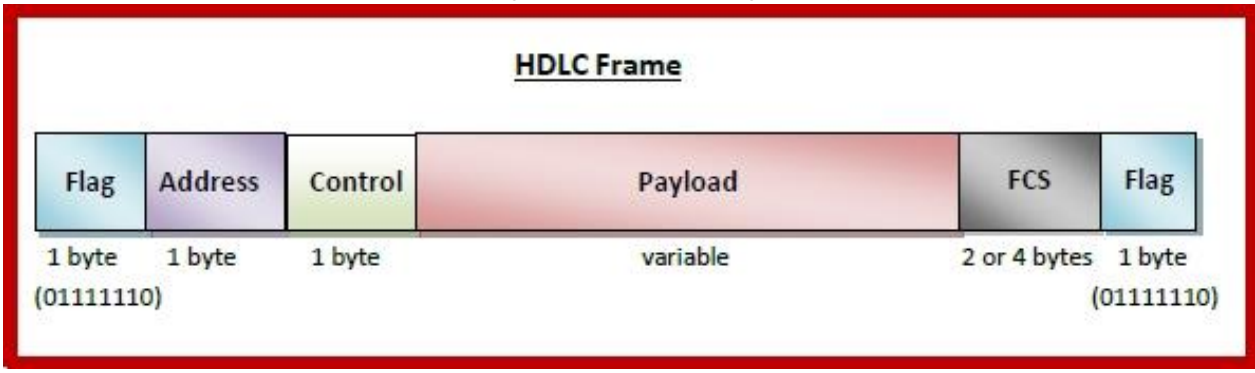



- Normal Response Mode is used most frequently in multi-point lines, where the primary station controls the link.

- Asynchronous Response Mode is better for point-to-point links, as it reduces overhead.
- Asynchronous Balanced Mode is not used widely today.
- The term "***asynchronous***" in both ARM & ABM doesn't refer to the format of the data on the link. It refers to the fact that any given station can transfer frames without explicit permission or instruction from any other station.

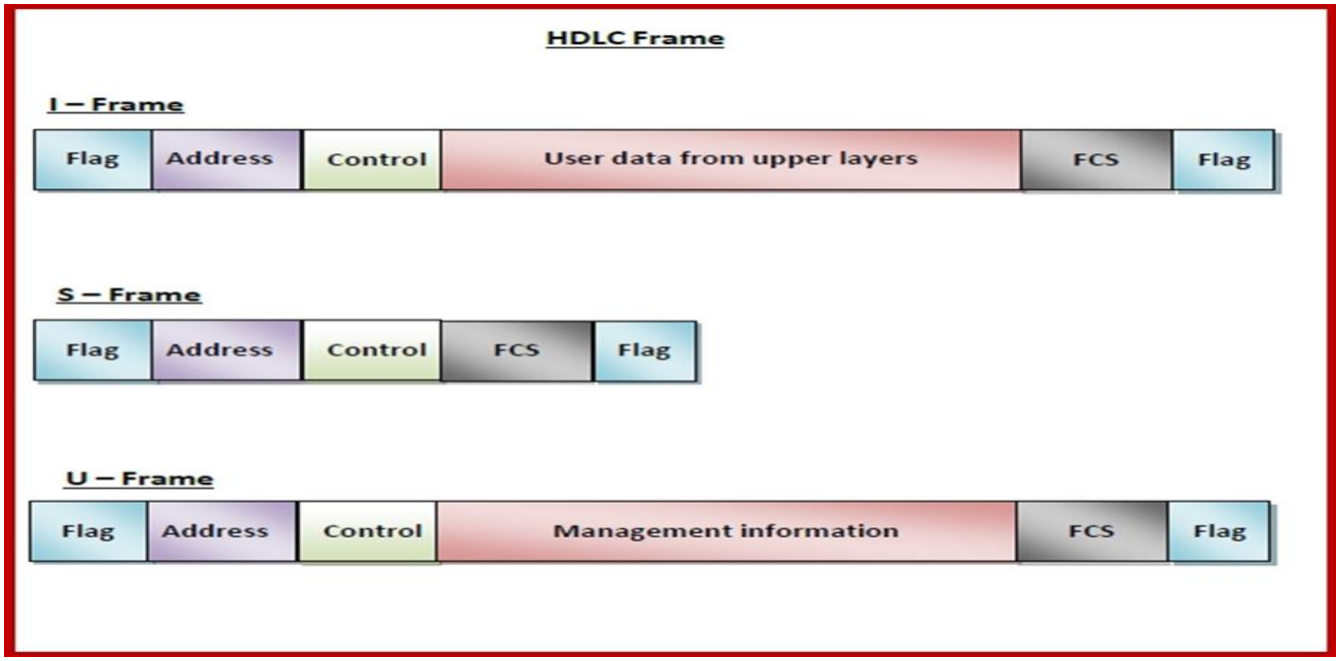
## HDLC Frame

- HDLC is a bit - oriented protocol where each frame contains up to six fields.
- The structure varies according to the type of frame. The **fields of an HDLC** frame are −
    - **Flag** − It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
    - **Address** − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.
    - **Control** − It is 1- or 2-bytes containing flow & error control information.
    - **Payload/User Information** − This carries the data from the network layer. Its length may vary from one network to another.
    - **FCS** − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)




## Types of HDLC Frames

- There are **three types of HDLC frames**. The type of frame is determined by the ***control field*** of the frame –




## I-frame

- I-frames or **Information frames** carry user data from the network layer.
- They also include flow & error control information that is piggybacked on user data.
- User's information is sent
- Control field in I-Frame

| | Sequence Number | | | | Acknowledgement Number | | |
|---|---|---|---|---|---|---|---|
| 0 | | | | P/F | | | |

- The **1st bit** of control field of I-frame is **0**.
- **Next 3 bits (2nd, 3rd, 4th)** indicate **Sequence Number**. (0-7, i.e., 000, 001, 010, 011, ….111)
- **5th bit** indicates **P/F (Poll/Final).**

| **P/F** | |
|---|---|
| 1 | Primary Section is sending data |
| 0 | Secondary Section is sending data |

- Last **3 bits (6th, 7th, 8th)** indicate **Acknowledgement number**, which is used to check whether acknowledgement is received or not.

## S-frame

- S-frames or **Supervisory frames** don't contain information field.

- They are used for flow & error control when piggybacking is not required.
- It is used to pass only control information.
- Control field in S-Frame

To indicate S-Frame — Acknowledgement Number

| 1 | 0 | S1 | S2 | P/F | | | |
|---|---|---|---|---|---|---|---|

  - The **1st two bits** of control field of S-frame is **10**.
  - Value of **S1 & S2** will decide the status provided:

| S1 | S2 | Status |
|---|---|---|
| 0 | 0 | Receiver is ready |
| 0 | 1 | Receiver is not ready |
| 1 | 0 | Send negative acknowledgement |
| 1 | 1 | Specified in Selective-Repeat Transmission; The number being transmitted is specified in Acknowledgement Number |

## U-frame

- U-frames or **Un-numbered frames** are used for myriad miscellaneous functions, like link management.
- It may contain an information field, if required.
- We send System Management Information instead of User Information.
- Control field in U-Frame

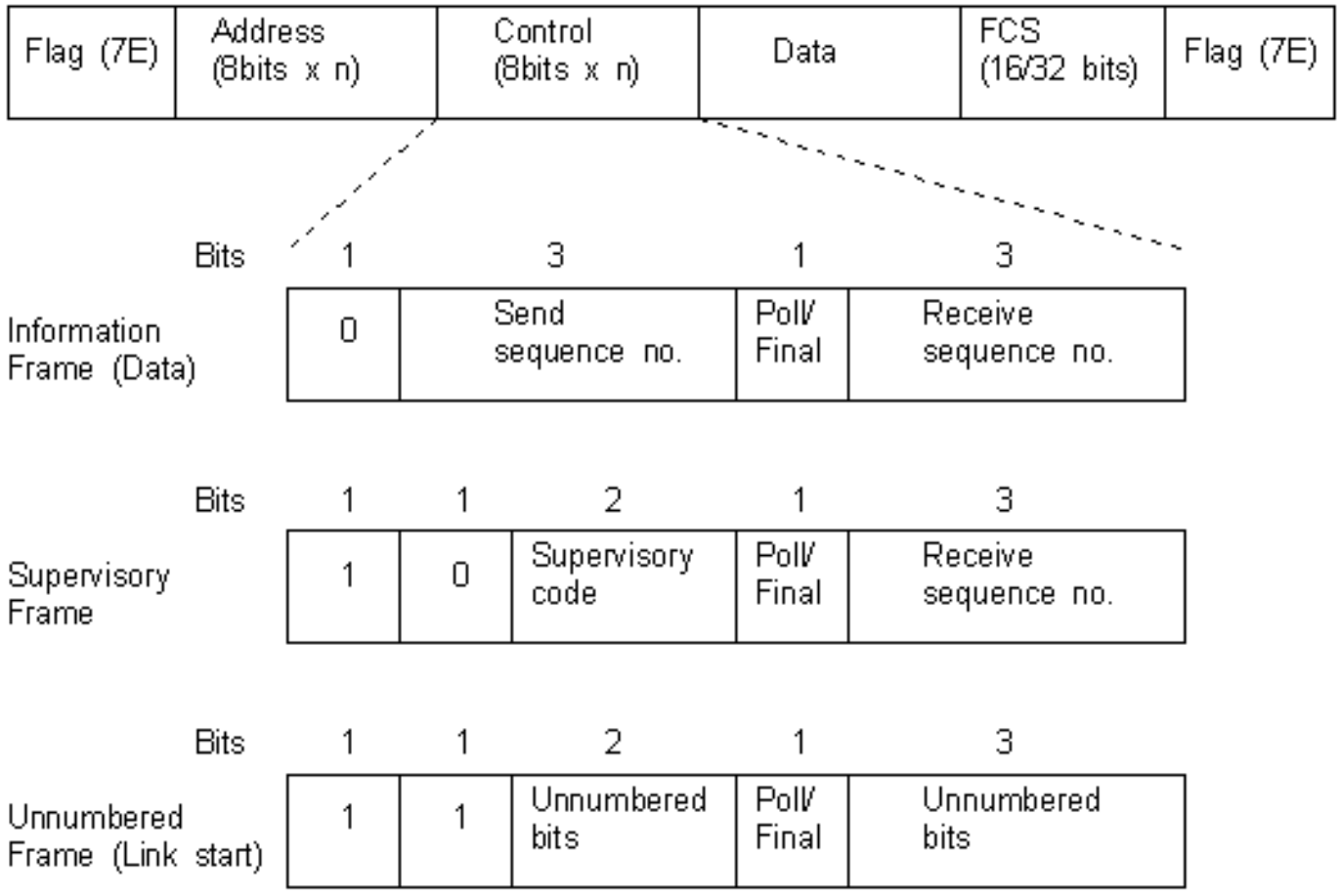
To indicate U-Frame

| 1 | 1 | | | P/F | | | |
|---|---|---|---|---|---|---|---|

  - The **1st two bits** of control field of U-frame is **11**.
  - Here, $(5\text{-}2)^5$, i.e., 243 combinations made. So, that much data can be transmitted, & each one is used to manage each system.
- If particular data of flag comes in the data field, we can't distinguish between flag & user information. To prevent this, after 5 consecutive 1's an additional 0 is inserted.
- E.g.:
  Flag: 01111110

User Information: 01111110
Updated User Information: 011111010

- So, a redundant 0 is inserted in between, & is called zero stuffing, or bit stuffing.
- When it comes to receiver side, that particular 0 is deleted.




# Medium Access Control (MAC) sublayer

- The Media Access Control (MAC) Data Communication Networks Protocol Sub-layer
- Also known as the **Medium Access Control**
- A sub-layer of the data link layer specified in the seven-layer OSI model.
- The **medium access layer** was made necessary by systems that share common communications medium. Typically, these are local area networks.
- The MAC layer is the "lower" part of the second OSI layer, i.e., datalink layer.
- In fact, the IEEE divided this layer into two layers: "above" is the control layer, the logical connection, **Logical Link Control** (LLC), & "below" is the control layer, **Medium Access Control** (MAC).

- The **LLC layer** is standardized by IEEE as the 802.2 since the beginning of 1980.
- Its purpose is to allow level 3 network protocols, (for e.g., IP) to be based on a single layer (the LLC layer) regardless of underlying protocol used, including Wi-Fi, Ethernet or Token Ring. All Wi-Fi data packets, so carry a pack LLC, which contains itself packets from the upper network layers.
- The header of a packet LLC indicates the type of layer 3 protocol in it: most of the time, it is IP protocol, but it could be another protocol, such as **IPX (Internet Packet Exchange)** for example.
- With the help of LLC layer, it is possible to have multiple Layer 3 protocols on the same network, at a time. So, LAN nodes use same communication channel for transmission.

- The **MAC sub-layer** has two primary responsibilities:
  - **Data encapsulation**, including frame assembly before transmission; &
  - **Frame parsing/Error detection** during & after reception.
- Media access control include **initiation of frame transmission** & recovery from transmission failure.

## Protocols used by Medium Access Layer

- ALOHA
- CSMA
- CSMA/CD
- CSMA/CA

### ALOHA

- ALOHA is a system to coordinate & arbitrate access to a shared communication channel.
- Developed in the 1970s at the University of Hawaii
- The original system used terrestrial radio broadcasting, but the system has been implemented in satellite communication systems.
- A shared communication system like ALOHA requires a method of handling collisions that occur when two or more systems attempt to transmit on the channel at the same time.
- **In the ALOHA system, a node transmits whenever data is available to send**. If another node transmits at the same time, a collision occurs, & the

frames that were transmitted are lost. But a node can listen to broadcasts on the medium, even its own, & determine whether the frames were transmitted.

## Carrier Sensed Multiple Access (CSMA)

- CSMA is a network access method used on shared network topologies such as Ethernet to control access to the network.
- Devices attached to the network cable listen (carrier sense) before transmitting. If the channel is in use, devices wait before transmitting.
- MA (Multiple Access) indicates that many devices can connect to & share the same network. All devices have equal access to use the network when it is clear.
- Even though devices attempt to sense whether the network is in use, there is a good chance that two stations will attempt to access it at the same time.
- On large networks, the transmission time between one end of the cable & another is enough that one station may access the cable even though another has already just accessed it. There are two methods for avoiding these so-called collisions, listed below.

## CSMA/CD (Carrier Sense Multiple Access/Collision Detection)

- In **CD (Collision Detection)**, when two devices sense a clear channel, they attempt to transmit at the same time, a collision occurs, & both devices stop transmission, wait for a random amount of time, & then retransmit. This is the technique used to access the 802.3 Ethernet network channel.
- This method handles collisions as they occur, but if the bus is constantly busy, collisions can occur so often that performance drops drastically.
- It is estimated that **a network traffic must be less than 40 percent of the bus capacity, for the network to operate efficiently**.
- If distances are long, time lags occur that may result in inappropriate carrier sensing, & hence collisions.

## CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)

- In **CA (Collision Avoidance),** collisions are avoided because each node signals its intent to transmit before actually doing so.
- This method is not popular because it requires excessive overhead that reduces performance.

# Channel allocation problem

- In broadcast networks, a single channel is shared by several stations. This channel can be allocated to only one transmitting user at a time.
- The channel might be a portion of the wireless spectrum in a geographic region, or a single wire or optical fibre to which multiple nodes are connected. It does not matter.
- The channel connects each user to all other users & any user who makes full use of the channel & interferes with other users who also wish to use the channel.
- There are two different methods of channel allocations:
- Static Channel Allocation
- Dynamic Channel Allocation

## Static Channel Allocations

- In this method, a single channel is divided among various users either on the basis of frequency or on the basis of time.
- It either uses
    - **FDM (Frequency Division Multiplexing)** - In FDM, a fixed frequency is assigned to each user; or
    - **TDM (Time Division Multiplexing)**, In TDM, a fixed time slot is assigned to each user.
- Traditional way of allocating a single channel, like a telephone trunk, among multiple competing users is Frequency Division Multiplexing (FDM).
- If there are N users, the bandwidth is divided into N equal- sized portions, * each user is assigned by one portion. Since each user has a private frequency band, there is no interference between users.
- When there is only a small & constant number of users, each of which has a heavy (buffered) load of traffic, FDM is a simple & efficient allocation mechanism.

- But, when the number of senders is large & continuously varying, or, if the traffic is bursty, **FDM** presents some problems.
- If the spectrum is cut up into N regions & fewer than N users are currently interested in communicating, a large piece of valuable spectrum will be wasted.
- Also, if more than N users want to communicate, some of them will be denied permission for lack of bandwidth, even if some of the users are assigned with a frequency band that can hardly transmit or receive anything.

- Same problem is applicable to **Time Division Multiplexing (TDM)**, where each user is statically allocated every $N^{th}$ time slot.
- If a user doesn't use the allocated slot, it just lies idle.

## Dynamic Channel Allocation

- In this method, no user is assigned fixed frequency or fixed time slot.
- All users are dynamically assigned with a frequency or time slot, depending upon the requirements of the user.
- **Assumptions for Dynamic Channel Allocation** include:
    1. **Independent Traffic**
        - The model consists of N independent stations (e.g., computers, telephones), each with a program or user that generates frames for transmission.
        - The expected number of frames generated in an interval of length $\Delta t$ is $\lambda \Delta t$, where $\lambda$ is a constant (the arrival rate of new frames).
        - Once a frame has been generated, the station is blocked & does nothing until the frame has been successfully transmitted.
    2. **Single Channel**
        - A single channel is available for all communication.
        - All stations can transmit on it & all can receive from it.
        - The stations are assumed to be equally capable, though protocols may assign them different roles (e.g., priorities).
    3. **Observable Collisions**
        - If two frames are transmitted simultaneously, they overlap in time & the resulting signal is distorted. This event is called a **collision**.
        - All stations can detect that a collision has occurred.
        - A collided frame must be transmitted again later.
        - No errors other than those generated by collisions occur.
    4. **Continuous or Slotted Time**
        - Time may be assumed continuous, in which case frame transmission can begin at any instant.
        - Alternatively, time may be slotted or divided into discrete intervals (called slots). Frame transmissions must then begin at the start of a slot.
        - A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.
    5. **Carrier Sense or No Carrier Sense**

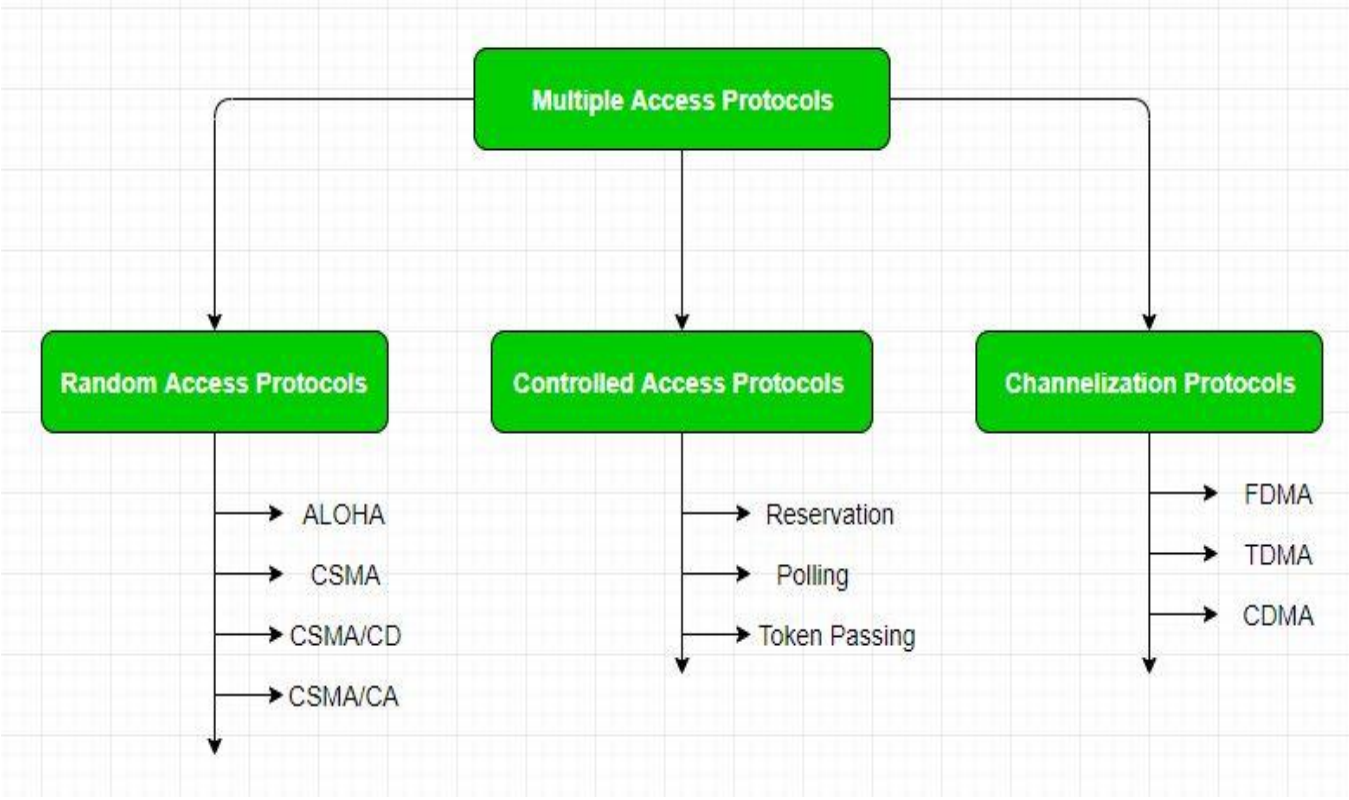
        - With the carrier sense assumption, stations can tell if the channel is in use before trying to use it.
        - No station will attempt to use the channel while it is sensed as busy.
        - If there is no carrier sense, stations can't sense the channel before trying to use it. They just go ahead & transmit. Only later can they determine whether the transmission was successful.
- The **first assumption** says that frame arrivals are independent, both across stations & at particular station, & that frames are generated unpredictably but at a constant rate. This is not a good model of network traffic, as it is well known that packets come in bursts over a range of time scales.
- The **single-channel assumption** is the heart of the model. No external ways to communicate exist. Stations can't raise their hands to request that the teacher call on them, so we will have to come up with better solutions.
- The remaining three assumptions depend on the engineering of the system, & we will say which assumptions hold when we examine a particular protocol.
- The **collision assumption** is basic. Stations need some way to detect collisions if they are to retransmit frames rather than let them be lost.
    - For *wired channels*, node hardware can be designed to detect collisions when they occur. The stations can then terminate their transmissions prematurely to avoid wasting capacity.
    - This detection is much harder for *wireless channels*, so collisions are usually inferred after the fact by the lack of an expected acknowledgement frame.
    - It is also possible for some frames involved in a collision to be successfully received, depending on the details of the signals & the receiving hardware. But this situation is not the common case, so we will assume that all frames involved in a collision are lost.
- The reason for the two alternative **assumptions about time** is that slotted time can be used to improve performance. But it requires the stations to follow a master clock or synchronize their actions with each other to divide time into discrete intervals. Hence, it is not always available. For a given system, only one of them holds.
- Similarly, a network may have **carrier sensing or not** have it.
    - *Wired networks* will generally have carrier sense.
    - *Wireless networks* can't always use it effectively because not every station may be within radio range of every other station.
    - Similarly, carrier sense won't be available in other settings in which a station can't communicate directly with other stations, for example, a

cable modem in which stations must communicate via the cable headend.

- The word "carrier" in this sense refers to a signal on the channel & has nothing to do with the common carriers (e.g., telephone companies).
- No multiaccess protocol guarantees reliable delivery.
- Even in the absence of collisions, the receiver may have copied some of the frame incorrectly for various reasons. Other parts of the link layer or higher layers provide reliability.

# Multiple access protocols

- Many protocols have been defined **<u>to handle the access to shared link</u>**. These protocols are organized in three different groups:
    - Random Access Protocols
    - Controlled Access Protocols
    - Channelization Protocols




## <u>Random Access Protocols</u>

- It is also called **Contention Method**.
- In this method, there is no control station. Any station can send the data.
- The station can make a decision on whether or not to send data. This decision depends on the state of the channel, i.e., channel is busy or idle.

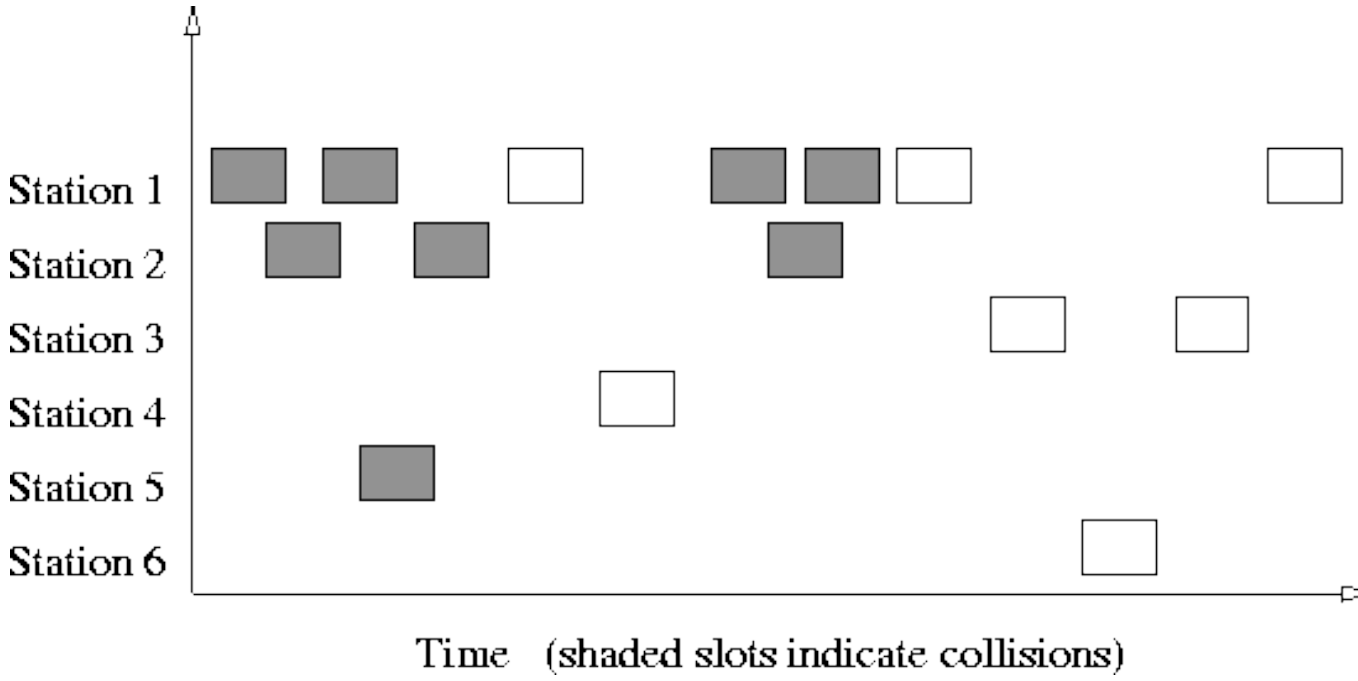
- There is no scheduled time for a station to transmit. They can transmit in random order.
- There is no rule that decides which station should send next. If two stations transmit at the same time, there is collision & the frames are lost.
- The various random-access methods are:
    - ALOHA
    - CSMA (Carrier Sense Multiple Access)
    - CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
    - CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

## ALOHA

- Developed at University of Hawaii in early 1970s by Norman Abramson.
- Used for ground-based radio broadcasting.
- Stations share a common channel.
- When two stations transmit simultaneously, collision occurs & frames are lost.
- There are two different versions of ALOHA:
    - Pure ALOHA
    - Slotted ALOHA

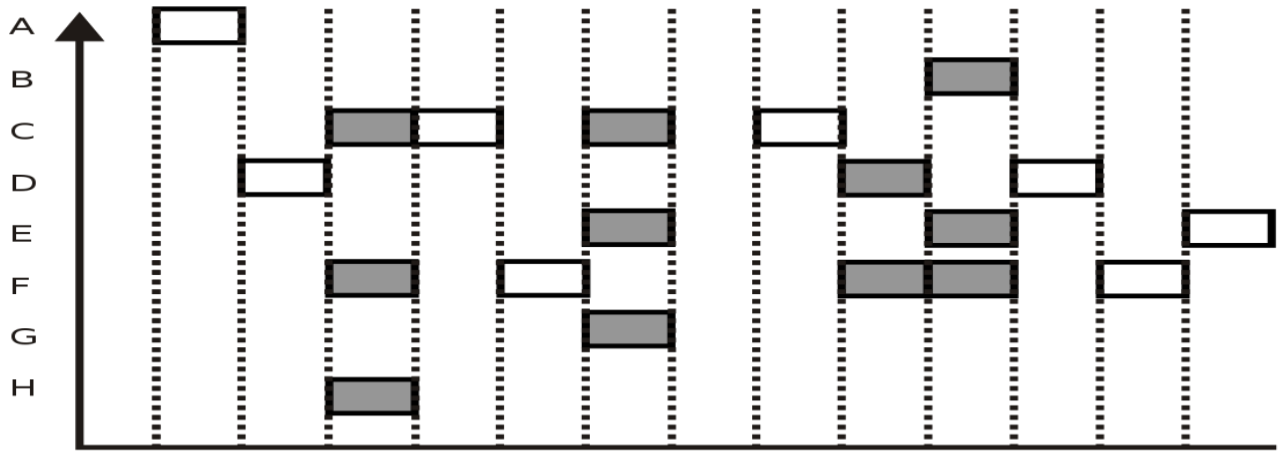
## Pure ALOHA

- In pure ALOHA, stations transmit frames whenever they have data to send.
- When two stations transmit simultaneously, there is collision & frames are lost.
- In pure ALOHA, whenever any station transmits a frame, it expects an acknowledgement from the receiver. If acknowledgement is not received within specified time, the station assumes that the frame has been lost. If the frame is lost, station waits for a random amount of time & sends it again. This waiting time must be random, otherwise, same frames will collide again & again.
- Whenever two frames try to occupy the channel at the same time, there will be collision & both the frames will be lost.
- If first bit of a new frame overlaps with the last bit of a frame almost finished, both frames will be lost & both will have to be retransmitted.



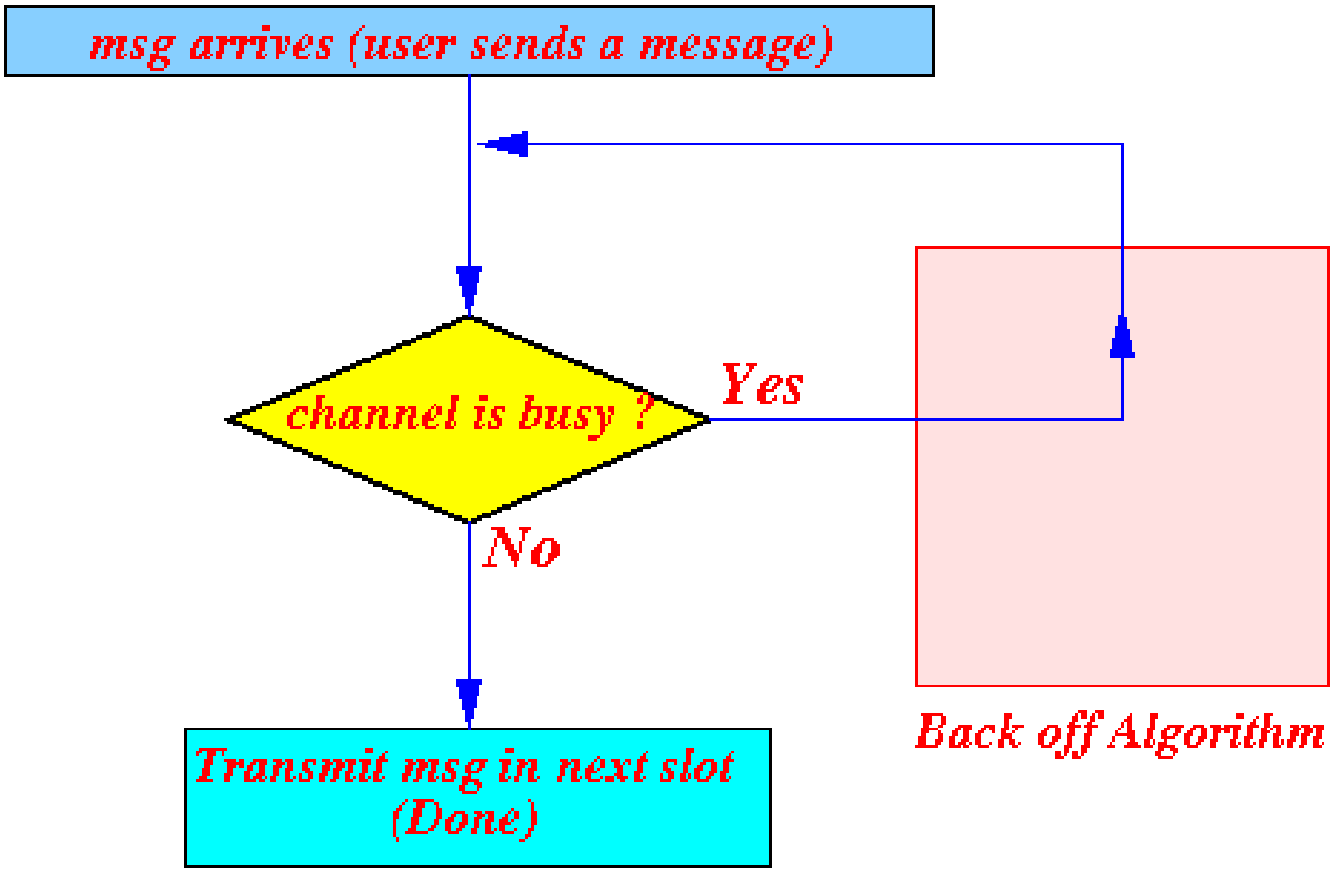

## Slotted ALOHA

- Invented to improve the efficiency of pure ALOHA.
- In slotted ALOHA, time of the channel is divided into intervals called **slots**. The station can send a frame only at the beginning of the **slot** & only one frame is sent in each **slot**.
- If any station is not able to place the frame onto the channel at the beginning of the slot, it has to wait until the next time slot.
- There is still a possibility of collision if two stations try to send at the beginning of the same time slot.




Slotted ALOHA protocol (shaded slots indicate collision)

## Carrier Sense Multiple Access (CSMA)




- Developed to overcome the problems of ALOHA, i.e., to minimize the chances of collision.
- CSMA is based on the principle of "carrier sense". The station senses the carrier or channel before transmitting a frame. It means the station checks whether the channel is idle or busy.
- The chances of collision reduce to a great extent if a station checks the channel before trying to use it.
- The chances of collision still exist because of propagation delay. The frame transmitted by one station takes some time to reach the other station. In the meantime, other station may sense the channel to be idle & transmit its frames. This results in the **collision**.
- There are three different types of CSMA protocols:
    - 1-Persistent CSMA
    - Non-Persistent CSMA
    - P-Persistent CSMA

## 1-Persistent CSMA

- In this method, station that wants to transmit data, continuously senses the channel to check whether the channel is idle or busy. If the channel is busy, station waits until it becomes idle.
- When the station detects an idle channel, it immediately transmits the frame.
- This method has the highest chance of collision because two or more stations may find channel to be idle at the same time & transmit their frames.

### Non-Persistent CSMA

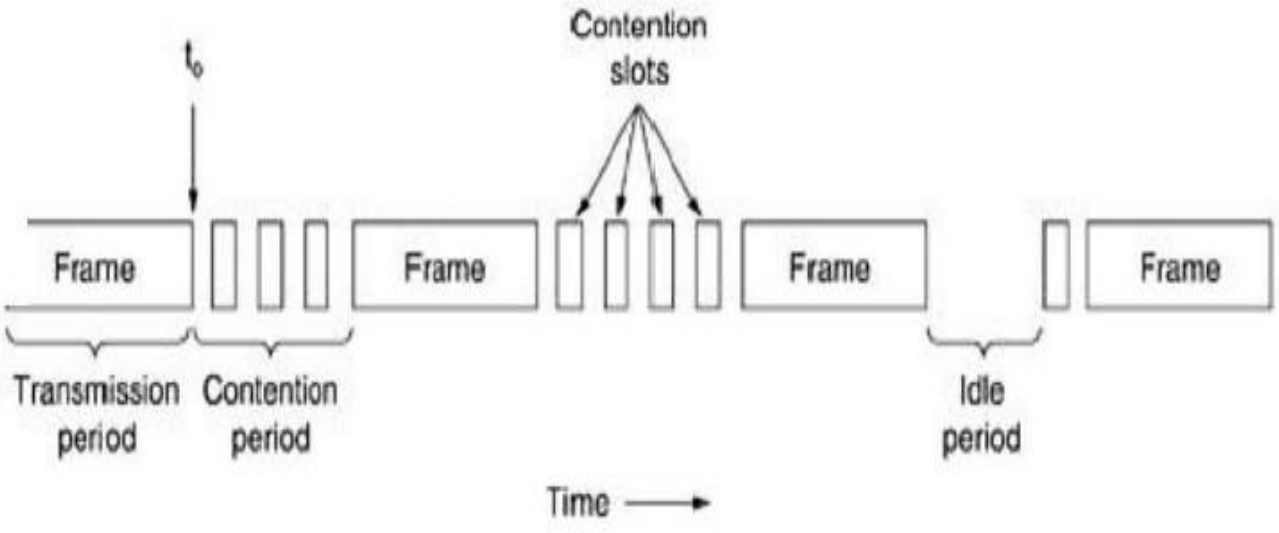
- A station that has a frame to send senses the channel. If the channel is idle, it sends immediately. If the channel is busy, it waits a random amount of time & then senses the channel again.
- It reduces the chance of collision because the stations wait for a random amount of time.
- It is unlikely that two or more stations will wait for the same amount of time & will retransmit at the same time.

### P-Persistent CSMA

- In this method, the channel has time slots such that the time slot duration is equal to or greater than the maximum propagation delay time.
- When a station is ready to send, it senses the channel. If the channel is busy, station waits until next slot. If the channel is idle, it transmits the frame.
- It reduces the chance of collision & improves the efficiency of the network.

## CSMA/CD

- CSMA with Collision Detection (CSMA/CD)
- Persistent & Non-persistent CSMA protocols ensure that no station begins to transmit when it senses the channel busy
- Another improvement for stations is to abort their transmissions as soon as they detect a collision
- If two stations sense the channel to be idle & begin transmitting simultaneously, they will both detect the collision almost immediately. Rather than finish transmitting their frames, they should stop transmitting as soon as the collision is detected.
- Advantage:
    - Saves time & bandwidth
- Widely used on LANs in the MAC sublayer

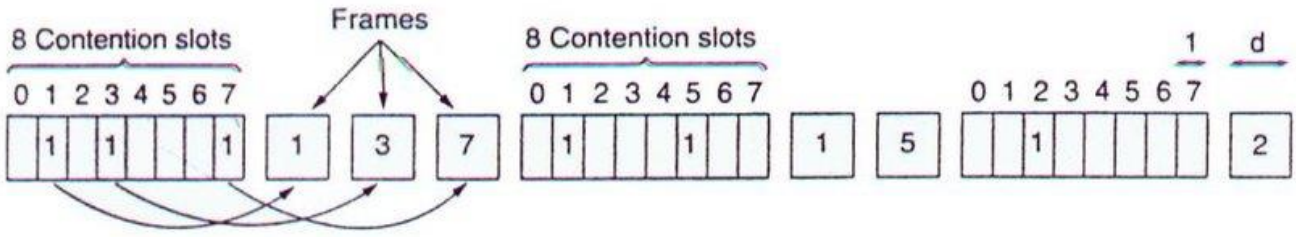



## Collision-Free Protocols

- Even with CSMA/CD, collision can occur during the contention period (when a node can transmit data at any time), especially when the cable is long & the frames are short.
- So, **collision free Protocols** are required:
  - Bit-Map Protocol
  - Binary Countdown

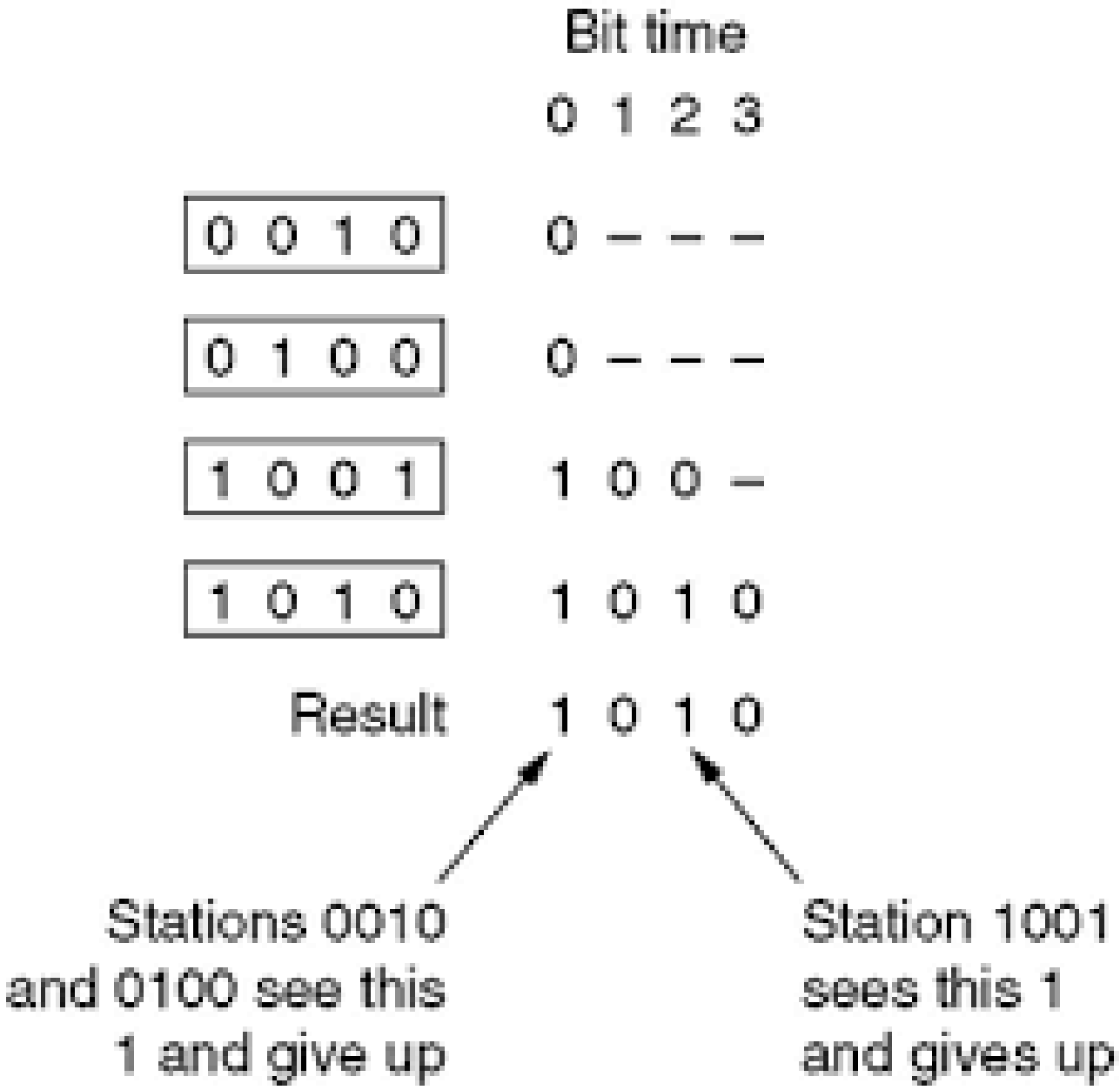
## Bit-Map Protocol

- Follows the basic bit-map method
  - Each contention period consists of exactly N slots.
  - If station 0 has a frame to send, it transmits a 1 bit during the zeroth slot.
  - No other station is allowed to transmit during this slot.
  - Regardless of what station 0 does, station 1 gets the opportunity to transmit a 1 during slot 1, but only if it has a frame queued.
- In general, station j may announce that it has a frame to send by inserting a 1 bit into slot j.
- After all N slots have passed by, each station has complete knowledge of which stations wish to transmit.
- At that point, they begin transmitting in numerical order. Since everyone agrees on who goes next, there won't be any collisions
- Protocols in which the desire to transmit is broadcasted before the actual transmission are called **reservation protocols**.




## Binary Countdown

- Problem with the basic bit-map protocol, & by extension token passing, is that the overhead is 1 bit per station, so it doesn't scale well to networks with thousands of stations.
- We can do better than that by using binary station addresses with a channel that combines transmissions. **A station wanting to use the channel now broadcasts its address as a binary bit string, starting with the high order bit.**
- All addresses are assumed to be the same length. The bits in each address position from different stations are BOOLEAN ORed together by the channel when they are sent at the same time. We will call this protocol **binary countdown**.
- It implicitly assumes that the transmission delays are negligible so that all stations see asserted bits essentially instantaneously.
- To avoid conflicts, an arbitration rule must be applied:
    - As soon as a station sees that a high-order bit position that is 0 in its address has been overwritten with a 1, it gives up.
    - For example, if stations 0010, 0100, 1001, & 1010 are all trying to get the channel, in the first bit time the stations transmit 0, 0, 1, & 1, respectively.
    - These are ORed together to form a 1.
    - Stations 0010 & 0100 see the 1 & know that a higher-numbered station is competing for the channel, so they give up for the current round. Stations 1001 & 1010 continue.
    - The next bit is 0, & both stations continue. The next bit is 1, so station 1001 gives up.
    - The winner is station 1010 because it has the highest address. After winning the bidding, it may now transmit a frame, after which another bidding cycle starts.
- The protocol is illustrated here:




- It has the property that higher-numbered stations have a higher priority than lower-numbered stations, which may be either good or bad, depending on the context.
- The channel efficiency of this method is $d / (d + \log_2 N)$. If, however, the frame format has been cleverly chosen so that the sender's address is the first field in the frame, even these $\log_2 N$ bits are not wasted, & the efficiency is 100%.
- Binary countdown is an example of a simple, elegant, & efficient protocol that is waiting to be rediscovered. Hopefully, it will find a new home someday.

# Limited Contention Protocols

- Media access control (MAC) protocols
- Combines the advantages of collision-based protocols & collision free protocols.
- Behave like **slotted ALOHA under light loads** & **bitmap protocols under heavy loads**.
- When more than one station tries to transmit simultaneously via a shared channel, the transmitted data is garbled, causing an event called **collision**.
- In **collision-based protocols** like **ALOHA**, all stations are permitted to transmit a frame without trying to detect whether the transmission channel is idle or busy.
- In slotted ALOHA, the shared channel is divided into a number of discrete time intervals called **slots**. Any station having a frame can start transmitting at the beginning of a slot. Since, this works very good under light loads, **limited contention protocols behave like slotted ALOHA under low loads**.
- But, with the increase in loads, there occurs exponential growth in number of collisions & so the performance of slotted ALOHA degrades rapidly. So, **under high loads, collision free protocols like bitmap protocols** work best.
- In collision free protocols, channel access is resolved in the contention period & so the possibilities of collisions are eliminated.
- In bit map protocol, the contention period is divided into N slots, where N is the total number of stations sharing the channel. If a station has a frame to send, it sets the corresponding bit in the slot. So, before transmission, each station knows whether the other stations want to transmit.
- Collisions are avoided by mutual agreement among the contending stations on who gets the channel.
- An example of limited contention protocol is **Adaptive Tree Walk Protocol**.

## Working Principle of Limited Contention Protocol

- Limited contention protocols divide the contending stations into groups, which may or not be disjoint.
- At slot 0, only stations in group 0 can compete for channel access.
- At slot 1, only stations in group 1 can compete for channel access & so on.
- In this process, if a station successfully acquires the channel, then it transmits its data frame.
- If there is a collision or there are no stations competing for a given slot in a group, the stations of the next group can compete for the slot.

**Adaptive Tree Walk Protocol**

- A technique to transmit data over shared channels that combines the advantages of collision-based protocols & collision free protocols.
- In adaptive tree walk protocol, the stations are partitioned into groups in a hierarchical manner.
- The contention period is divided into discrete time slots, & for each slot the contention rights of the stations are limited.
- Under light loads, all the stations can participate for contention each slot like ALOHA.
- But, under heavy loads, only a group can try for a given slot.

# Wireless LAN Protocols

- Wireless LANs refer to LANs (Local Area Networks) that use high frequency radio waves instead of cables to connect devices.
- It can be conceived as a set of laptops & other wireless devices communicating by radio signals.
- Users connected by WLANs can move around within the area of network coverage. Most WLANs are based upon the standard IEEE 802.11 or Wi-Fi.
- Each station in a Wireless LAN has a wireless network interface controller. A station can be of **two categories**:
    - **Wireless Access Point (WAP) or Access Points (AP)**
        - Wireless routers that form the base stations or access points.
        - APs are wired together using fibre or copper wires, through the distribution system.
    - **Client**
        - Clients are workstations, computers, laptops, printers, smart phones etc.
        - They are around tens of metres within the range of an AP

**Types of WLAN Protocols**

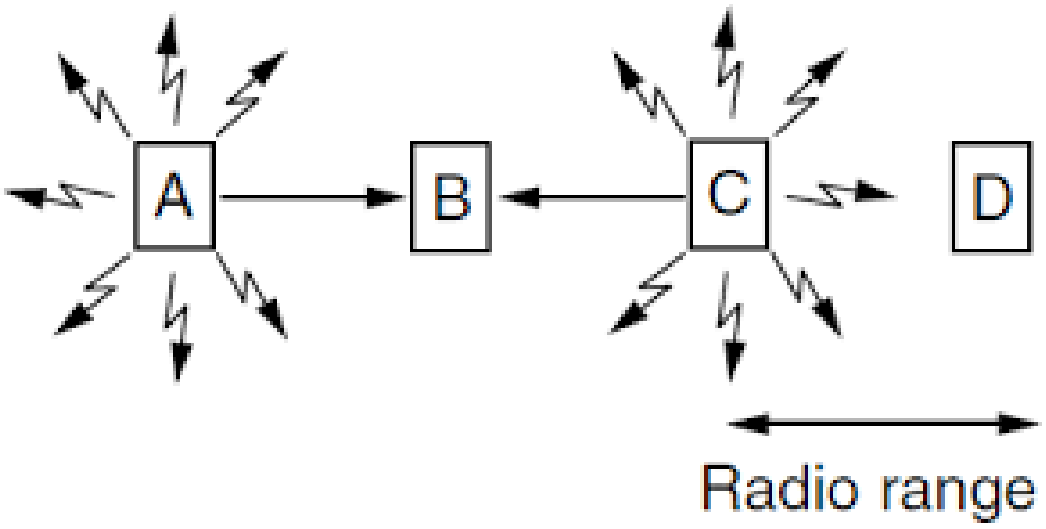
IEEE 802.11 or Wi-Fi has a number of variations, the main among which are

- ***802.11a Protocol***
    - Supports very high transmission speeds of 54Mbps
    - Has a high frequency of 5GHz range, due to which signals have difficulty in penetrating walls & other obstructions
    - Employs **Orthogonal Frequency Division Multiplexing (OFDM).**
- ***802.11b Protocol***

- Operates within the frequency range of 2.4GHz & supports 11Mbps speed.
- Facilitates path sharing.
- Less vulnerable to obstructions.
- Uses **Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)** with Ethernet protocol.

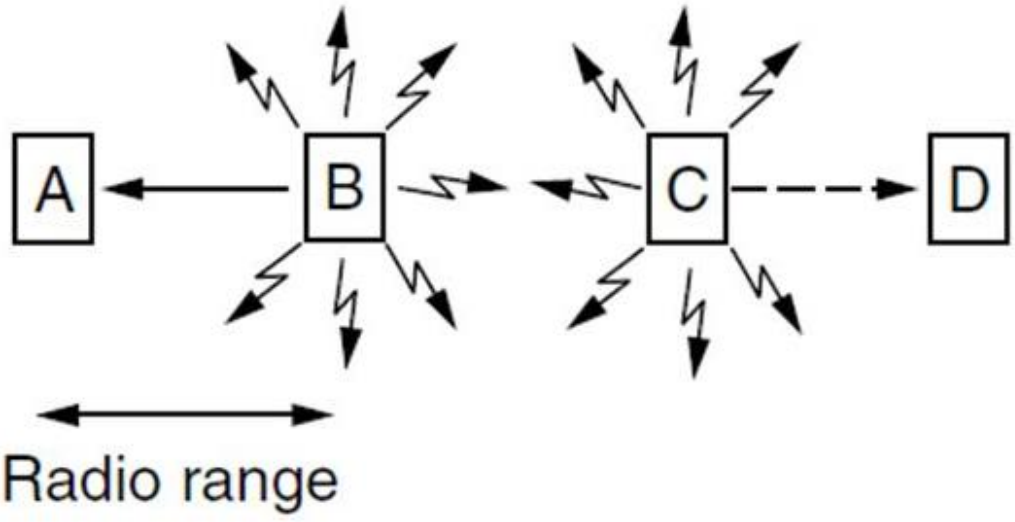
- ***802.11g Protocol***
  - Combines the features of 802.11a & 802.11b protocols.
  - Supports both the frequency ranges 5GHz (as in 802.11a standard) & 2.4GHz (as in 802.11b standard).
  - Owing to its dual features, 802.11g is backward compatible with 802.11b devices. 802.11g provides high speeds, varying signal range, & resilience to obstruction. But it is more expensive for implementation.
- ***802.11n Protocol***
  - Popularly known as Wireless N.
  - An upgraded version of 802.11g.
  - Provides very high bandwidth up to 600Mbps & provides signal coverage.
  - Uses Multiple Input/Multiple Output (MIMO), having multiple antennas at both the transmitter end & receiver ends.
  - In case of signal obstructions, alternative routes are used. But the implementation is highly expensive.

## Hidden Terminal & Exposed Terminal Problem




- Suppose A, B, C, & D are 2 different nodes in a network. Let A & C transmit to B, as depicted in the above figure.

- If A sends & then C immediately senses the medium, it won't hear A because A is out of range. Thus, C will falsely conclude that it can transmit to B.
- If C starts transmitting, it will interfere at B, wiping out the frame from A. (No CDMA-type scheme is used to provide multiple channels, so collisions garble the signal & destroy both frames).
- This **problem of a station not being able to detect a potential competitor for the medium because the competitor is too far away is called the <u>hidden terminal problem</u>**

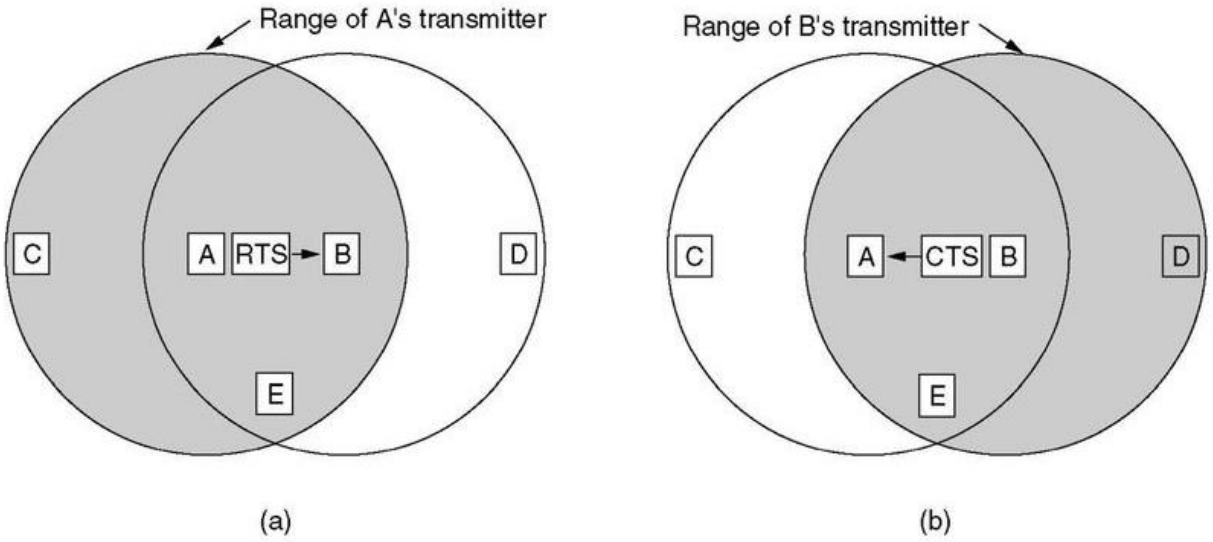



- Now, look at a different situation: B transmitting to A, at the same time that C wants to transmit to D, as shown in the above figure.
- If C senses the medium, it will hear a transmission & falsely conclude that it may not send to D (shown as a dashed line).
- In fact, such a transmission would cause bad reception only in the zone between B & C, where neither of the intended receivers is located.
- This problem is called the **<u>exposed terminal problem</u>**.

## <u>MACA (Multiple Access with Collision Avoidance)</u>

- A protocol that tackles the above-mentioned problems for wireless LANs is **MACA (Multiple Access with Collision Avoidance).**
- Sender stimulates by sending **RTS (Request to Send)** to the receiver into outputting a short frame
- **CTS (Clear to Send)** is transmitted by receiver so stations nearby can detect this transmission & avoid transmitting for the duration of the upcoming (large) data frame

## MACA PROTOCOL




The MACA protocol. (a) A sending an RTS to B.
(b) B responding with a CTS to A.

# Ethernet

- IEEE 802.3 Local Area Network (LAN) Protocols
- Ethernet protocols refer to the family of local-area network (LAN) covered by the IEEE 802.3.
- In the Ethernet standard, there are two modes of operation:
  - Half-duplex
  - Full-duplex modes
- In half duplex mode, data are transmitted using **Carrier-Sense Multiple Access/Collision Detection (CSMA/CD)** protocol on a shared medium.
- Main disadvantages of the half-duplex are the efficiency & distance limitation, in which the link distance is limited by minimum MAC frame size. This restriction reduces efficiency for high-rate transmission. So, the carrier extension technique is used to ensure minimum frame size of 512 bytes in Gigabit Ethernet to achieve a reasonable link distance.
- Four data rates are currently defined for operation over optical fibre & twisted-pair cables:
  - **10 Mbps: 10Base-T Ethernet (IEEE 802.3)**

- **100 Mbps: Fast Ethernet (IEEE 802.3u)**
- **1000 Mbps: Gigabit Ethernet (IEEE 802.3z)**
- **10-Gigabit: 10 Gbps Ethernet (IEEE 802.3ae)**

- The Ethernet System consists of **three basic elements**:
  1. A physical medium used to carry Ethernet signals between computers
  2. A set of medium access control rules embedded in each Ethernet interface that allow multiple computers to fairly arbitrate access to the shared Ethernet channel
  3. An Ethernet frame that consists of a standardized set of bits used to carry data over the system
- ISO data link layer is divided into two IEEE 802 sub-layers, the **Media Access Control (MAC) sub-layer** & the **MAC-client sub-layer**.
- The IEEE 802.3 physical layer corresponds to ISO physical layer.
- Each Ethernet-equipped computer operates independently of all other stations on the network. There is no central controller.
- All stations attached to an Ethernet are connected to a shared signalling system, also called the **medium**. To send a data, a station first listens to the channel, & when the channel is idle the station transmits its data in the form of an **Ethernet frame, or packet**.
- After each frame transmission, all stations on the network must contend equally for the next frame transmission opportunity.
- Access to the shared channel is determined by the **Medium Access Control (MAC) mechanism** embedded in the Ethernet interface located in each station. The medium access control mechanism is based on a system called **Carrier Sense Multiple Access with Collision Detection (CSMA/CD).**
- As each Ethernet frame is sent onto the shared signal channel, all Ethernet interfaces look at the destination address. If the destination address of the frame matches with the interface address, the frame will be read entirely & be delivered to the networking software running on that computer. All other network interfaces will stop reading the frame when they discover that the destination address doesn't match their own address.
- IEEE 802.3 is a working group & a collection of **Institute of Electrical & Electronics Engineers (IEEE) standards** produced by the working group defining the physical layer & data link layer's media access control (MAC) of wired Ethernet.
- This is generally a Local area network (LAN) technology with some wide area network (WAN) applications.
- Physical connections are made between nodes and/or infrastructure devices (hubs, switches, routers) by various types of copper or fibre cable.

- 802.3 is a technology that supports the IEEE 802.1 network architecture.
- 802.3 also defines LAN access method using CSMA/CD.
- The IEEE 802.3 is for **1 persistent CSMA/CD (Carrier sense multiple access with collision detection) LAN**. Here, when a station wants to transmit, it listens to the cable. If the cable is busy, the station waits until it goes ideal, otherwise it transmits immediately.
- If 2 or more stations simultaneously begin transmitting on an ideal cable, they will collide. All colliding stations then terminate their transmission, wait a random time, & repeat the whole process all over again. So, the data is sent when the carrier is free.
- Since the name "Ethernet" refers to the cable, two types of coaxial cables are used.
    - **Thick Ethernet**
        - Resembles a yellow garden hose, with markings every 2.5 to show where the taps go
    - **Thin Ethernet**
        - Small, flexible, & cheaper

# Ethernet cabling

**Four types of cabling** are commonly used:

| Name | Cable | Max. Seg. | Nodes/Seg. | Advantages |
|---|---|---|---|---|
| 10 Base 5 | Thick coax | 500 m | 100 | Original cable; Now old-fashioned |
| 10 Base 2 | Thin coax | 185 m | 30 | No hub needed |
| 10 Base-T | Twisted Pair | 100 m | 1024 | Cheapest System |
| 10 Base-F | Fibre Optics | 2000 m | 1024 | Best between buildings |

## 10Base5 cabling

- Popularly called thick Ethernet, came first
- Resembles a yellow garden hose, with markings every 2.5 meters to show where the taps go.
- Connections to it are generally made using vampire taps, in which a pin is very carefully forced halfway into the coaxial cable's core.
- A transceiver is clamped securely around the cable so that its tap makes contact with the inner core. The transceiver contains the electronics that handle carrier detection & collision detection. When a collision is detected,

the transceiver also puts a special invalid signal on the cable to ensure that all other transceivers also realize that a collision has occurred.

- The notation **10Base5** means that it **operates at 10 Mbps, uses baseband signalling, & can support segments of up to 500 meters**.
    - The first number is the speed in Mbps.
    - Then comes the word "Base" to indicate baseband transmission. There used to be a broadband variant, 10Broad36, but it never caught on in the marketplace & has since vanished.
    - Finally, if the medium is coax, its length is given rounded to units of 100 m after "Base".

**10Base2 cabling**

- The 2nd cable type
- Thin Ethernet
- Bends easily
- Connections to it are made using industry standard BNC connectors to form T junctions, rather than using vampire taps.
- BNC connectors are easier to use & more reliable.
- Thin Ethernet is much cheaper & easier to install, but it can run for only 185 meters per segment, each of which can handle only 30 machines.

**10Base-T cabling**

- The problems associated with finding cable breaks led to a different kind of wiring pattern, in which all stations have a cable running to a central hub in which they are all connected electrically (as if they were soldered together).
- Usually, these wires are telephone company twisted pairs, since most office buildings are already wired this way, & normally plenty of spare pairs are available. This scheme is called **10Base-T**. Hubs don't buffer incoming traffic.

**10Base-F cabling**

- A fourth cabling option
- Uses fibre optics
- Expensive due to the cost of the connectors & terminators, but has an excellent noise immunity & is the method of choice when running between buildings or widely-separated hubs.
- Runs of up to km are allowed
- Also offers good security since wiretapping fibre is much more difficult than wiretapping copper wire.

# Ethernet Frame Format

| 7 | 1 | 6 | 6 | 2 | 0-1500 | 0-46 | 4 |
|---|---|---|---|---|---|---|---|
| Preamble | SFD | Destination Address | Source Address | Length of Data | Data Field | Padding | FCS |

- Preamble
  - Used for synchronization, i.e., synchronization of receiver's clock with sender's
  - 7bytes
- SFD (Start Frame Delimit)
  - Used to show the start of the frame
- Destination address & Source address
  - Used to denote the destination & source of the data
  - There is group address which sends data to multiple stations
  - Sending data to group of stations is known as multicast
  - The address consisting of all 1 bits is reserved for broadcast
- Length of data & Data field
  - Tells how many bytes of data are present in the data field, from a minimum of 0 to maximum of 1500
  - While a data field of 0 byte is illegal, it causes problem.
  - When transceiver detects collision, it cuts current frame due to this stray bits & pieces of frame appear on the cable.

Min size of Ethernet frame =64 bytes

| 6 | 6 | 2 | 0 | 46 | 4 |
|---|---|---|---|---|---|

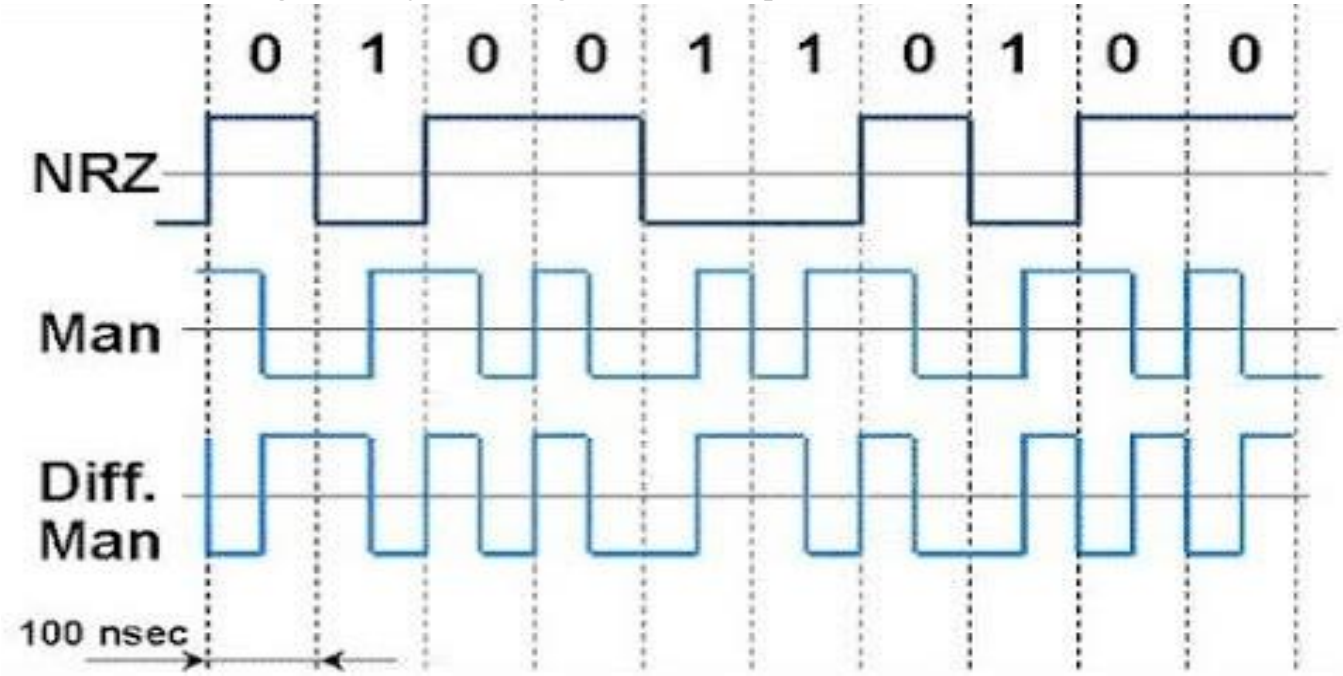
Max size of Ethernet frame =1518 bytes

| 6 | 6 | 2 | 1500 | 0 | 4 |
|---|---|---|---|---|---|

- Padding
  - Used to fill out frame to minimum size
  - Extra 0's added to data
  - If the data send is 12 bits, we make it 2 bytes by adding 4 0's.
- FCS (Frame Check Sequence)
  - Used to check errors

# Manchester encoding

- None of the versions of Ethernet uses straight binary encoding with 0 volts for a 0 bit & 5volts for a 1 bit because it leads to ambiguities.
- If one station sends the bit string 0001000, others might falsely interpret it as 10000000 or 01000000 because they can't tell the difference between an idle sender (0 volts) & a 0 bit (0 volts).
- This problem can be solved by using +1 volts for a 1 & -1 volts for a 0, but there is still the problem of a receiver sampling the signal at a slightly different frequency than the sender used to generate it.
- Different clock speeds can cause the receiver & sender to get out of synchronization about where the bit boundaries are, especially after a long run of consecutive 0s or a long run of consecutive 1s.
- What is needed is a way for receivers to unambiguously determine the start, end, or middle of each bit without reference to an external clock.
- Two such approaches are called **Manchester encoding & Differential Manchester encoding.**
- With Manchester encoding, each bit period is divided into two equal intervals.
- **A binary 1 bit** is sent by having the voltage set high during the first interval & low in the second one **(high to low**).
- **A binary 0** is just the reverse: first low and then high (**low to high**).
- This scheme ensures that every bit period has a transition in the middle, making it easy for the receiver to synchronize with the sender.
- A disadvantage of Manchester encoding is that it requires twice as much bandwidth as straight binary encoding because the pulses are half the width.



# Ethernet MAC sublayer protocol

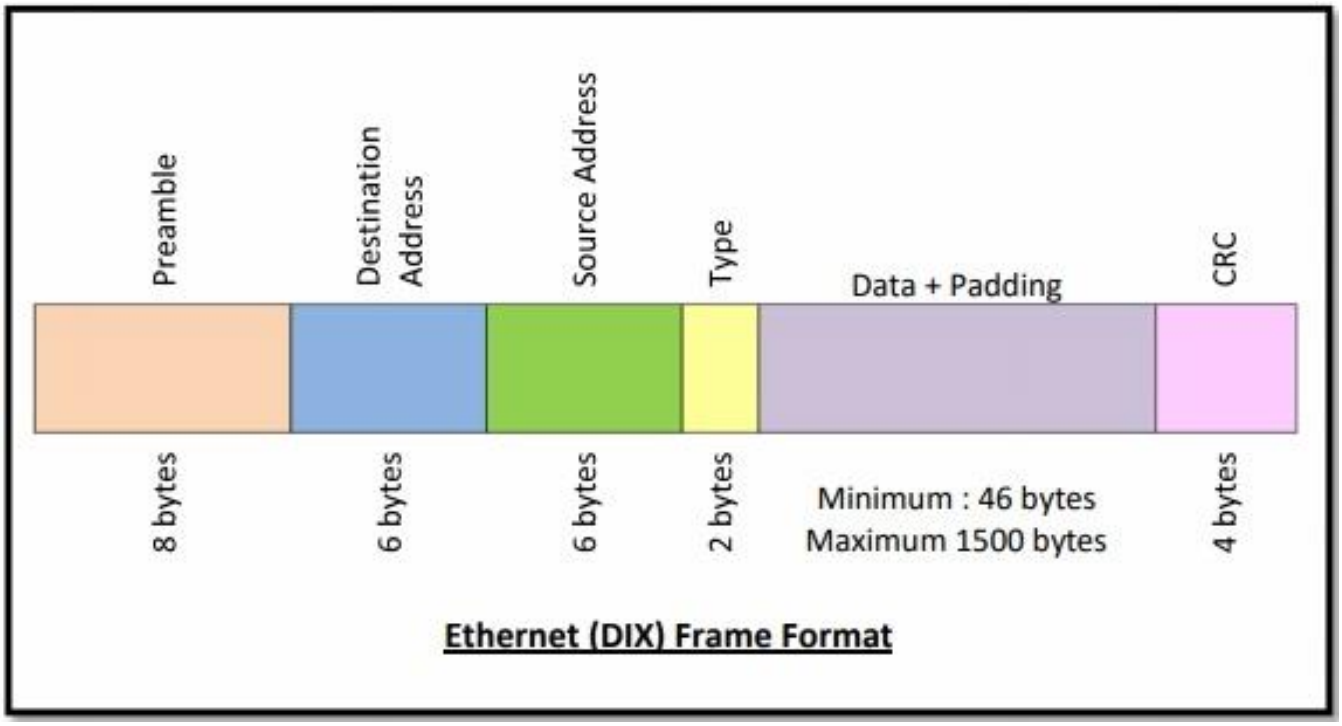
- Classic Ethernet is the original form of Ethernet used primarily in LANs.
- It provides data rates between 3 to 10 Mbps.
- It operates both in physical layer & in MAC sublayer of OSI model.
- In the physical layer, the features of the cables & networks are considered.
- In MAC sublayer, the frame formats for the Ethernet data frame are laid down.
- Classic Ethernet was first standardized in 1980s as IEEE 802.3 standard.

**Frame Format of Classic Ethernet**

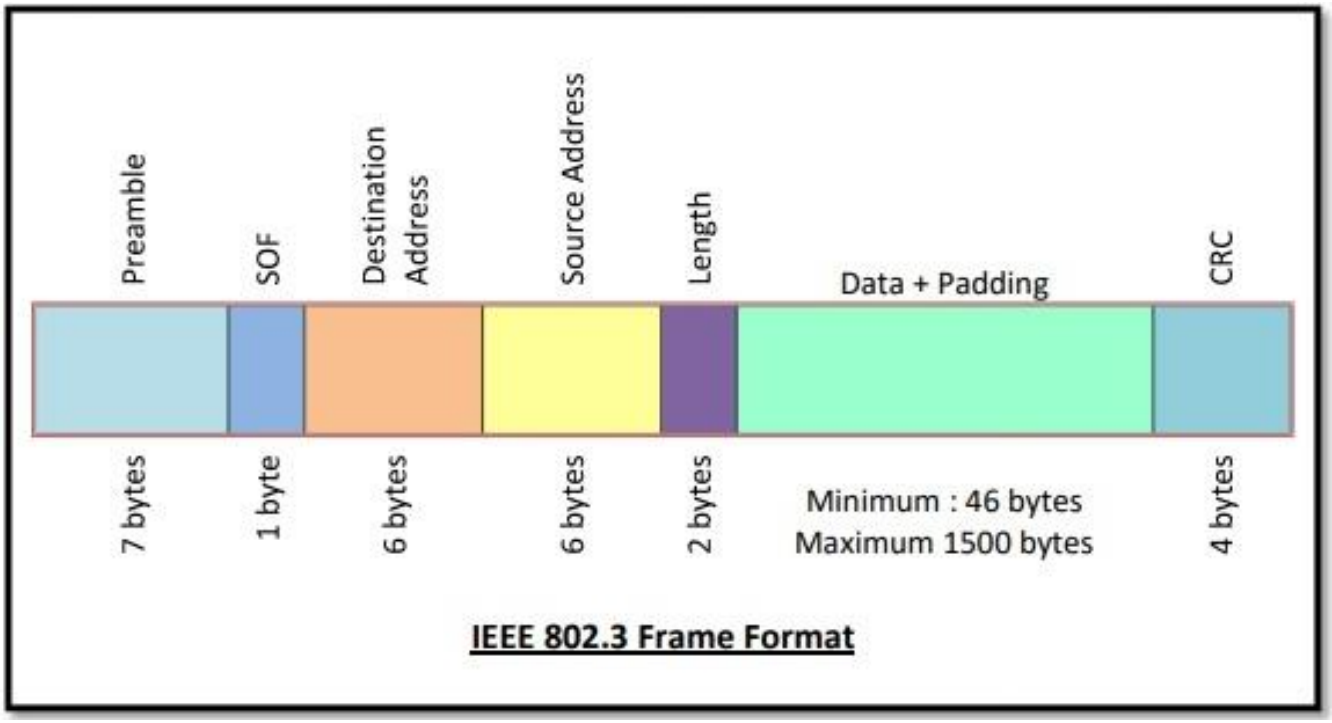
- Classic Ethernet frames can be either of Ethernet (DIX) or of IEEE 802.3 standard.
- The frames of the two standards are very similar except for one field.
- The main fields of a frame of classic Ethernet are −
    - **Preamble**
        - It is the starting field that provides alert & timing pulse for transmission.
        - In case of Ethernet (DIX), it is an 8-byte field & in case of IEEE 802.3 it is of 7 bytes.
    - **Start of Frame Delimiter (SOF)**
        - It is a 1-byte field in an IEEE 802.3 frame that contains an alternating pattern of ones & zeros ending with two ones.
    - **Destination Address**
        - It is a 6-byte field containing physical address of destination stations.
    - **Source Address**
        - It is a 6-byte field containing the physical address of the sending station.
    - **Type/Length**
        - This is a 2-byte field.
        - In case of Ethernet (DIX), the field is type that instructs the receiver which process to give the frame to.
        - In case of IEEE 802.3, the field is length that stores the number of bytes in the data field.
    - **Data**
        - This is a variable sized field carries the data from the upper layers.
        - The maximum size of data field is 1500 bytes.
    - **Padding**

- This is added to the data to bring its length to the minimum requirement of 46 bytes.

- **CRC**
  - CRC stands for cyclic redundancy check.
  - It contains the error detection information.



**Ethernet (DIX) Frame Format**



**IEEE 802.3 Frame Format**

# Binary Exponential Backoff algorithm

A collision resolution mechanism used in random access MAC protocols. This algorithm is used in Ethernet (IEEE 802.3) wired LANs.

- In Ethernet networks, this algorithm is commonly **used to schedule retransmissions after collisions**.
- After a collision, time is divided into discrete slots whose length is equal to $2\tau$, where $\tau$ is the maximum propagation delay in the network.
- The reason for this choice is that $2\ \tau$ is the minimum amount of time a source needs to listen to the channel to always detect a collision.
- The stations involved in the collision randomly pick an integer from the set {0,1}. This set is called the **contention window**.
- If the sources collide again because they picked the same integer, the contention window size is doubled & it becomes {0,1,2,3}. Now the sources involved in the second collision randomly pick an integer from the set {0,1,2,3} & wait that number of slot times before trying again. Before they try to transmit, they listen to the channel & transmit only if the channel is idle. This causes the source which picked the smallest integer in the contention window to succeed in transmitting its frame.
- In general, after collisions, a random number between 0 & $2\tau$ is chosen.
- After a station detects collision, it aborts its transmission in the slot duration itself in which it started transmitting.
- In Ethernet, the doubling of the contention window stops after 10 collisions & the contention window remains {0, 1...,1023}.
- After 16 collisions, the process is aborted & the source stops trying.

# Ethernet performance

- Ethernet is a set of technologies & protocols that are used primarily in LANs.
- The performance of Ethernet is analysed by computing the efficiency of the channel under different load conditions.
- Let us assume an Ethernet network has k stations & each station transmits with a probability p during a contention slot. Let A be the probability that some station acquires the channel. A is calculated as:

$A = kp\ (1-p)\ kp$

- The value of A is maximized at $p = 1/k$. If there can be innumerable stations connected to the Ethernet network, i.e., $k \rightarrow \infty$, the maximum value of A will be 1/e.

- Let Q be the probability that the contention period has exactly j slots. Q is calculated as:

$Q = A\,(1-A)\,j-1$

- Let M be the mean number of slots per contention. So, the value of M will be:

$$M = \sum_{j=0}^{\infty} jA\,(1-A)^{j-1} = \frac{1}{A}$$

- Given that τ is the propagation time, each slot has duration 2τ. Hence the mean contention interval, *w* will be 2τ/A.
- Let P be the time in seconds for a frame to propagate.
- The channel efficiency, when a number of stations want to send frame, can be calculated as:

$$Channel\ Efficiency = \frac{P}{P + 2\tau/A}$$

- Let F be the length of frame, B be the cable length, L be the cable length, c be the speed of signal propagation and e be the contention slots per frame. The channel efficiency in terms of these parameters is:

$$Channel\ Efficiency = \frac{1}{1 + 2BLe/cF}$$

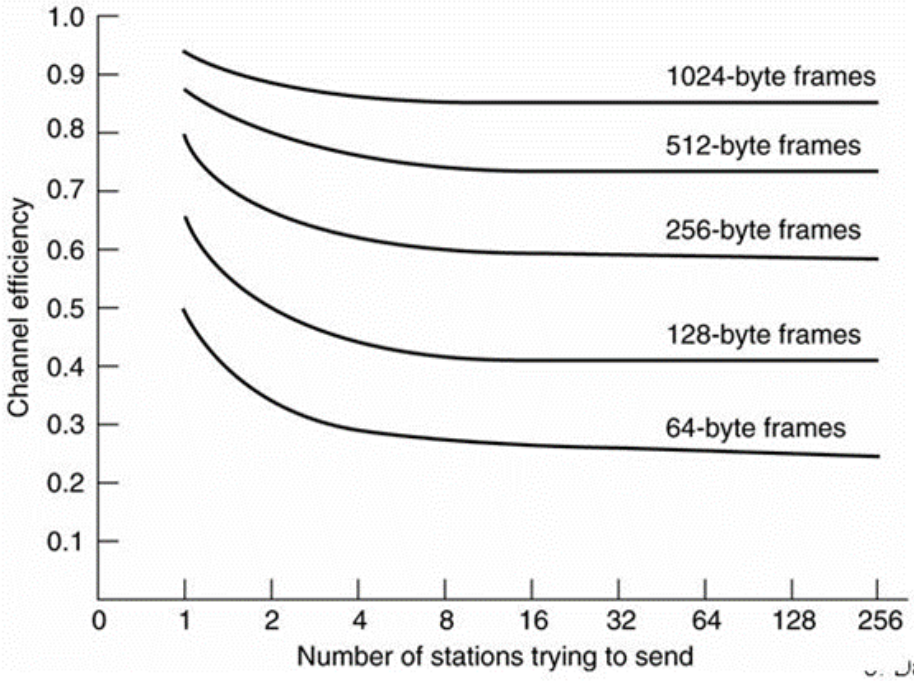
The channel efficiency is plotted versus the number of ready stations for 2τ = 51.2 μsec & a data rate of 10 Mbps, using the above equation.

With a 64- byte slot time, it's not surprising that 64-byte frames are not efficient.

On the other hand, with 1024-byte frames & an asymptotic value of *e* 64-byte slots per contention interval, the contention period is 174 bytes long & the efficiency is 85%.

This result is much better than the 37% efficiency of slotted ALOHA.

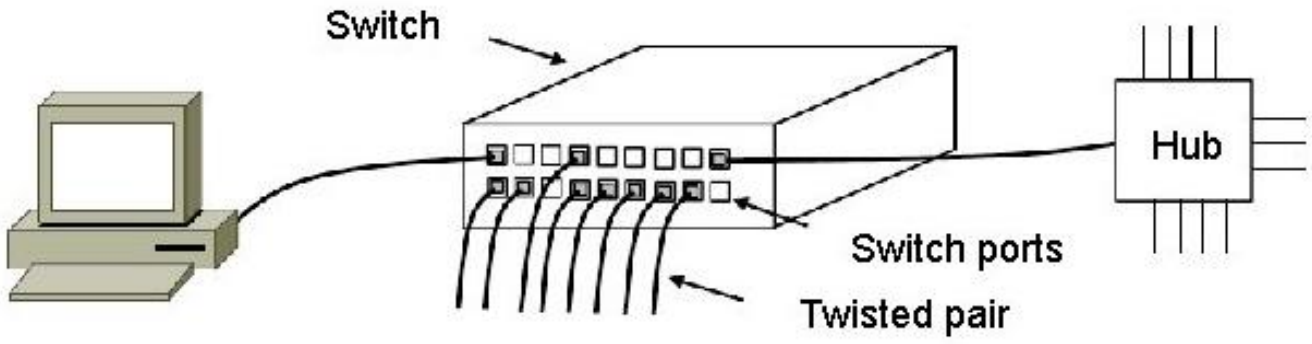
## Efficiency of Ethernet at 10 Mbps with 512-bit slot times.






# Switched Ethernet

- As more & more stations are added to an Ethernet, the traffic will go up. Eventually, the LAN will saturate.
- One way out is to go to a higher speed, say, from 10 Mbps to 100 Mbps. But with the growth of multimedia, even a 100-Mbps or 1-Gbps Ethernet can become saturated.
- There is an additional way to deal with increased load: **switched Ethernet**, as shown in the figure.



- The heart of this system is a switch containing a high-speed backplane & room for typically 4 to 32 plug-in line cards, each containing one to eight connectors.
- Each connector has a 10Base-T twisted pair connection to a single host computer.
- When a station wants to transmit an Ethernet frame, it outputs a standard frame to the switch.
- The plug-in card getting the frame may check to see if it is destined for one of the other stations connected to the same card. If so, the frame is copied there. If not, the frame is sent over the high-speed backplane to the destination station's card.
- The backplane typically runs at many Gbps, using a proprietary protocol.
- What happens if two machines attached to the same plug-in card transmit frames at the same time? It depends on how the card has been constructed.
- One possibility is for all the ports on the card to be wired together to form a local on-card LAN. Collisions on this on-card LAN will be detected & handled the same as any other collisions on a CSMA/CD network—with retransmissions using the binary exponential backoff algorithm.
- With this kind of plug-in card, only one transmission per card is possible at any instant, but all the cards can be transmitting in parallel. With this design, each card forms its own collision domain, independent of the others. With only one station per collision domain, collisions are impossible & performance is improved.
- With the other kind of plug-in card, each input port is buffered, so incoming frames are stored in the card's on-board RAM as they arrive. This design allows all input ports to receive (& transmit) frames at the same time, for parallel, full-duplex operation, something not possible with CSMA/CD on a single channel.
- Once a frame has been completely received, the card can then check to see if the frame is destined for another port on the same card or for a distant port.
- In the former case, it can be transmitted directly to the destination.
- In the latter case, it must be transmitted over the backplane to the proper card.
- With this design, each port is a separate **collision domain**, so collisions don’t occur. The total system throughput can often be increased by an order of magnitude over 10Base5, which has a single collision domain for the entire system.
- Since the switch just expects standard Ethernet frames on each input port, it is possible to use some of the ports as concentrators.

- In the above figure, the port in the upper-right corner is connected not to a single station, but to a 12-port hub. As frames arrive at the hub, they contend for the ether in the usual way, including collisions & binary backoff.
- Successful frames make it to the switch & are treated there like any other incoming frames: they are switched to the correct output line over the high-speed backplane.
- **Hubs** are cheaper than switches, but due to falling switch prices, they are rapidly becoming obsolete. Nevertheless, legacy hubs still exist.

# Fast Ethernet (IEEE 802.3u)

- 100 Mbps bandwidth
- Uses same CSMA/CD media access protocol & packet format as in Ethernet.
- 100BaseTX (UTP) & 100BaseFX (Fibre) standards
- Physical media: -
  - ▪ 100 BaseTX - UTP Cat 5e
  - ▪ 100 BaseFX - Multimode / Single mode Fibre
- Full Duplex/Half Duplex operations.
- Provision for Auto-Negotiation of media speed:

10 Mbps or 100Mbps (popularly available for copper media only).

- Maximum Segment Length
  - 100 Base TX - 100 m
  - 100 Base FX - 2 Km (Multimode Fibre)
  - 100 Base FX - 20 km (Single mode Fibre)

# Gigabit Ethernet

All configurations of the gigabit Ethernet are point-to-point.

It supports 2 different modes of operation:

Full duplex mode

Half duplex mode

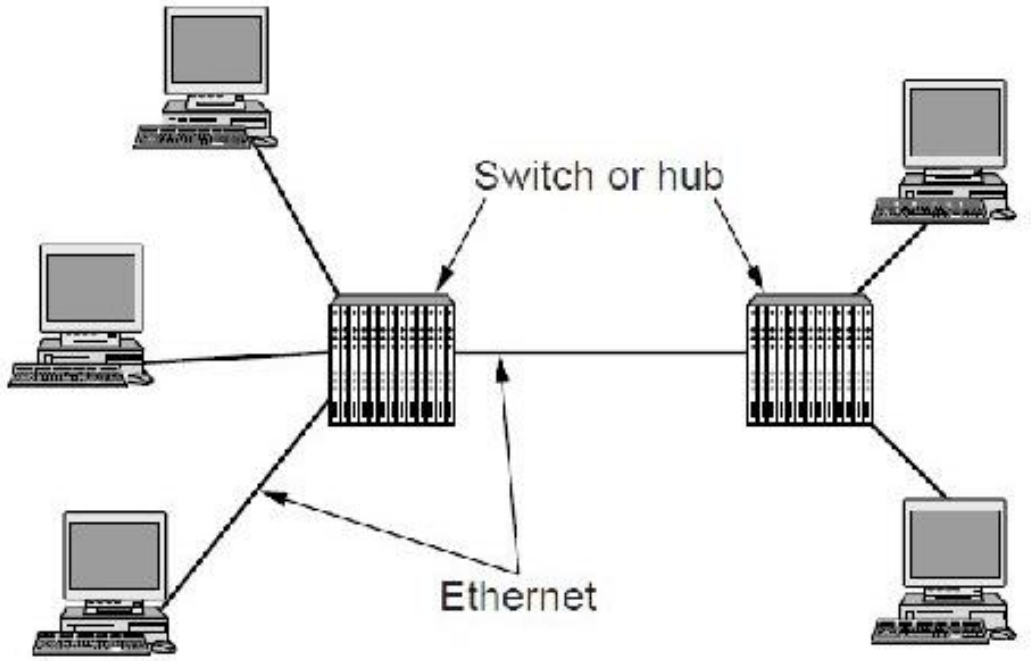
The normal mode is full duplex mode, which allows traffic in both the direction at the same time.

This mode is used when there is a central switch connected to computers on the periphery.

Switches are free to mix & match speed.

Half duplex is used when the computers are connected to a hub rather than a switch.

A hub doesn't buffer incoming frames, instead it electrically connects




A two-station Ethernet

# IEEE 802.2: Logical Link Control

- The IEEE divided Datalink layer into two layers: "above" is the control layer, the logical connection, **Logical Link Control** (LLC), & "below" is the control layer, **Medium Access Control** (MAC).
- The **LLC layer** is standardized by IEEE as 802.2 since the beginning of 1980.
- Its purpose is to allow level 3 network protocols, (for e.g., IP) to be based on a single layer (the LLC layer) regardless of underlying protocol used, including Wi-Fi, Ethernet or Token Ring. All Wi-Fi data packets, so carry a pack LLC, which contains itself packets from the upper network layers.

- The header of a packet LLC indicates the type of layer 3 protocol in it: most of the time, it is IP protocol, but it could be another protocol, such as **IPX (Internet Packet Exchange)** for example.
- With the help of LLC layer, it is possible to have multiple Layer 3 protocols on the same network, at a time. So, LAN nodes use same communication channel for transmission.
- Logical Link Control (LLC) sublayer is responsible for data transmission between computers or devices on a network.
- The function of the Logical Link Control (LLC) is to manage & ensure the integrity of data transmissions. The LLC provides Data Link Layer links to services for the Network Layer protocols. This is accomplished by the LLC Service Access Points (SAPs) for the services residing on network computers. Also, there is an LLC Control field for delivery requests or services.
- The Logical Link Control (LLC) has several service types:
    - Service type 1, is a connectionless service with no establishment of a connection, & an unacknowledged delivery.
    - Service type 2, is a logical connection service with an acknowledgement of delivery.
    - Service type 3, is a connectionless service with an acknowledgement of delivery.

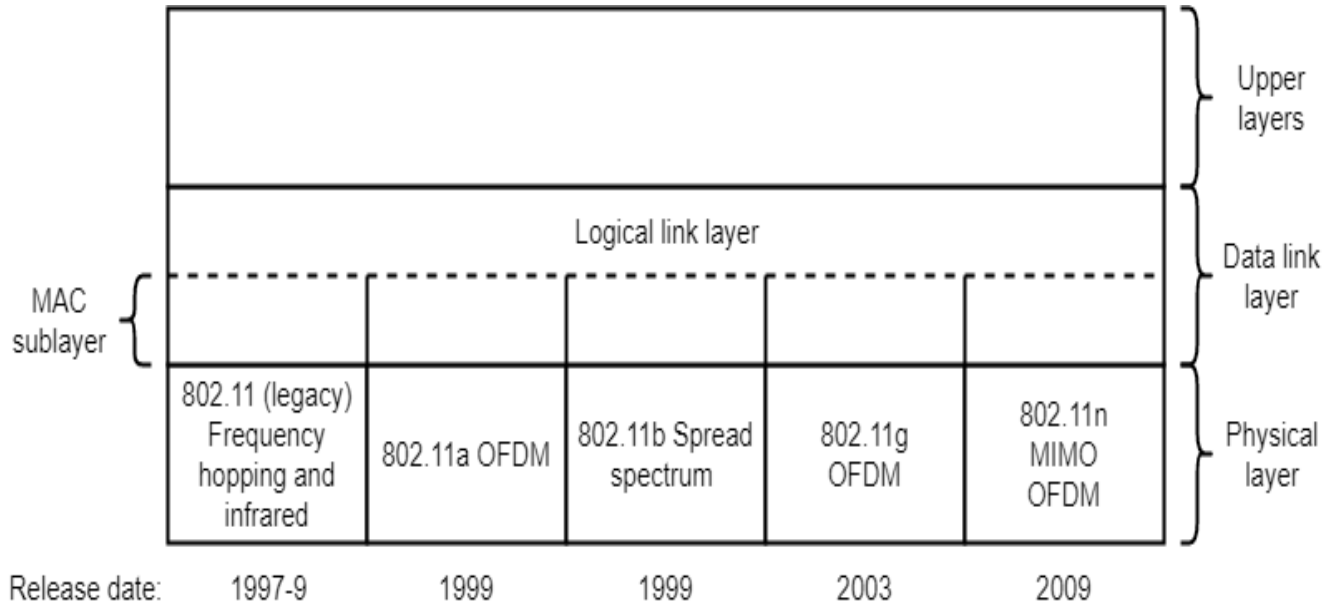
# Wireless LANs

- Wireless LANs are increasingly popular, & homes, offices, cafes, libraries, airports, zoos, & other public places are being outfitted with them to connect computers, PDAs, & smart phones to the Internet.
- Wireless LANs can also be used to let two or more nearby computers communicate without using the Internet.
- The main wireless LAN standard is 802.11.
- A system of notebook computers that communicate by radio can be regarded as a wireless LAN
- A common configuration for a wireless LAN is an office building with base stations (also called access points) strategically placed around the building.
- All the base stations are wired together using copper or fibre.
- A simplifying assumption that all radio transmitters have some fixed range will be used to follow.

- Wireless clients associate to a wired AP (Access Point) Called infrastructure mode; there is also ad-hoc mode with no AP, but that is rare.

# 802.11 protocol stack

- In 802.11, the MAC (Medium Access Control) sublayer determines how the channel is allocated, that is, who gets to transmit next.
- Above it is the LLC (Logical Link Control) sublayer, whose job it is to hide the differences between the different 802 variants & make them indistinguishable as far as the network layer is concerned.
- The 1997 802.11 standard specifies three transmission techniques allowed in the physical layer.
- The **infrared method** uses the same technology as television remote controls do.
- The other two use short-range radio, using techniques called **FHSS (Frequency-hopping spread spectrum)** & **DSSS (Direct Sequence Spread Spectrum).** Both of these use a part of the spectrum that doesn't require licensing (the 2.4 GHz ISM band).
- All of these techniques operate at 1 or 2 Mbps & at low enough power that they don't conflict too much.
- In 1999, two new techniques were introduced to achieve higher bandwidth. These are called **OFDM (Orthogonal Frequency-Division Multiplexing)** & **HR-DSSS (High Rate / High Rate / Direct Sequence Spread Spectrum Physical Layer.).** They operate at up to 54 Mbps & 11 Mbps, respectively.
- In 2001, a second OFDM modulation was introduced, but in a different frequency band from the first one.




# 802.11 Physical layer

- All of the 802.11 techniques use short-range radios to transmit signals in either the 2.4-GHz or the 5-GHz ISM frequency bands. These bands have the advantage of being unlicensed & hence freely available to any transmitter willing to meet some restrictions, such as radiated power of at most 1 W (though 50 mW is more typical for wireless LAN radios).
- Unfortunately, this fact is also known to the manufacturers of garage door openers, cordless phones, microwave ovens, & countless other devices, all of which compete with laptops for the same spectrum. The 2.4-GHz band tends to be more crowded than the 5-GHz band, so 5 GHz can be better for some applications even though it has shorter range due to the higher frequency.
- All of the transmission methods also define multiple rates. The idea is that different rates can be used depending on the current conditions. If the wireless signal is weak, a low rate can be used. If the signal is clear, the highest rate can be used. This adjustment is called **rate adaptation**.
- Since the rates vary by a factor of 10 or more, good rate adaptation is important for good performance. Since it is not needed for interoperability, the standards don't say how rate adaptation should be done.
- The first transmission method is **802.11b**.
    - A spread-spectrum method that supports rates of 1, 2, 5.5, & 11 Mbps
    - The operating rate is always nearly 11 Mbps.

- Similar to CDMA (Code Division Multiple Access) system, except that there is only one spreading code that is shared by all users
- It has the property that its autocorrelation is low except when the sequences are aligned. This property allows a receiver to lock onto the start of a transmission.
- To send at a rate of 1 Mbps, the **Barker sequence** is used with BPSK (Binary Phase Shift Keying) modulation to send 1 bit per 11 chips. The chips are transmitted at a rate of 11 Mchips/sec.
- To send at 2 Mbps, it is used with QPSK (Quadrature Phase Shift Keying) modulation to send 2 bits per 11 chips.
- The higher rates are different. These rates use a technique called **CCK (Complementary Code Keying)** to construct codes instead of the Barker sequence. The 5.5-Mbps rate sends 4 bits in every 8-chip code, & the 11-Mbps rate sends 8 bits in every 8-chip code.

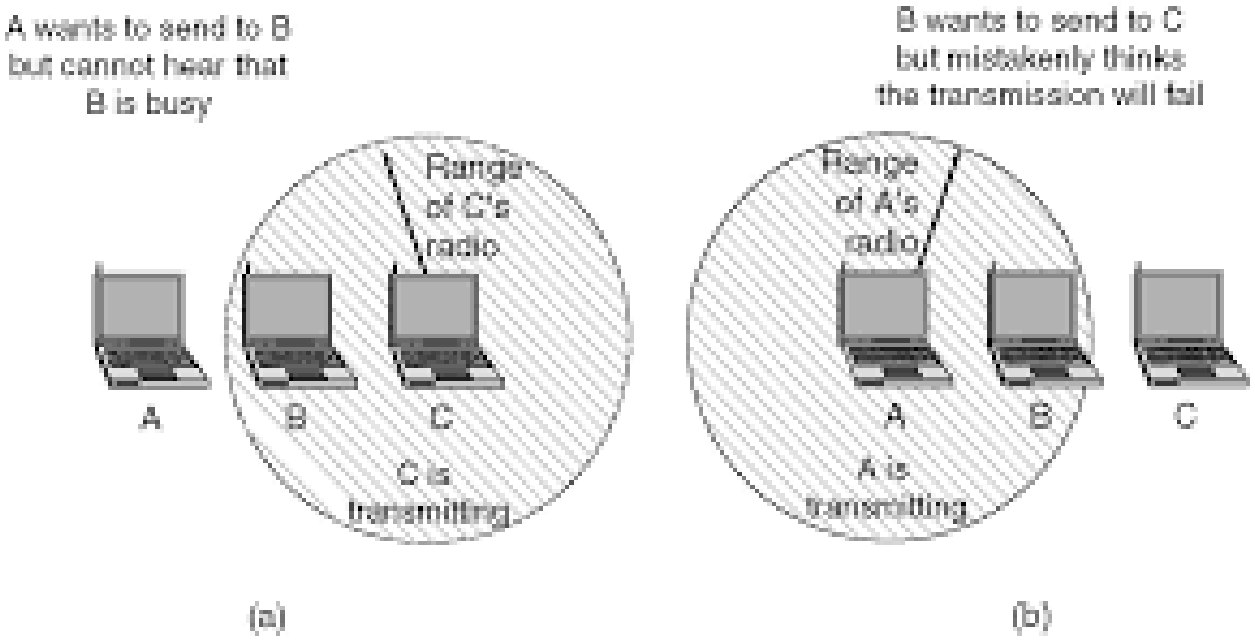
- Next, is **802.11a**
  - Supports rates up to 54 Mbps in the 5-GHz ISM band.
  - Although the 802.11a group was set up first, the 802.11b standard was approved first & its product got to market well ahead of the 802.11a products, partly because of difficulty of operating in higher 5-GHz band.
  - The 802.11a method is based on **OFDM (Orthogonal Frequency Division Multiplexing)** because OFDM uses the spectrum efficiently & resists wireless signal degradations such as multipath.
  - Bits are sent over 52 subcarriers in parallel, 48 carrying data & 4 used for synchronization. Each symbol lasts 4μs & sends 1, 2, 4, or 6 bits.
  - The bits are coded for error correction with a binary convolutional code first, so only 1/2, 2/3, or 3/4 of the bits are not redundant.
  - With different combinations, 802.11a can run at eight different rates, ranging from 6 to 54 Mbps. These rates are significantly faster than 802.11b rates, & there is less interference in the 5-GHz band.
  - But, 802.11b has a range that is about seven times greater than that of 802.11a, which is more important in many situations.
- **802.11g**
  - In May 2002, FCC (Federal Communications Commission) dropped its long-standing rule requiring all wireless communications equipment operating in the ISM bands in the U.S. to use spread spectrum, so it got to work on 802.11g, which was approved by IEEE in 2003.
  - It copies OFDM modulation methods of 802.11a but operates in narrow 2.4-GHz ISM band along with 802.11b.

    - It offers same rates as 802.11a (6 to 54 Mbps) plus compatibility with any 802.11b devices that happen to be nearby.
    - All of these different choices can be confusing for customers, so it is common for products to support 802.11a/b/g in a single NIC (Network Interface Controller).
- **802.11n**
    - A high-throughput physical layer
    - It was approved in 2009
    - The goal for 802.11n was throughput of at least 100 Mbps after all the wireless overheads were removed.
    - This goal called for a raw speed increase of at least a factor of four. To make it happen, the committee doubled the channels from 20 MHz to 40 MHz & reduced framing overheads by allowing a group of frames to be sent together.
    - But, 802.11n uses up to four antennas to transmit up to four streams of information at the same time.
    - The signals of streams interfere at receiver, but they can be separated using **MIMO (Multiple Input Multiple Output)** communications techniques.
    - The use of multiple antennas gives a large speed boost, or better range & reliability instead.
    - MIMO, like OFDM, is one of those clever communications ideas that is changing wireless designs & which we are all likely to hear a lot about in the future.

# MAC Sublayer protocol

- The IEEE 802.11 MAC protocol is the standard for wireless LANs
- It is widely used in testbeds & simulations for wireless multihop ad hoc networks.
- But this protocol was not designed for multihop networks. Though it can support some ad hoc network architecture, it is not intended to support the wireless mobile ad hoc network, in which multihop connectivity is one of the most prominent features.
- Two problems have been explained before: the **hidden station problem & the exposed station problem**.
- 802.11 supports two modes of operation:
- The first, called **DCF (Distributed Coordination Function)**, doesn't use any kind of central control (in that respect, similar to Ethernet).
    - It is a mandatory function used in CSMA/CA.

    - It is used in distributed contention-based channel access.
    - It is deployed in both Infrastructure BSS (basic service set) as well as Independent BSS
- The other, called **PCF (Point Coordination Function)**, uses the base station to control all activity in its cell.
    - It is an optional function used by 802.11 MAC Sublayer.
    - It is used in centralized contention-free channel access.
    - It is deployed in Infrastructure BSS only.
- All implementations must support DCF but PCF is optional.



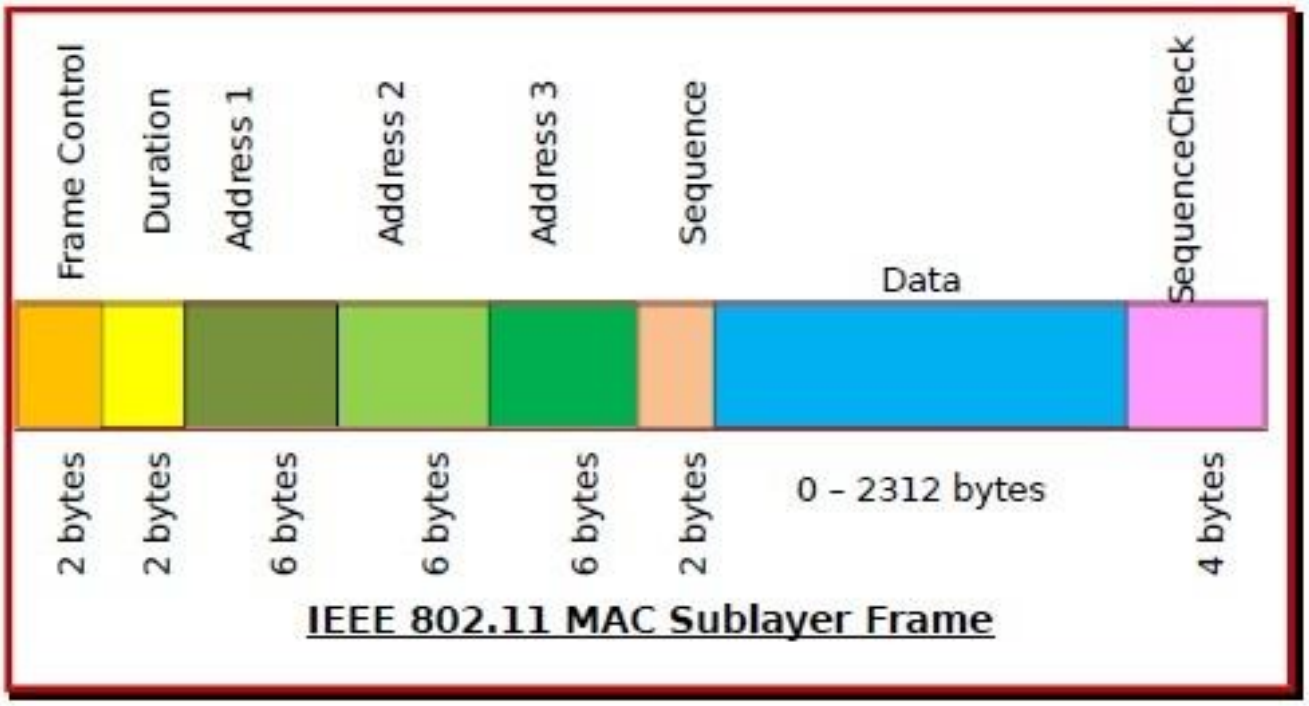

In wireless systems, the method of collision detection doesn't work. It uses a protocol called **carrier sense multiple access with collision avoidance (CSMA/CA)**.

The method of CSMA/CA is −

- When a frame is ready, the transmitting station checks whether the channel is idle or busy.
- If the channel is busy, the station waits until the channel becomes idle.
- If the channel is idle, the station waits for an Inter-frame gap (IFG) amount of time and then sends the frame.
- After sending the frame, it sets a timer.
- The station then waits for acknowledgement from the receiver. If it receives the acknowledgement before expiry of timer, it marks a successful transmission.
- Otherwise, it waits for a back-off time period and restarts the algorithm.

# Frame structure

- The main fields of a frame of wireless LANs as laid down by IEEE 802.11 are:
  - Frame Control − It is 2 bytes starting field composed of 11 subfields. It contains control information of the frame.
  - Duration − It is a 2-byte field that specifies the time period for which the frame and its acknowledgement occupy the channel.
  - Address fields − There are three 6-byte address fields containing addresses of source, immediate destination and final endpoint respectively.
  - Sequence − It a 2 bytes field that stores the frame numbers.
  - Data − This is a variable sized field carries the data from the upper layers. The maximum size of data field is 2312 bytes.
  - Check Sequence − It is a 4-byte field containing error detection information.




| Version=00 (2 bits) | Type=10 (2 bits) | Subtype=0000 (4 bits) | To DS (1 bit) | From DS (1 bit) | More Fragments (1 bit) | Retry (1 bit) | Power Management (1 bit) | More Data (1 bit) | Protected (1 bit) | Order (1 bit) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |

In the **Frame control** field:
The first is the **Protocol version**, set to 00.

It is there to allow future versions of 802.11 to operate at the same time in the same cell.

Then come the **Type** (data, control, or management) & **Subtype** fields (e.g., RTS or CTS).

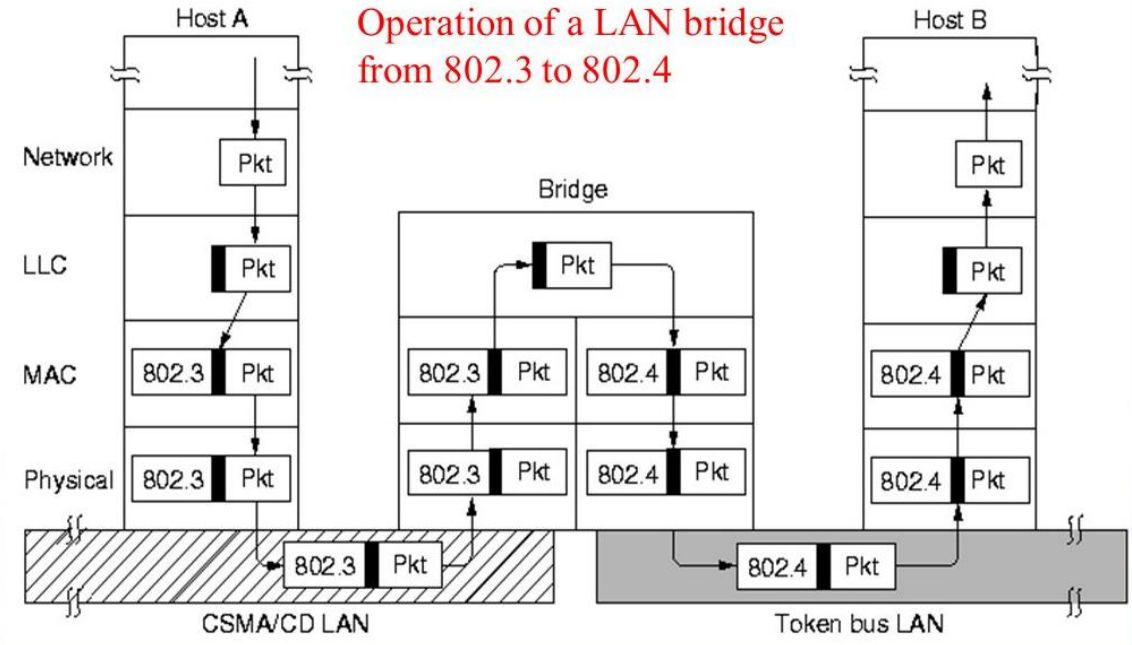
For a regular data frame (without quality of service), they are set to 10 & 0000 in binary.

The **To DS** & **From DS** bits are set to indicate whether the frame is going to or coming from the network connected to the APs, which is called the **distribution system.**
The **More fragments** bit means that more fragments will follow.
The **Retry** bit marks a retransmission of a frame sent earlier.
The **Power management** bit indicates that the sender is going into power-save mode.
The **More data** bit indicates that the sender has additional frames for the receiver.
The **Protected** Frame bit indicates that the frame body has been encrypted for security.
The **Order** bit tells the receiver that the higher layer expects the sequence of frames to arrive strictly in order.

# Bridges

- A network bridge is a computer networking device that creates a single aggregate network from multiple communication networks or network segment. This function is called **network bridging**.
- Something is required to join these Networks so that they can become part of the whole network.
- In a divided LAN, if there is no medium to join the separated LAN parts, an enterprise may be Limited in its growth potential. The bridge is one of the tools to join these LANs.
- A LAN (for example, Ethernet), can be limited in its transmission distance. We can eliminate this problem using bridges as repeaters, so that we can connect a geographically extensive network within the building or campus using bridges. Hence, geographically challenged networks can be created using Bridges.
- A network administrator can control the amount of traffic going through bridges sent across the expensive network media.
- A bridge is a plug & play device. Others need not configure the bridge.

- If a machine is taken out from a network, then there is no need for the Network administrator to update the bridge configuration information as bridges are self-configured.
- It also provides easy transfer of data.
- Host A has a packet to send. The packet descends into the LLC sublayer & acquires an LLC header. Then it passes into the MAC sublayer & an 802.3 header is prepended to it. This unit goes out onto the cable & eventually is passed up to the MAC sublayers in the bridge, where the 802.3 header is stripped off. The bare packet is then headed off to the LLC sublayer in the bridge.
- In this example, the packet is destined for an 802.4 subnet connected to the bridge, so it works its way down the 802.4 side of the bridge & off it goes.




## Types of Bridges

- **Transparent basic bridge**
- **Source routing bridge**
- **Transparent learning bridge**
- **Transparent spanning bridge**

**The Transparent Bridge**

- The transparent bridge finds location of user using source & destination address.
- When frame is received at the bridge, it checks its source address & Destination address.

- The destination address is stored if it was not found in a routing table.
- Then the frame is sent to all LAN excluding the LAN from which it came.
- The source Address is also stored in the routing table.
- If another frame arrives having the destination address as previous source address, then it is forwarded to that port.

**The Transparent Spanning Tree Bridge**

- These bridges use a subnet of full topology to create a loop free operation.
- The received frame is checked by the bridge in following manner:
  - The destination address of arrived frame is checked with routing table in the database. Here more information is required for bridges. Other bridge port is also stored in the database. This information is known as **Port State Information** & it helps in deciding whether, a port can be used for this destination address or not.
  - The port can be in a block state to fulfil the requirements of spanning tree operations or in a forwarding state.
  - If the port is in forwarding state, the frame is routed across the port.
  - The port can have different status such as; it may be in disabled state for the maintenance reason or may also be unavailable temporarily if data bases are being changed in the bridge because of result of the change in the routed network.

**Source Routing Bridge**

- A communication protocol in which the sending station is aware of all the bridges in the network & predetermines the complete route to the destination station before transmitting.
- The Routing Information Field (RIF)in the LAN frame header, contains the information of route followed by the LAN network.
- The intermediate nodes that are required to receive & send the frame must be identified by the routing information.
- For this reason, source routing requires that the user traffic should follow the path determined by the routing information field.

- The frames of the source routing protocol are different from other bridge frames because the source routing information must be contained within the frame.
- The architecture of the other bridges & source routing bridges is similar.

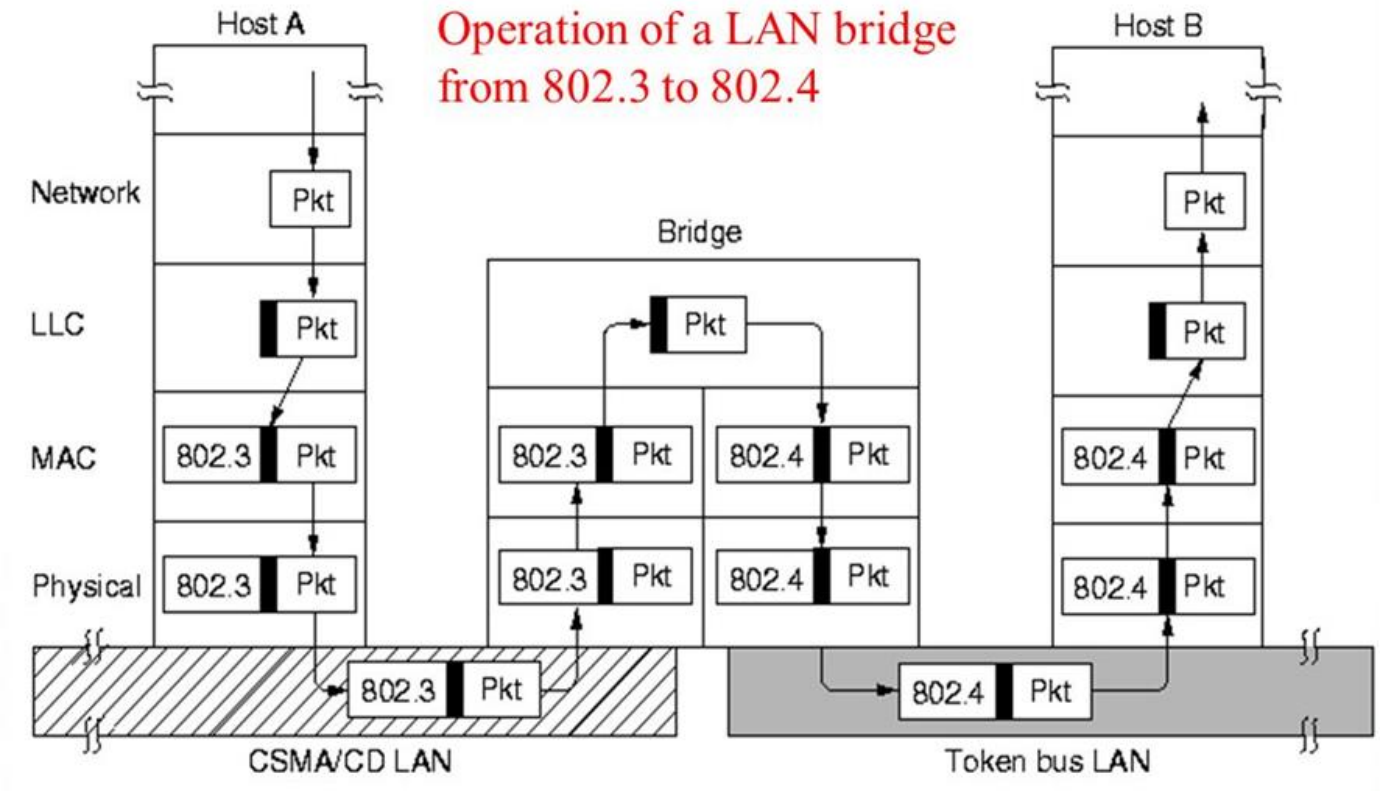
# Switches

- A network switch
- Also called switching hub or bridging hub
- A computer networking device that connects devices together on a computer network by using packet switching to receive, process & forward data to destination device.

**Typical switch management features**

- A couple of managed D-Link Gigabit Ethernet switches, connected to the Ethernet ports on a few patch panels using Category 6 patch cables(all equipment is Installed in a standard19-inch rack)
    1. Turn particular port range on or off
    2. Link band width & duplex settings
    3. Priority settings for ports
    4. IP management by IP clustering
    5. MAC filtering & other types of "port security" features which prevent MAC flooding
    6. Use of Spanning Tree Protocol (STP) & Shortest Path Bridging (SPB) technologies
    7. Simple Network Management Protocol (SNMP) monitoring of device & link health
- **Layer-specific functionality**
    - Modern commercial switches use primarily Ethernet interfaces.
    - The core function of an Ethernet switch is to provide a multiport layer 2 bridging function.
    - Many switches also perform operations at other layers.
    - A device capable of more than bridging is known as a multilayer switch.
    - Switches may learn about topologies at many layers & forward at one or more layers.

# Bridges from 802.x to 802.y




# Repeaters

- A repeater operates at the physical layer.
- Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network.
- An important point to be noted about repeaters is that they don't amplify the signal.
- When the signal becomes weak, they copy the signal bit by bit & regenerate it at the original strength.
- It is a 2-port device.

# Hubs

- A hub is basically a multiport repeater.
- A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations.
- Hubs can't filter data, so data packets are sent to all connected devices.

- In other words, collision domain of all hosts connected through Hub remains one. Also, they don't have intelligence to find out best path for data packets which leads to inefficiencies & wastage.

# Bridges

A bridge operates at data link layer. A bridge is a repeater, with add on functionality of filtering content by reading the MAC addresses of source & destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input & single output port, thus making it a 2-port device.

# Switches

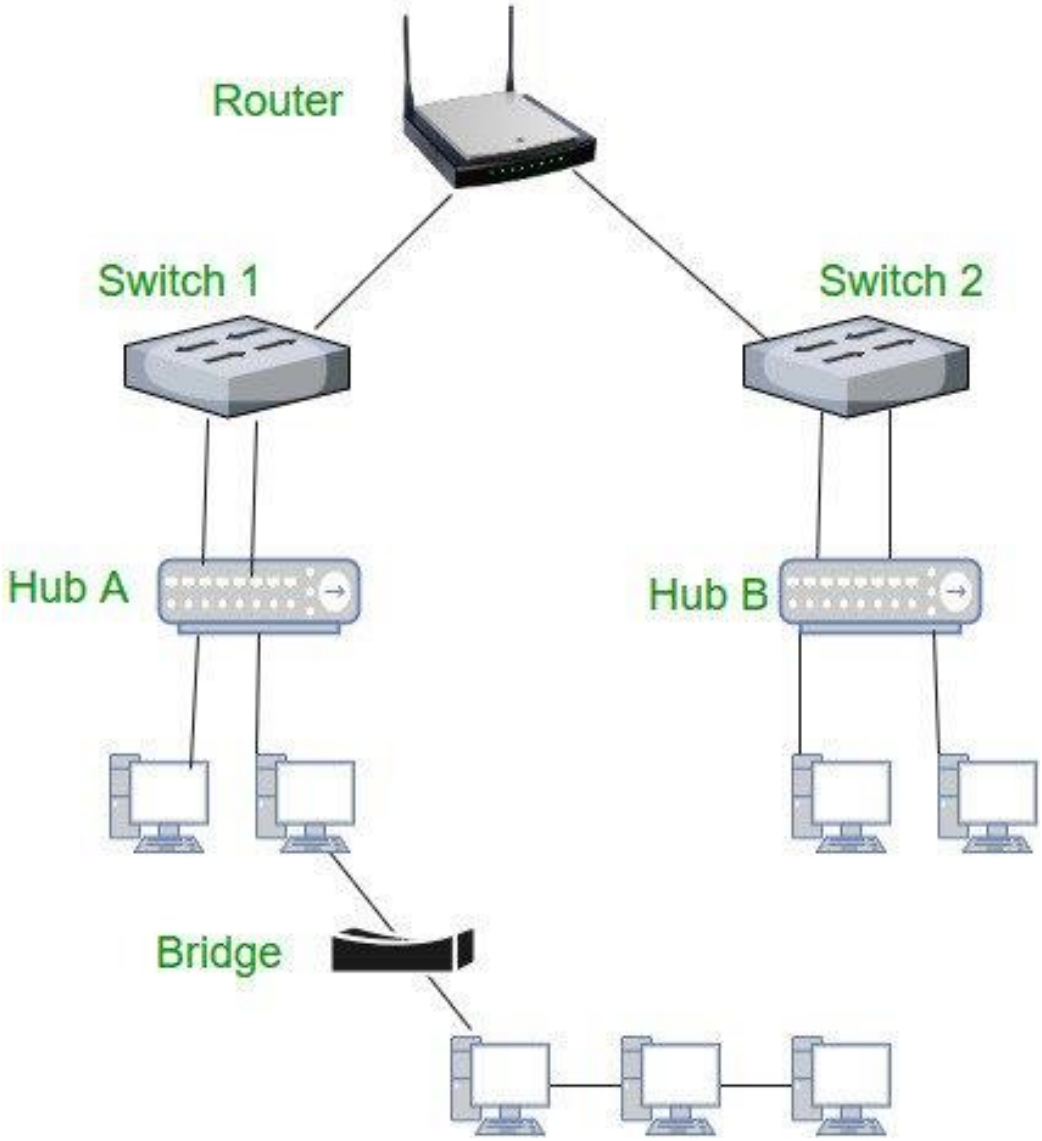
A switch is a multi-port bridge with a buffer & a design that can boost its efficiency (large number of ports imply less traffic) & performance. Switch is a data link layer device. Switch can perform error checking before forwarding data, that makes it very efficient as it doesn't forward packets that have errors & forward good packets selectively to correct port only. In other words, switch divides collision domain of hosts, but broadcast domain remains same.

# Routers

A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs & WANs together & have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.

# Gateways

A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, & transfer it to another system. Gateways are also called **protocol converters** & can operate at any network layer. Gateways are generally more complex than switch or router.




| | |
|---|---|
| Application layer | Application Gateway |
| Transport layer | Transport Gateway |
| Network Layer | Router |
| Datalink Layer | Bridge, Switch |
| Physical Layer | Repeater, Hub |

# Module - 3 (Network Layer)

Network layer design issues. Routing algorithms - The Optimality Principle, Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Multicast routing, Routing for mobile hosts. Congestion control algorithms. Quality of Service (QoS) - requirements, Techniques for achieving good QoS.

## Network Layer

- Controls the operation of the subnet
- Main aim of this layer is to deliver packets from source to destination across multiple links (networks)
- If two computers (system) are connected on the same link, then there is no need for a network layer
- Network layer routes the signal through different channels to the other end and acts as a network controller
- It also divides the outgoing messages into packets and assemble incoming packets into messages for higher levels

## Functions Of Network Layer

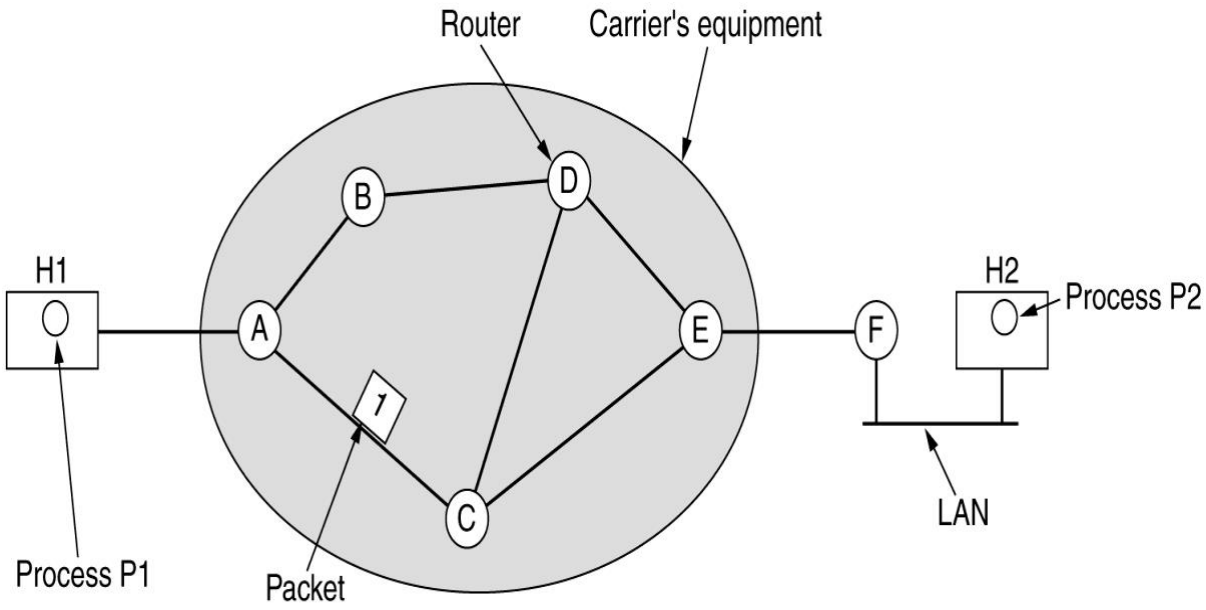
1. Translates logical network address into physical address, concerned with circuit, message or packet switching
2. Routers and gateways operate in the network layer: A mechanism is provided by Network Layer for routing the packets to final destination.
3. Connection services are provided including network layer flow control, network layer error control and packet sequence control
4. Breaks larger packets into small packets

## Network layer design issues

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

**Store and Forward packet switching**

The host sends the packet to the nearest router. This packet is stored there until it has fully arrived. Once the link is fully processed by verifying the checksum, then it is forwarded to the next router till it reaches the destination. This mechanism is called "**Store and Forward packet switching**."




**Services Provided to the Transport Layer**

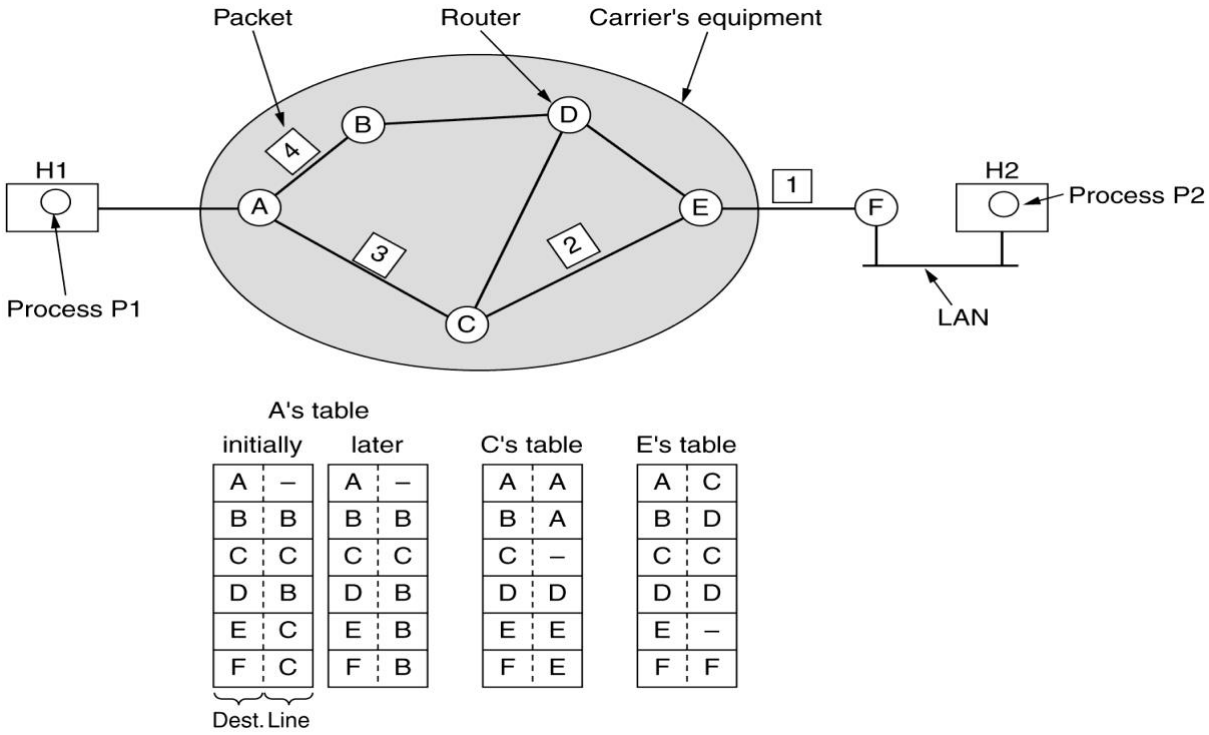
- Network layer provides services to transport layer at the network layer/transport layer interface
- These services need to be carefully designed with the following **goals** in mind:
  1. The services should be independent of the router technology
  2. The transport layer should be shielded from the number, type, & topology of the routers present
  3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs & WANs

- With these goals, the designers of network layer have a lot of freedom in deciding the services to be offered to the transport layer
- This freedom raised a battle between two warring factions, that is, **whether the network layer should provide connection-oriented service or connectionless service**
- Those who represented the **Internet community** argues that the routers' job is moving packets around & nothing else
  - Based on 40 years of experience with real computer network, they claim that the network is inherently unreliable, no matter how it is designed
  - So, the hosts should accept this fact & do error control (i.e., error detection & correction) & flow control themselves
  - This viewpoint leads to the conclusion that the **network service should be connectionless,** with primitives **SEND PACKET** & **RECEIVE PACKET** & little else
  - Also, no packet ordering & flow control should be done, because hosts are going to do that

    - This reasoning is an example of the **end-to-end argument**, a design principle that has been very influential in shaping the Internet
    - Also, each packet must carry the full destination address, because each packet sent is carried independently of its predecessors, if any.
    - e.g.: Internet
    - The packets are frequently called **datagrams** (in analogy with telegrams)
- Those who represented the **telephone companies** argue that the **network should provide a reliable, connection-oriented service**.
    - They claim that 100 years of successful experience with the worldwide telephone system taught that, **quality of service** is the dominant factor, & **without connections in the network, quality of service is very difficult** to achieve, especially for real-time traffic such as voice & video.
    - e.g.: ATM (Asynchronous Transfer Mode)
    - The connection is called a **VC (virtual circuit),** in analogy with the physical circuits set up by the telephone system
- This controversy is still alive
- The data networks, widely used in olden times, such **as X.25 in 1970s** and its successor **Frame Relay in 1980s**, were **connection-oriented**
- But since the days of ARPANET & early Internet, **connectionless network** layers **have grown tremendously in popularity**
- The **IP protocol** is now an ever-present symbol of success
- But Internet is evolving **connection-oriented features as quality of service becomes more important**
- Two examples of connection-oriented technologies are MPLS (Multiprotocol Label Switching), & VLANs. Both technologies are widely used.

**Implementation Of Connectionless Service**

- Let's assume that **a message** is **4 times longer than the maximum packet size**. So, the network layer has to **break it into 4 packets**: 1, 2, 3, & 4; & send each of them in turn to router A using PPP (Point-to-Point Protocol).
- At this point the carrier takes over
- Every router has an internal table telling it where to send packets for each possible destination
- Each table entry is a pair consisting of a destination & the outgoing line to use for that destination
- Only directly-connected lines can be used
- For example, in the below figure (routing within a datagram network), A has only two outgoing lines—to B & C. So, every incoming packet must be sent to one of these routers, even if the ultimate destination is some other router.




| A | – |
|---|---|
| B | B |
| C | C |
| D | B |
| E | C |
| F | C |

| A | – |
|---|---|
| B | B |
| C | C |
| D | B |
| E | B |
| F | B |

| A | A |
|---|---|
| B | A |
| C | – |
| D | D |
| E | E |
| F | E |

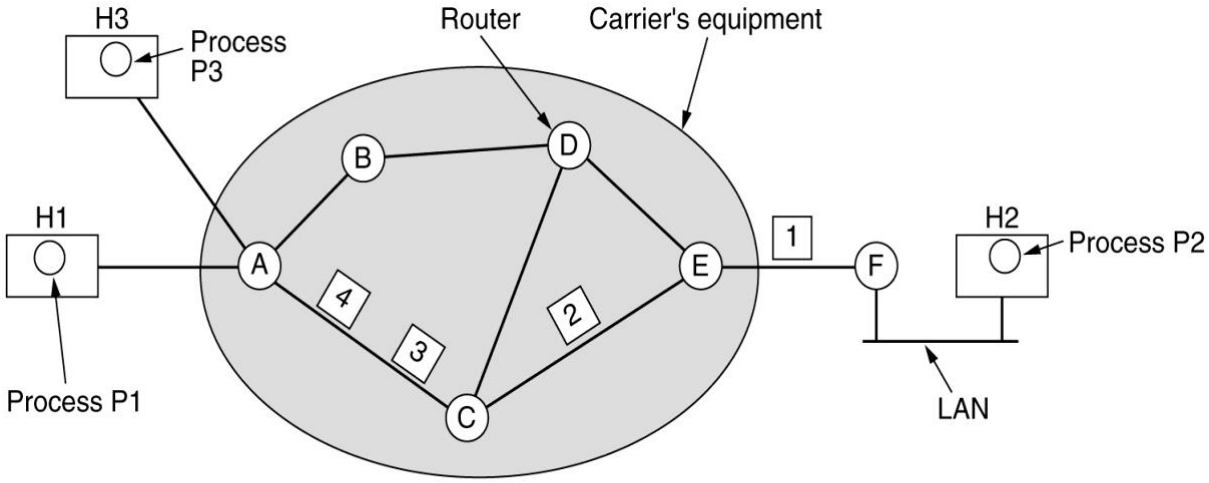
| A | C |
|---|---|
| B | D |
| C | C |
| D | D |
| E | – |
| F | F |

- A's initial routing table is shown in the figure under the label '**initially**'.
- As they arrived at A, packets 1, 2, & 3 are stored there, to verify their checksums
- Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame. Packet 1 is then forwarded to E & then to F. When it gets to F, it is sent within a frame to H2 over the LAN. Packets 2 & 3 follow the same route.
- But, when packet 4 reaches A it is sent to router B, even though it's also destined for F. **For some reason, A decided to send packet 4 via a different route than that of the first three packets**.
- Maybe it has learned of a traffic jam somewhere along the path ACE & updated its routing table, as shown under the label "later"
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**. Routing algorithms are one of the main topics we will study in this Module. There are several different kinds of them.

**IP (Internet Protocol)**, which is the basis for the entire Internet, is the dominant **example of a connectionless network** service. Each packet carries a destination IP address that routers use to individually forward each packet. The addresses are 32 bits in IPv4 packets & 128 bits in IPv6 packets.

## Implementation of Connection-Oriented Service

- For connection-oriented service, we need a virtual-circuit network
- It is to avoid the confusion of choosing a new route for every packet sent, as in the above situation
- Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup & stored in tables inside the routers
- That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works
- When the connection is released, the virtual circuit is also terminated
- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to
- As an example, consider the situation shown in the figure below (routing within a virtual-circuit network)




| A's table | | | | C's table | | | | E's table | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| H1 | 1 | C | 1 | A | 1 | E | 1 | C | 1 | F | 1 |
| H3 | 1 | C | 2 | A | 2 | E | 2 | C | 2 | F | 2 |
| In | | Out | | | | | | | | | |

- Here, host H1 has established connection 1 with host H2. This connection is remembered as the first entry in each of the routing tables. The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1.
- H3 also want to establish a connection to H2
- First, it chooses connection identifier 1 & tells the network to establish the virtual circuit. This leads to second row in the tables.

- But the conflict is that A can identify the packets from H1 & those from H3, but C can't
- So, A assigns a different connection identifier to the outgoing traffic for the second connection
- So, to avoid these conflicts, routers need to replace the connection identifiers for outgoing packets. This process is called label switching.
- An example of a connection-oriented network service is **MPLS** (**Multiprotocol Label Switching)**
    - Used within ISP (Internet Service Provider) networks in the Internet
    - IP packets are wrapped in an MPLS header having a 20-bit connection identifier or label
    - MPLS is often hidden from customers, with the ISP establishing long-term connections for large amounts of traffic, but it's increasingly being used to help when quality of service is important but also with other ISP traffic management tasks

## Comparison of Virtual-Circuit & Datagram Subnets

| Issue | Datagram subnet | Virtual-circuit subnet |
|---|---|---|
| Circuit setup | Not needed | Required |
| Addressing | Each packet contains the full source and destination address | Each packet contains a short VC number |
| State information | Routers do not hold state information about connections | Each VC requires router table space per connection |
| Routing | Each packet is routed independently | Route chosen when VC is set up; all packets follow it |
| Effect of router failures | None, except for packets lost during the crash | All VCs that passed through the failed router are terminated |
| Quality of service | Difficult | Easy if enough resources can be allocated in advance for each VC |
| Congestion control | Difficult | Easy if enough resources can be allocated in advance for each VC |

- Using **virtual circuits requires a setup phase**, which takes time and consumes resources. But, once this price is paid, **figuring out what to do with a data packet in a virtual-circuit network is easy**: the router just uses the circuit number to index into a table to find out where the packet goes. In a

**datagram network, no setup is needed but a more complicated** lookup procedure is required to locate the entry for the destination.

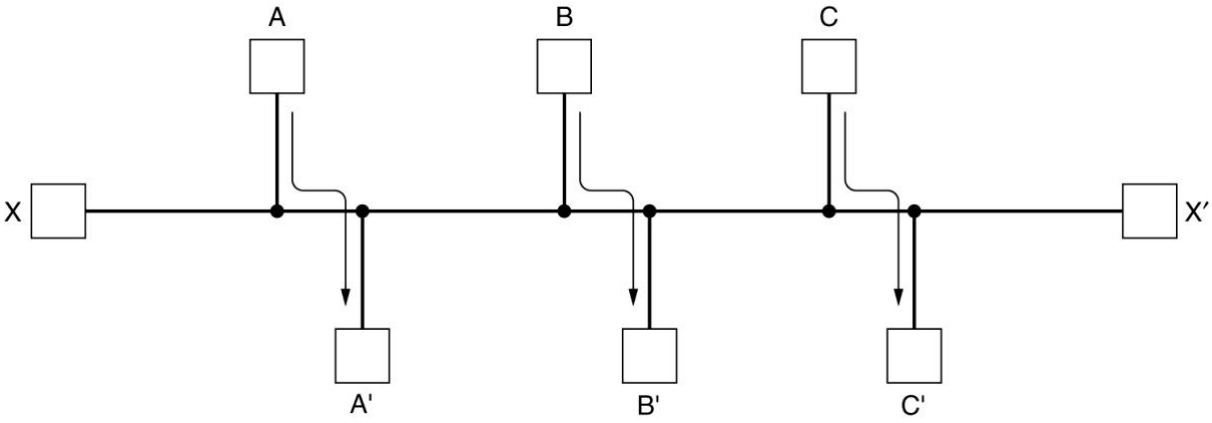
- **Destination addresses used in datagram networks are longer than circuit numbers used in virtual-circuit networks**. Circuit numbers have a global meaning. Including a full destination address in every packet (that is fairly short), cause a significant amount of overhead, & hence waste of bandwidth.
- **Datagram networks need an entry for every possible destination, whereas virtual-circuit networks just need an entry for each virtual circuit**
- **Virtual circuits guarantee quality of service & avoid congestion** within the network because resources (e.g., buffers, bandwidth, & CPU cycles) can be reserved in advance, when the connection is established. **Congestion avoidance is more difficult with datagram network**.

## ROUTING ALGORITHMS

- Main function of network layer is routing packets from source machine to destination machine
- Packets require multiple hops to reach destination, except for broadcast networks, but it is an issue if the source & destination are not on the same network
- The **routing algorithm** is the **part of network layer software responsible for deciding which output line an incoming packet should be transmitted on**
- If a network uses datagrams internally, the routing decision must be made again & again for every arriving data packet, because, the best route may have changed since last time
- If the network uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the already established route. The latter case is sometimes called **session routing** because a route remains in force for an entire session (e.g., while logged in over a VPN or a file transfer).
- Let's distinguish routing & forwarding:
- Routing makes the decision of which routes to use, & forwarding, happens when a packet arrives
- A router as has two processes: One, to handle each arriving packet, assigning the outgoing line to use for it in the routing tables. This process is **forwarding**; The other is to fill in & update the routing tables. That is where the routing algorithm comes into play.
- Properties desirable in a routing algorithm:
    - Correctness
    - Simplicity
    - Robustness

- Stability
- Fairness
- Efficiency

- **Robustness**
  - A big network is expected to run continuously for years without system-wide failures
  - There will be hardware & software failures of all kinds
  - Hosts, routers, & lines will fail repeatedly, & the topology will change many times
  - The routing algorithm should be able to cope with changes in the topology & traffic without affecting its job.
- **Stability**
  - A stable algorithm reaches equilibrium & stays there
- **Fairness & efficiency** are contradictory goals. As a simple example of this conflict, look at the below figure (network with a conflict between fairness & efficiency):



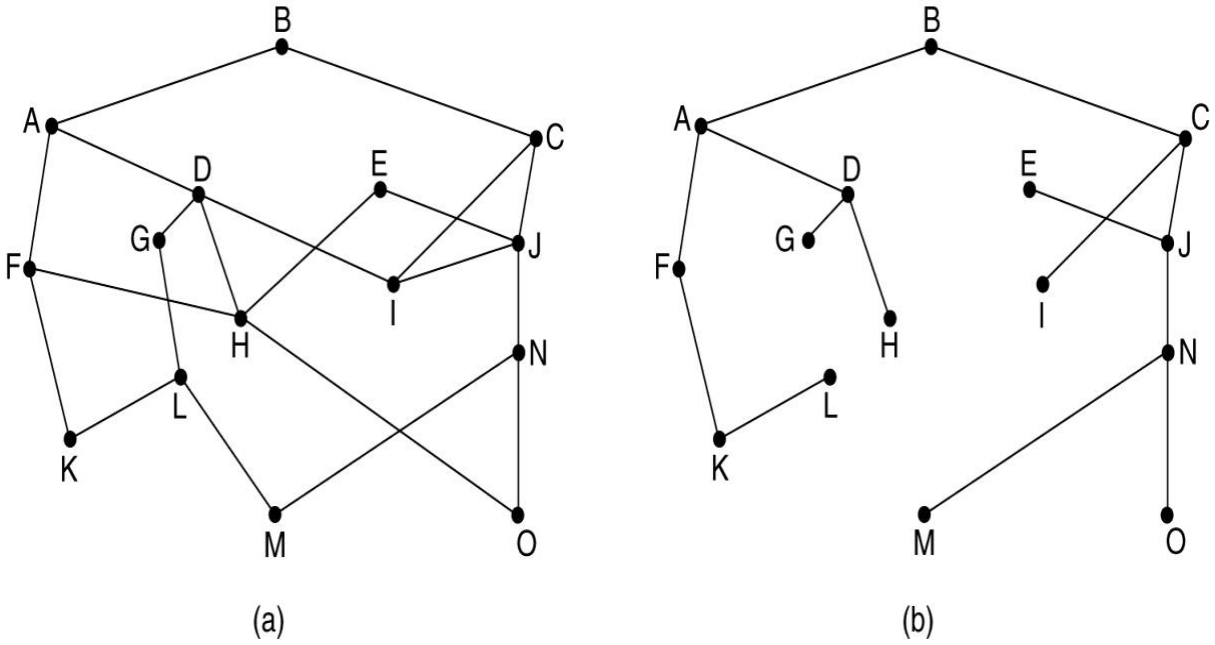

- Suppose that there is enough traffic between A & A′, between B & B′, & between C & C′
- To maximize the total flow, the X to X′ traffic should be shut off altogether.
- As a compromise, networks attempt to minimize the distance a packet must travel, or simply reduce the number of hops a packet must make.
- **Routing algorithms** can be grouped into two major classes: nonadaptive & adaptive.
- **Nonadaptive algorithms**
  - Do not change their routing decisions on any measurements or estimates of the current topology & traffic
  - Instead, the route is computed in advance, & downloaded to the routers when the network is booted
  - This procedure is sometimes called **static routing**
  - Doesn’t respond to failures

- So, static routing is mostly useful for situations in which the routing choice is clear
- The two **types of non – adaptive routing algorith**ms are:
  I. Flooding − In flooding, when a data packet arrives at a router, it is sent to all the outgoing links except the one it has arrived on. Flooding may be uncontrolled, controlled or selective flooding.
  II. Random walks − This is a probabilistic algorithm where a data packet is sent by the router to any one of its neighbours randomly.

- **Adaptive algorithms**
  - They change their routing decisions with respect to the changes in the topology, & sometimes changes in the traffic as well
  - The three popular **types of adaptive routing algorithms** are:
    I. Centralized algorithm − It finds the least-cost path between source and destination nodes by using global knowledge about the network. So, it is also known as global routing algorithm.
    II. Isolated algorithm − This algorithm procures the routing information by using local information instead of gathering information from other nodes.
    III. Distributed algorithm − This is a decentralized algorithm that computes the least-cost path between source and destination iteratively in a distributed manner
  - These **dynamic routing algorithm**s differ in
    - where they get their information (e.g., locally, from adjacent routers, or from all routers);
    - when they change the routes (e.g., when the topology changes, or every ΔT seconds as the load changes); &
    - what metric is used for optimization (e.g., distance, number of hops, or estimated transit time)

## The Optimality principle

- It is regarded as a general statement (anyone can make) about optimal routes without regard to network topology or traffic. This statement is known as **the optimality principle** (Bellman, 1957).
- It states that if a router A is on the optimal path from router L to router B, then the optimal path from A to B also falls along the same route. To see this, call the part of the route from L to A "***r1***" & the rest of the route "***r2"***. If a route better than "***r2"*** existed from A to B, it could be concatenated with "***r1***" to improve the route from L to B, contradicting our statement that "***r1r2"*** is optimal.

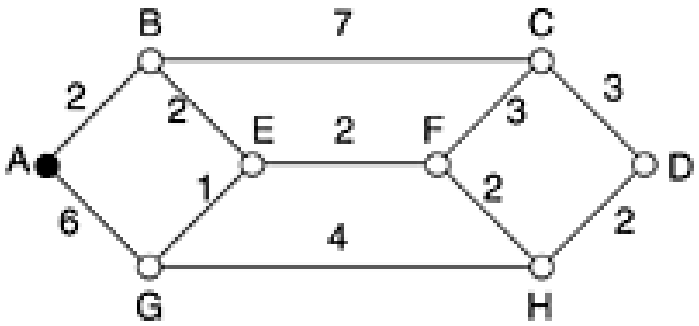



- As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree** & is illustrated in the above figure (a network & a sink tree for router B), where the distance metric is the number of hops.
- The goal of all routing algorithms is to discover & use the sink trees for all routers.
- A sink tree is not unique
- Other trees with the same path lengths may exist. If we allow all of the possible paths to be chosen, the tree becomes a more general structure called a DAG (**Directed Acyclic Graph**). DAGs have no loops.
- Since a sink tree is indeed a tree, it doesn't contain any loops, so each packet will be delivered within a finite & bounded number of hops.
- Links & routers can go down & come back up during operation, so different routers may have different ideas about the current topology.

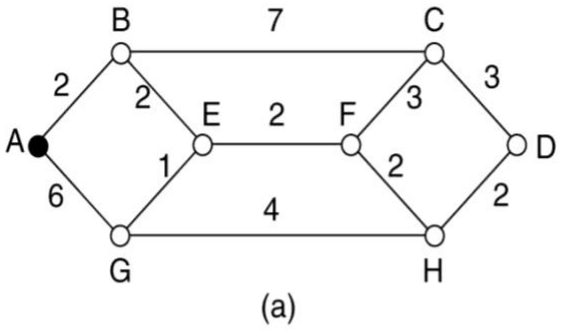
## Shortest Path Routing

- A simple technique to compute optimal paths from a network
- The idea is to **build a graph of the network**, with **each node** of the graph **representing a router** & **each edge** of the graph **representing a communication line**, or link
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph
- One way of measuring path length is the **number of hops**. Using this metric, the paths ABC & ABE in the below figure are equally long.

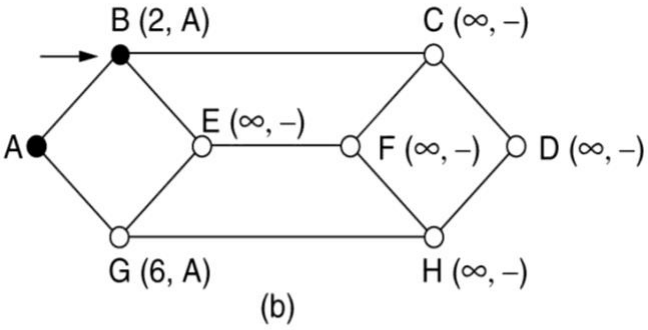
- Another metric is the geographic distance in kilometres, in which case **ABC (2+7=9) is clearly much longer than ABE (2+2=4)**.



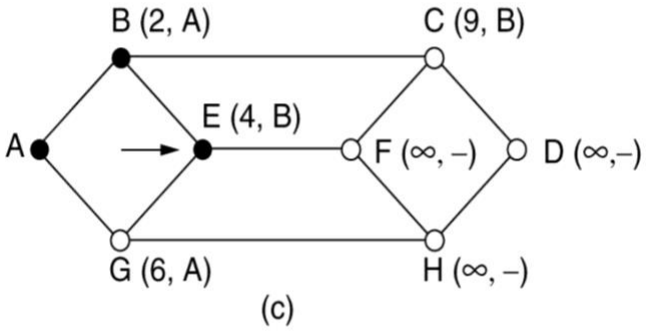

- Many other metrics can also be used other than **hops** & **physical distance**, like, **mean delay** of a standard test packet. So, in such case, the shortest path is the fastest path rather than the path with the fewest edges or kilometres.
- In general case, labels on the edges could be computed as the function of **distance**, **bandwidth**, **average traffic**, **communication cost**, **measured delay**, & other factors
- Several algorithms are known to compute the shortest path between two nodes of a graph
- Dijkstra, in 1959, finds the shortest paths between a source & all destinations in the network
- Each node is labelled (in parentheses) with its distance from the source node along the best-known path
- The distances must be non-negative
- Initially, no paths are known, so all nodes are labelled with infinity
- As the algorithm proceeds & paths are found, the labels may change, reflecting better paths
- A label may be either tentative or permanent. Initially, all labels are tentative (can be changed).
- When it is found that a label represents the shortest possible path from the source to that node, it is made permanent & never changed thereafter
- To illustrate how the labelling algorithm works, look at the weighted, undirected graph of the below figure (the 1$^{st}$ 6 steps used in computing the shortest path from A to D; The arrows indicate the working node), where the weights represent distance
- We want to find the **shortest path from A to D**
- Mark node A as permanent, by a filled-in circle
- Then we examine, each of the nodes adjacent to A (the working node), relabelling each one with the distance to A
- Whenever a node is relabelled, we also label it with the node from which the probe was made so that we can reconstruct the final path later



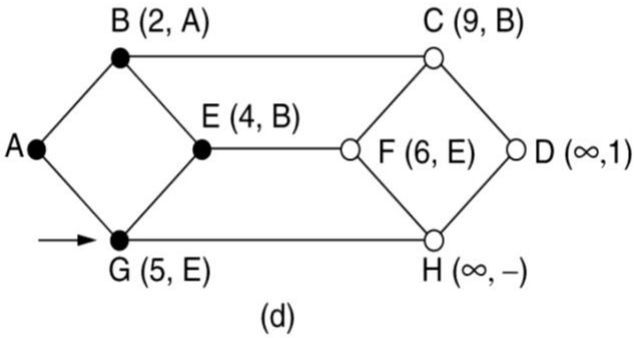
(a)



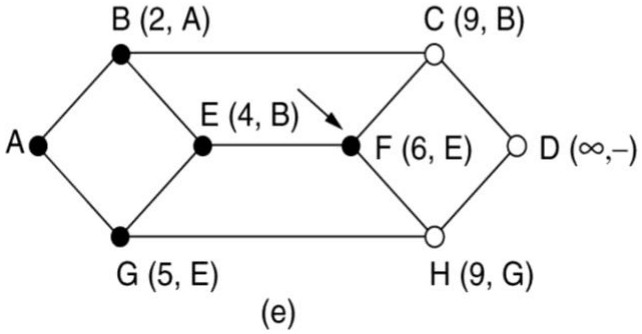
(b)



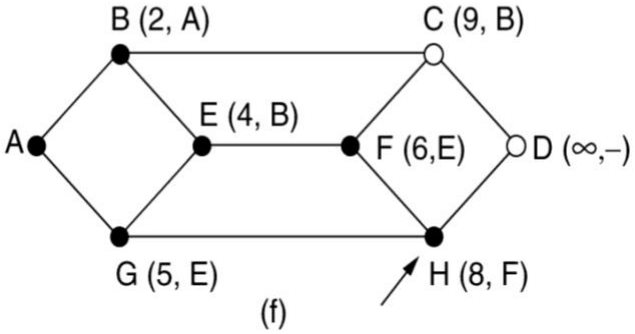
(c)



(d)



(e)



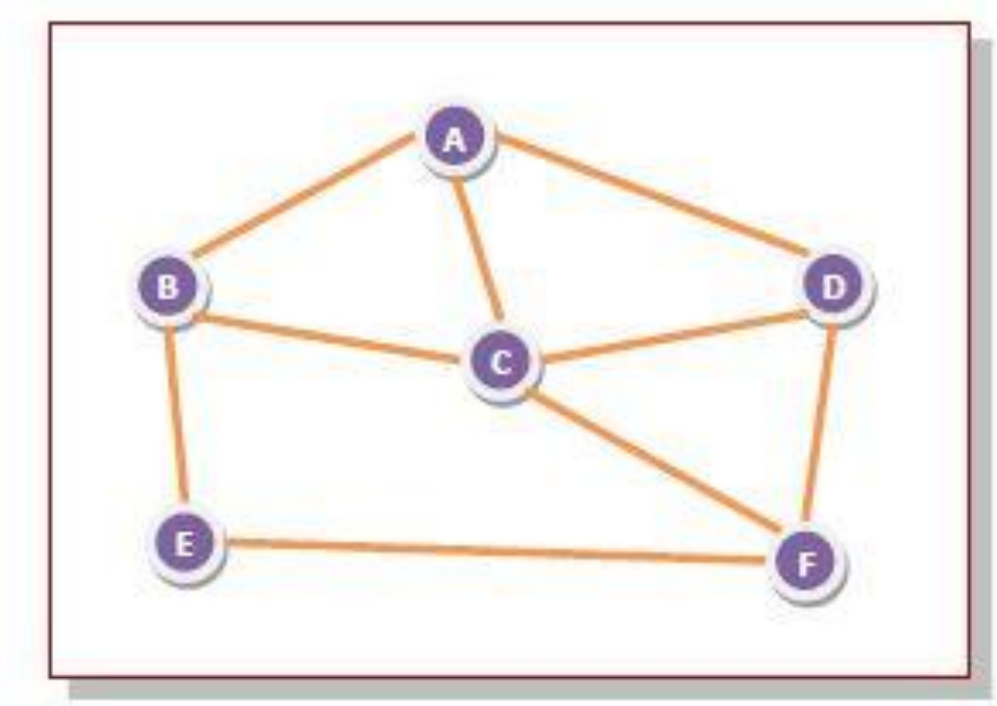
(f)

- If the network had more than one shortest path from A to D & we wanted to find all of them, we would need to remember all of the probe nodes that could reach a node with the same distance
- After examining each of the nodes adjacent to A, we examine all the tentatively labelled nodes in the whole graph & make the one with the smallest label permanent, as shown in figure(b)
- This one becomes the new working node. We now start at B & examine all nodes adjacent to it. If the sum of the label on B & the distance from B to the node being considered is less than the label on that node, we have a shorter path, so the node is relabelled.
- This Process is continued for all the nodes available in the graph.

## Flooding

- A non-adaptive routing technique
- In **flooding algorithm**, every incoming packet is sent out on every outgoing line except the one it arrived on
- It generates vast numbers of duplicate packets, unless some measures are taken to damp the process
- One such measure is to have a **hop counter** contained in the header of each packet that is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination. If the sender doesn't know how long the path is, it can initialize the counter to the worst case, namely, the full diameter of the network. Flooding with a hop count can produce an exponential number of duplicate packets as the hop count grows & routers duplicate packets they have seen before.
- A better technique for blocking the flood is to have **routers keep track of which packets have been flooded**, to avoid sending them out a second time. One way to achieve this goal is to have the **source router put a sequence number in each packet** it receives from its hosts. When a packet comes in, it is easy to check if the packet has already been flooded (by comparing its sequence number to k; if so, it is discarded.
- **Flooding** is not practical for sending most packets, but it has some important uses:
    - It **ensures that a packet is delivered to every node in the network**. This **may be wasteful** if there is a single destination that needs the packet, **but** it is **effective for broadcasting information**. In wireless networks, all messages transmitted by a station can be received by all other stations within its radio range, which is, in fact, flooding, & some algorithms utilize this property.
    - Flooding is tremendously strong. Even if large numbers of routers are blown to bits (e.g., in a **military** network located **in a war zone**), flooding will find a path if one exists, to get a packet to its destination.
    - Flooding also requires little in the way of setup. The routers only need to know their neighbours. This means that **flooding can be used as a building block for other routing algorithms** that are more efficient but need more in the way of setup.
    - Flooding can also be used as a metric against which other routing algorithms can be compared. **Flooding always chooses the shortest path** because it chooses every possible path in parallel. Consequently, no other algorithm can produce a shorter delay.

- **Flooding** may be of **three types**:
    - Uncontrolled flooding − Here, each router unconditionally transmits the incoming data packets to all its neighbours
    - Controlled flooding − They use some methods to control the transmission of packets to the neighbouring nodes. The two popular algorithms for controlled flooding are Sequence Number Controlled Flooding (SNCF) and Reverse Path Forwarding (RPF)
    - Selective flooding − Here, the routers don't transmit the incoming packets only along those paths which are heading towards approximately in the right direction, instead of every available paths



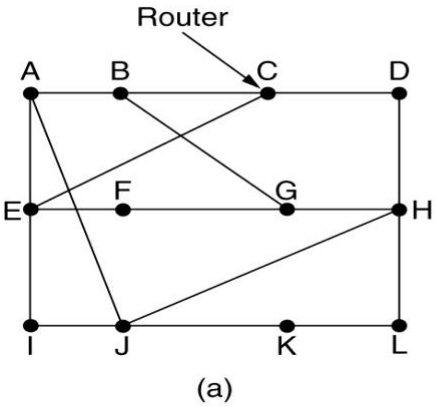

- In the above network, using flooding technique:
    - An incoming packet to A, will be sent to B, C and D.
    - B will send the packet to C and E.
    - C will send the packet to B, D and F.
    - D will send the packet to C and F.
    - E will send the packet to F.
    - F will send the packet to E.

## Distance Vector Routing

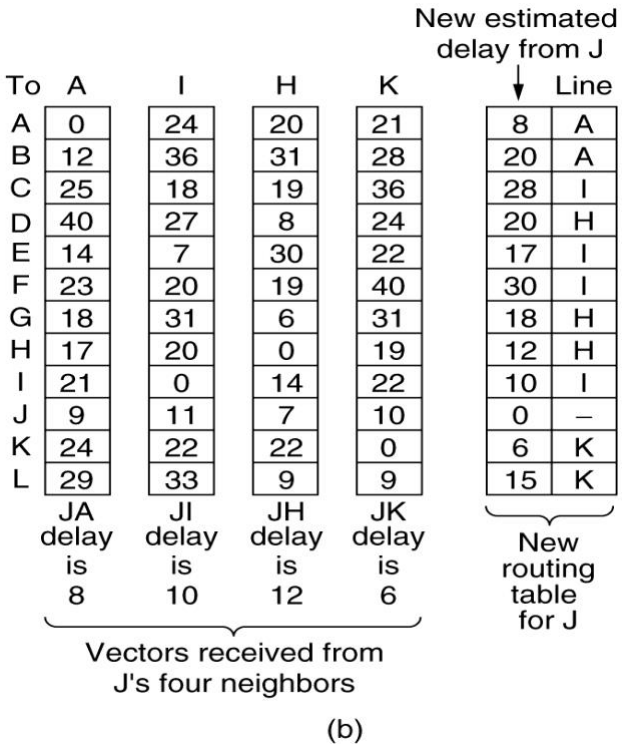
- A dynamic routing algorithm
- More complex than flooding, but more efficient, because it finds shortest paths for the current topology

- Two dynamic algorithms are most popular:
    - Distance vector routing (DVR)
    - Link state routing (LSR)
- In distance vector routing algorithm, each router maintains a table (i.e., a vector) having best-known distance to each destination & which link to use to get there. These tables are updated by exchanging information with the neighbours. Eventually, every router knows the best link to reach each destination.
- The **distance vector routing algorithm** is sometimes called by other names, most commonly the distributed **Bellman-Ford routing algorithm**, after the researchers who developed it (Bellman, 1957; & Ford & Fulkerson, 1962)
- It was the **original ARPANET routing algorithm** & was also used in the Internet under the name RIP (Routing Information Protocol).
- In distance vector routing, **each router maintains a routing table** indexed by, & containing one entry for each router in the network. This entry has 2 parts:
    - an outgoing line to the destination
    - an estimate of the distance to the destination
- The distance might be measured as number of hops or using another metric
- The router is assumed to know the "distance" to each of its neighbours. If the metric is hops, the distance is just one hop. If the metric is propagation delay, the router can measure it directly with special ECHO packets that the receiver just timestamps & sends back as fast as it can.
- As an example, assume that delay is used as a metric & the router knows the delay to each of its neighbours. Once every ***T*** msec, each router sends to each neighbour a list of its estimated delays to each destination. It also receives a similar list from each neighbour.
- Imagine that one of these tables has just come in from neighbour X, with Xi being X's estimate of how long it takes to get to router i. If the router knows that the delay to X is ***m*** msec, it also knows that it can reach router i via X in ***Xi + m*** msec.
- By performing this calculation for each neighbour, a router can find out which estimate seems the best & use that estimate & the corresponding link in its new routing table. The old routing table is not used in the calculation. This updating process is illustrated in the below figure (a network & input from A, I, H, K, & the new routing table for J).
- Part (a) shows a network. The 1st 4 columns of part (b) show the delay vectors received from the neighbours of router J. A claim to have a ***12***-msec delay to B, a ***25***-msec delay to C, a ***40***-msec delay to D, etc. Suppose that J has

measured or estimated its delay to its neighbours, A, I, H, & K, as ***8***, ***10***, ***12***, & ***6*** msec, respectively.




(a)



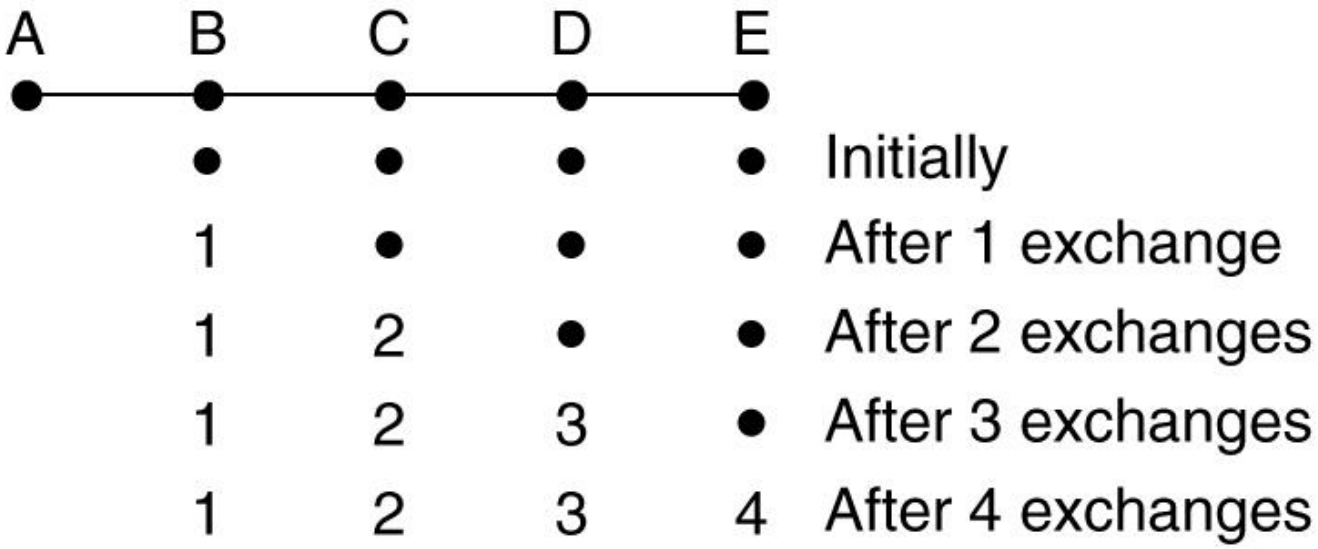

(b)

- Consider how J computes its new route to router G. It knows that it can get to A in ***8*** msec, & furthermore A claims to be able to get to G in ***18*** msec, so J knows it can count on a delay of ***26*** msec to G if it forwards packets bound for G to A
- Similarly, it computes the delay to G via I, H, & K as ***41*** (31 + 10), ***18*** (6 + 12), & ***37*** (31 + 6) msec, respectively
- The best of these values is ***18***, so it makes an entry in its routing table that the delay to G is ***18*** msec & that the route to use is via H. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the above figure.

## The Count-to-Infinity Problem

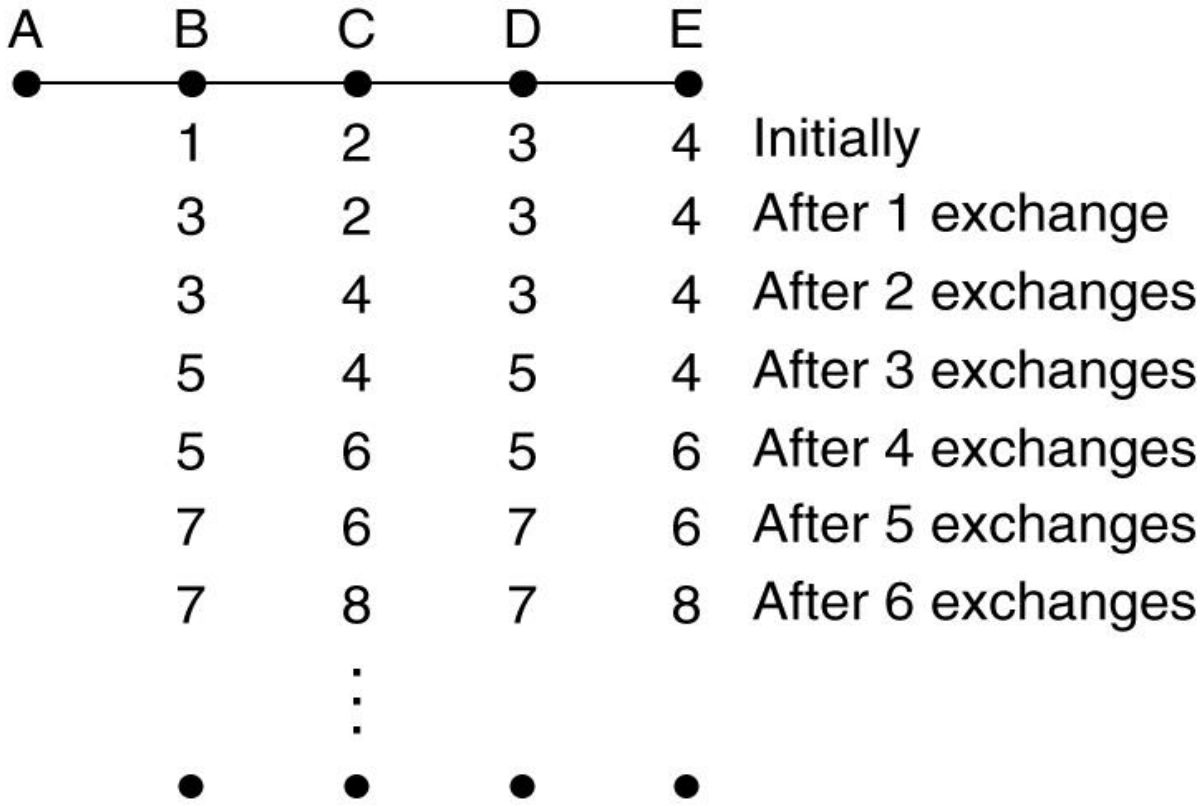
- Settling of routes to best paths across a network is called **convergence**
- Distance vector routing is useful as a simple technique by which routers can collectively compute shortest paths, but it has a serious drawback in practice: although it converges to the correct answer, it may do so slowly
- In particular, it reacts rapidly to good news, but leisurely to bad news
- Consider a router whose best route to destination X is long. If, on the next exchange, neighbour A suddenly reports a short delay to X, the router just

switches over to using the line to A to send traffic to X. In one vector exchange, the good news is processed. To see how fast good news propagates, consider the five-node (linear) network of below figure, where the delay metric is the number of hops. Suppose A is down initially & all the other routers know this. So, all routers have recorded the delay to A as infinity.




- When A comes up, the other routers learn about it via the vector exchanges
- At the time of the first exchange, B learns that its left-hand neighbour has zero delay to A. B now makes an entry in its routing table indicating that A is one hop away to the left. All the other routers still think that A is down. At this point, the routing table entries for A are as shown in the second row of above figure.
- On the next exchange, C learns that B has a path of length 1 to A, so it updates its routing table to indicate a path of length 2, but D & E don't hear the good news until later
- Clearly, the good news is spreading at the rate of one hop per exchange. In a network whose longest path is of length N hops, everyone will know about the new links & routers within N exchanges.
- Now let us consider the situation of the figure below, in which all the links & routers are initially up
- Routers B, C, D, & E have distances to A of 1, 2, 3, & 4 hops, respectively. Suddenly, either A goes down or the link between A & B is cut.
- At the first packet exchange, B doesn't hear anything from A. Fortunately, C says ***"Don't worry; I have a path to A of length 2"***.

- B doesn't think that C's path runs through B itself. For all B knows, C might have ten links all with separate paths to A of length 2. As a result, B thinks it can reach A via C, with a path length of 3.




- D and E don't update their entries for A on the first exchange
- On the second exchange, C notices that each of its neighbours claims to have a path to A of length 3. It picks one of them at random & makes its new distance to A 4, as shown in the third row of above figure.
- Subsequent exchanges produce the history shown in the rest of the above figure
- From this figure, it should be clear why bad news travels slowly: no router ever has a value more than one higher than the minimum of all its neighbours. Gradually, all routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity.
- For this reason, it is wise to set infinity to the longest path plus 1. Not entirely surprisingly, this problem is known as the **count-to-infinity problem**.
- There have been many attempts to solve it, for example, **preventing routers from advertising their best paths back to the neighbours** from which they heard them with the split horizon with poisoned reverse rule discussed in RFC (Request for Comments) 1058 (Routing Information Protocol).

- The core of the problem is that **when X tells Y that it has a path somewhere, Y has no way of knowing whether it itself is on the path**.

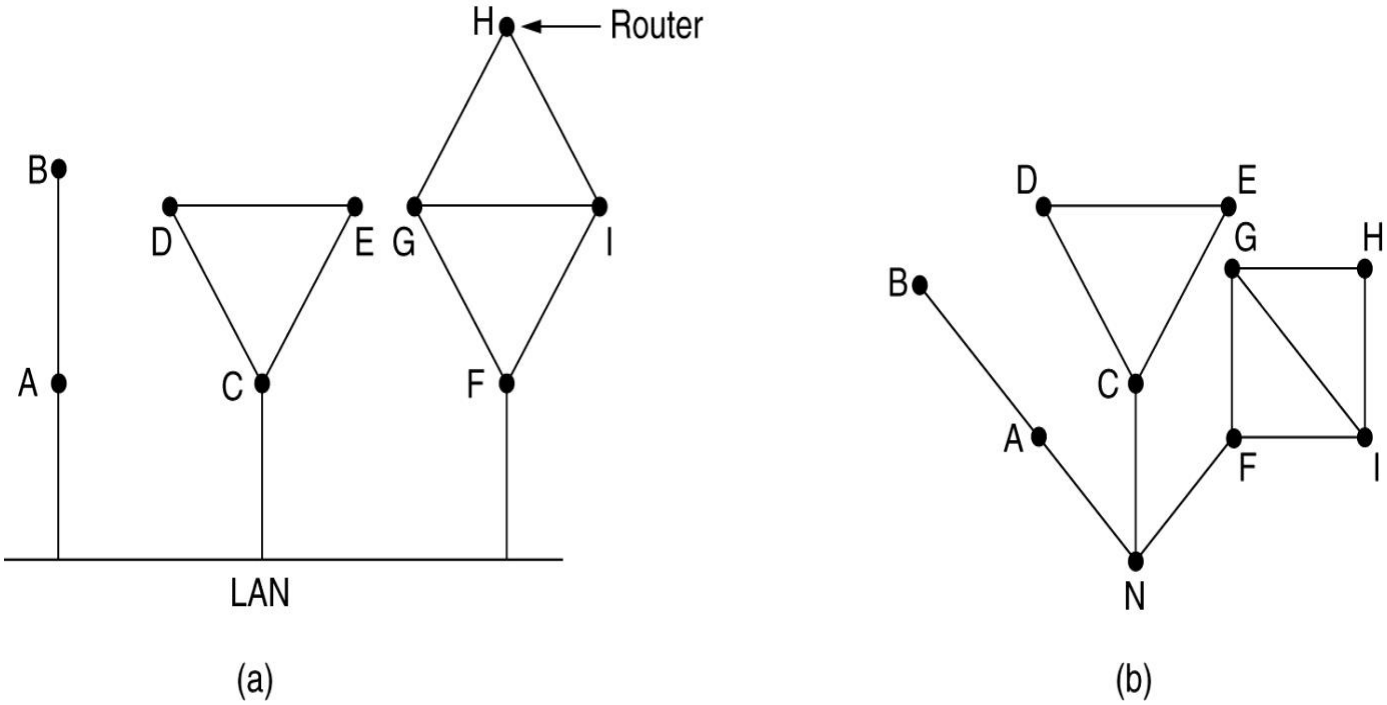
## Link State Routing

- **Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing**
- The primary problem about DVR is that it took too long to converge after the network topology changed (due to the count-to-infinity problem)
- Consequently, it was replaced by an entirely new algorithm, now called **link state routing (LSR)**
- Variants of link state routing called **IS-IS** & **OSPF are the routing algorithms that are most widely used inside large networks & the Internet today**
- The **idea behind link state routing** is simple & can be stated as 5 parts. Each router must do the following things to make it work:
    1. Discover its neighbours & learn their network addresses
    2. Set the distance or cost metric to each of its neighbours
    3. Construct a packet telling all it has just learned
    4. Send this packet to & receive packets from all other routers
    5. Compute the shortest path to every other router
- Complete topology is distributed to every router
- Then Dijkstra's algorithm can be run at each router to find the shortest path to every other router
- We will consider each of these 5 steps in more detail:

### Learning about the Neighbours

- When a router is booted, its first task is to learn who its neighbours are
- It accomplishes this goal by sending a special HELLO packet on each point-to-point line
- The router on the other end is expected to send back a reply giving its name. These names must be globally unique because when a distant router later hears that three routers are all connected to F, it is essential that it can determine whether all three mean the same F.
- When two or more routers are connected by a broadcast link (e.g., a switch, ring, or classic Ethernet), the situation is slightly more complicated
- The first illustration in the below figure (9 routers & a broadcast LAN) illustrates a broadcast LAN to which three routers, A, C, & F, are directly connected. Each of these routers is connected to one or more additional routers, as shown.

- The broadcast LAN provides connectivity between each pair of attached routers. But, **modelling the LAN as many point-to-point links increases the size of the topology & leads to wasteful messages**.




- A better way to model the **LAN** is to consider it **as a node** itself, as shown in the 2nd illustration in the above figure (a graph model of these 9 routers). Here, we have introduced a new, artificial node, N, to which A, C, & F are connected.
- One designated router on the LAN is selected to play the role of N in the routing protocol. The fact that it is possible to go from A to C on the LAN is represented by the path ANC here.

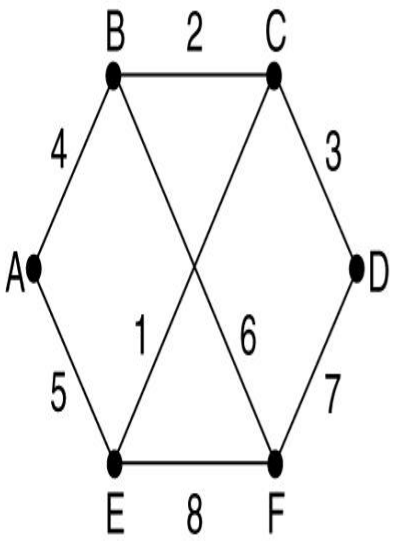
**<u>Setting Link Costs</u>**

- In link state routing algorithm, each link must have a distance or cost metric to find shortest paths
- The cost to reach neighbours can be set automatically, or configured by the network operator
- A common choice is to make the cost inversely proportional to the bandwidth of the link
- For example, 1-Gbps Ethernet may have a cost of 1 & 100-Mbps Ethernet a cost of 10. This makes higher-capacity paths better choices.
- If the network is geographically spread out, the delay of links may be factored into cost so that paths over shorter links are better choices

- The most direct way to determine this delay is to send a special ECHO packet over the line that other side is required to send back immediately. By measuring the round-trip time & dividing it by two, the sending router can get a reasonable estimate of the delay.

**Building Link State Packets**

- Once the information needed to exchange is collected, the next step for each router is to build a packet containing all the data
- The packet **starts with the identity of the sender**, followed by a sequence number & age & a list of neighbours. The cost to each neighbour is also given.
- An example network is presented in the 1st illustration in the below figure (a network) with costs shown as labels on the lines
- The corresponding link state packets for all 6 routers are shown in the 2nd figure below (the link state packets for this network).




(a)

Link State Packets

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| Seq. | Seq. | Seq. | Seq. | Seq. | Seq. |
| Age | Age | Age | Age | Age | Age |
| B 4 | A 4 | B 2 | C 3 | A 5 | B 6 |
| E 5 | C 2 | D 3 | F 7 | C 1 | D 7 |
| | F 6 | E 1 | | F 8 | E 8 |

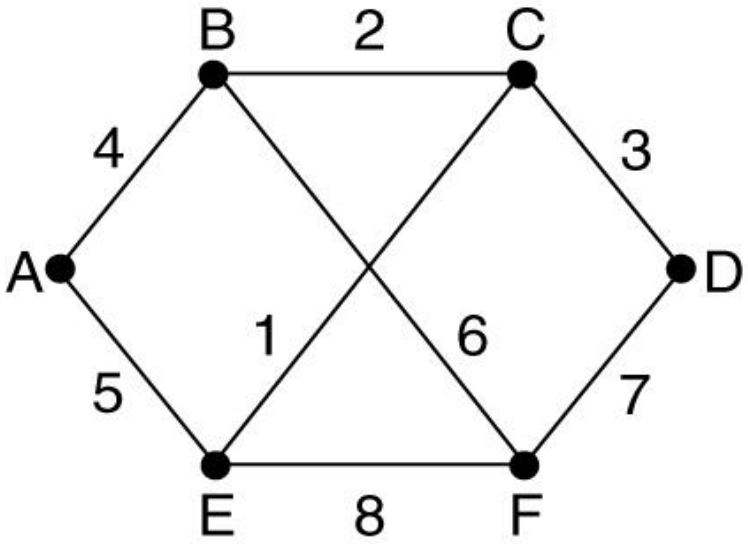
(b)

- Building the link state packets is easy. The hard part is determining when to build them.
- One possibility is to build them periodically, that is, at regular intervals
- Another possibility is to build them when some significant event occurs, such as a line or neighbour going down or coming back up again or changing its properties appreciably.

## Distributing the Link State Packets

- Trickiest part of this algorithm
- All of the routers must get all of the link state packets quickly & reliably
- If different routers use different versions of the topology, the routes they compute can have inconsistencies such as loops, unreachable machines, & other problems
- The fundamental idea is to **use flooding to distribute the link state packets to all routers**. To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see. When a new link state packet comes in, it is checked against the list of packets already seen. If it is new, it is forwarded on all lines except the one it arrived on. If it is a duplicate, it is discarded.
- If a packet with a sequence number lower than the highest one seen so far ever arrives, it is rejected as being obsolete as the router has more recent data
- A confusion will arise, if the sequence numbers wrap around. The solution here is to use a 32-bit sequence number. With one link state packet per second, it would take 137 years to wrap around, so this possibility can be ignored.
- Also, if a router ever crashes, it will lose track of its sequence number. If it starts again at 0, the next packet it sends will be rejected as a duplicate which is another issue.
- Nevertheless, if a sequence number is ever corrupted & 65,540 is received instead of 4 (a 1-bit error), packet 5 will be rejected as outdated, since the current sequence number is thought as 65,540
- The solution to all these problems is to include the age of each packet after the sequence number & decrement it once per second. When the age hits zero, the information from that router is discarded.
- Normally, a new packet comes in, suppose, every 10 sec, so router information only times out when a router is down (or six consecutive packets have been lost, an unlikely event)
- The Age field is also decremented by each router during the initial flooding process, to make sure no packet can get lost & live for an indefinite period of time (a packet whose age is zero is discarded)
- When a link state packet comes in to a router for flooding, it is not queued for transmission immediately. Instead, it is put in a holding area to wait for a while, in case, if more links are coming up or going down.
- If another link state packet from the same source comes in before the first packet is transmitted, their sequence numbers are compared. If they are equal, the duplicate is discarded. If they are different, the older one is thrown out.

- To guard against errors on the links, all link state packets are acknowledged.




(a)

- The data structure used by router B for the above network is depicted in the table (the packet buffer for router B in above network) below:

| | | | Send flags | | | ACK flags | | | |
|---|---|---|---|---|---|---|---|---|---|
| Source | Seq. | Age | A | C | F | A | C | F | Data |
| A | 21 | 60 | 0 | 1 | 1 | 1 | 0 | 0 | |
| F | 21 | 60 | 1 | 1 | 0 | 0 | 0 | 1 | |
| E | 21 | 59 | 0 | 1 | 0 | 1 | 0 | 1 | |
| C | 20 | 60 | 1 | 0 | 1 | 0 | 1 | 0 | |
| D | 21 | 59 | 1 | 0 | 0 | 0 | 1 | 1 | |

- Each row here corresponds to a recently arrived, but not fully processed, link state packet
- The table records where the packet originated, its sequence number & age, & the data

- In addition, there are send & acknowledgement flags for each of B's three links (to A, C, & F, respectively)
- The send flags mean that the packet must be sent on the indicated link
- The acknowledgement flags mean that it must be acknowledged there.
- In the above table, the link state packet arrives directly from A, so it must be sent to C & F & acknowledged to A, as indicated by the flag bits
- Similarly, the packet from F has to be forwarded to A & C & acknowledged to F
- But, the situation with the third packet, from E, is different. It arrives twice, once via EAB & once via EFB. Consequently, it has to be sent only to C but must be acknowledged to both A & F, as indicated by the bits. If a duplicate arrives while the original is still in the buffer, bits have to be changed.
- For example, if a copy of C's state arrives from F before the fourth entry in the table has been forwarded, the six bits will be changed to 100011 to indicate that the packet must be acknowledged to F but not sent there.

**Computing New Routes**

- Once a router accumulated a full set of link state packets, it can construct the entire network graph because every link is represented
- Every link is represented twice, once for each direction. The different directions may even have different costs.

- Dijkstra's algorithm can be run locally to construct the shortest paths to all possible destinations. The result of this algorithm tells the router which link to use to reach each destination. This information is installed in the routing tables, & normal operation is resumed.
- **Compared to distance vector routing, link state routing requires more memory and computation**
- For a network with ***n*** routers, each of which has ***k*** neighbours, the memory required to store the input data is proportional to ***kn***, which is at least as large as a routing table listing all the destinations
- Also, the computation time grows faster than ***kn***, even with the most efficient data structures, an issue in large networks
- But, in many practical situations, link state routing works well because it doesn't suffer from slow convergence problems
- Link state routing is widely used in actual networks
- Many ISPs use the **IS-IS (Intermediate System-Intermediate System) link state protocol** (Oran, 1990)

    - Designed for an early network called **DEC net** (a suite of network protocols created by Digital Equipment Corporation)
    - Later adopted by ISO for use with the OSI protocols
    - Later modified to handle other protocols as well, like, IP.
- **OSPF (Open Shortest Path First)** is another main link state protocol
    - Designed by IETF several years after IS-IS
    - Adopted many of the innovations designed for IS-IS
    - These innovations include
        - A self-stabilizing method of flooding link state updates;
        - The concept of a designated router on a LAN; &
        - The method of computing & supporting path splitting & multiple metrics.
- As a consequence, there is very little difference between **IS-IS** & **OSPF**
- The most important difference is that **IS-IS can carry information about multiple network layer protocols at the same time (e.g., IP, IPX, & AppleTalk)**
- **OSPF doesn't have this feature**, & it is an advantage in large multiprotocol environments

- Link state, distance vector, & other algorithms rely on processing at all the routers to compute routes
- Problems with the hardware or software at even a small number of routers can create disorder across the network
- For example, if a router claims to have a link it doesn't have, or forgets a link it has, the network graph will be incorrect
- If a router fails to forward packets or corrupts them while forwarding them, the route will not work as expected
- Finally, if it runs out of memory or does the routing calculation wrong, bad things will happen
- As the network grows into the range of tens or hundreds of thousands of nodes, the probability of some router failing occasionally becomes nonnegligible. The solution is to limit the damage when the unavoidable situation happens.

Difference between link state routing &distance vector routing

| Distance vector routing | Link state routing |
|---|---|
| Used in 1980 'S | Used in 1990"s |
| Band width is less | Band width is high |
| Traffic is les | Traffic is high |
| Count to infinity problem exist | Count to infinity problem not exists |

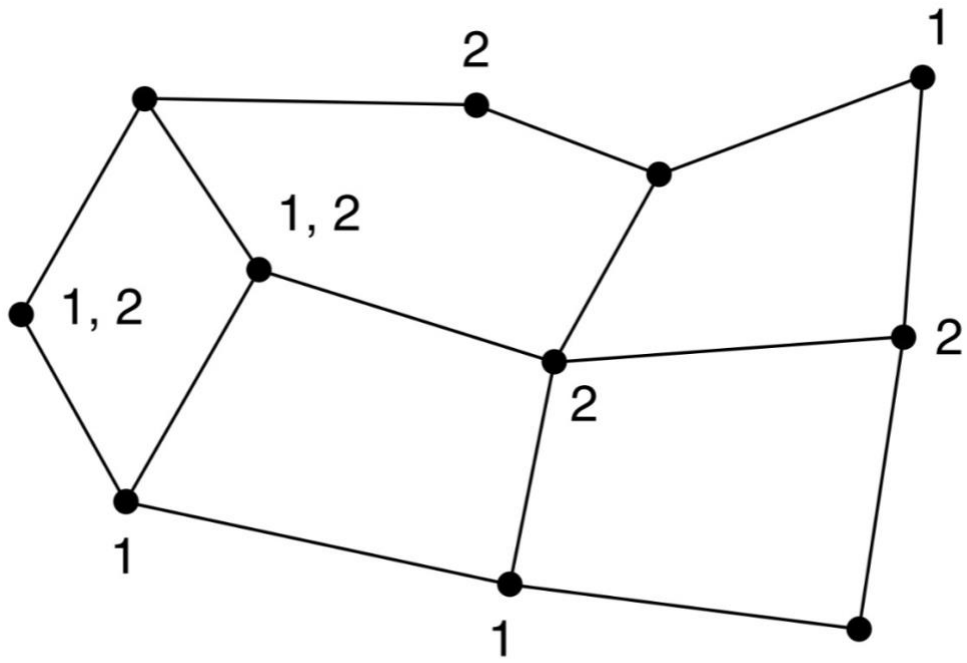
| Persistent loop | Transient loop |
|---|---|
| Protocol used is RIP | Protocol used is OSPF |
| Convergence is slow | Convergence is fast |

## Multicast Routing

- Applications like a multiplayer game or live video of a sports event streamed to many viewing locations, send packets to multiple receivers
- Unless the group is very small, sending a distinct packet to each receiver is expensive
- Broadcasting a packet is also wasteful if the group consists of 1000 machines on a million-node network, so that most receivers are not interested in the message, or, they are interested but are not supposed to see it.
- Thus, we need a way to send messages to well-defined groups that are numerically large in size but small compared to the network as a whole
- Sending a message to such a group is called **multicasting**, and the routing algorithm used is called **multicast routing**
- Multicasting schemes need some way to create & destroy groups & to identify which routers are members of a group
- Suppose, each group is identified by a multicast address & that routers know the groups to which they belong
- Multicast routing schemes build on broadcast routing schemes, sending packets along spanning trees (a subset of a network, having all its vertices

covered with minimum possible number of edges), to deliver the packets to the members of the group while making efficient use of bandwidth

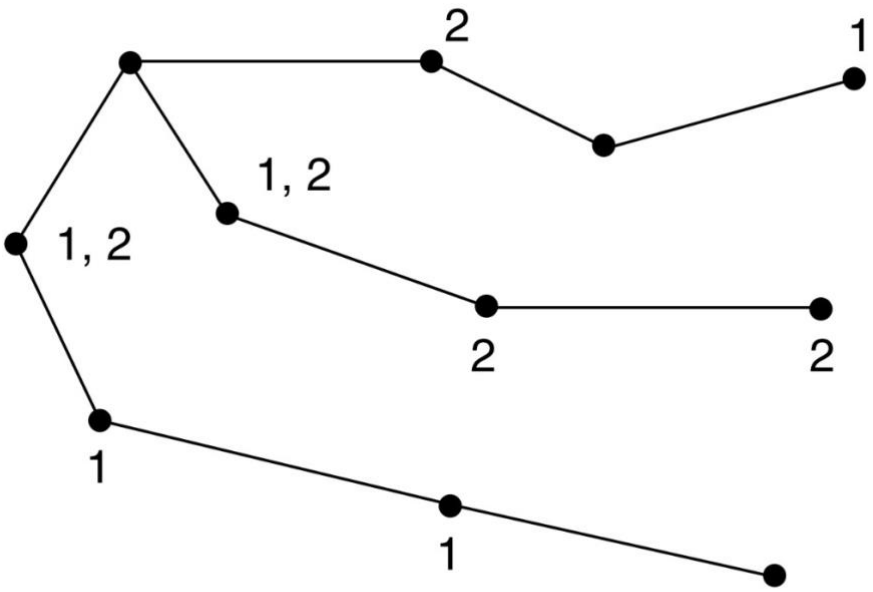
- But, the best spanning tree to use depends on **whether the group is dense**, with receivers scattered over most of the network, **or sparse**, with much of the network not belonging to the group
- **If** the group is **dense**, **broadcast is** a **good** start because it efficiently gets the packet to all parts of the network. **But broadcast will reach some routers that are not members of the group, which is wasteful**. The solution explored by Deering & Cheriton (1990) is to prune (trim) the broadcast spanning tree by removing links that do not lead to members. The result is an **efficient multicast spanning tree**.
- As an example, consider 2 groups, 1 & 2, in the network shown below:




(a)

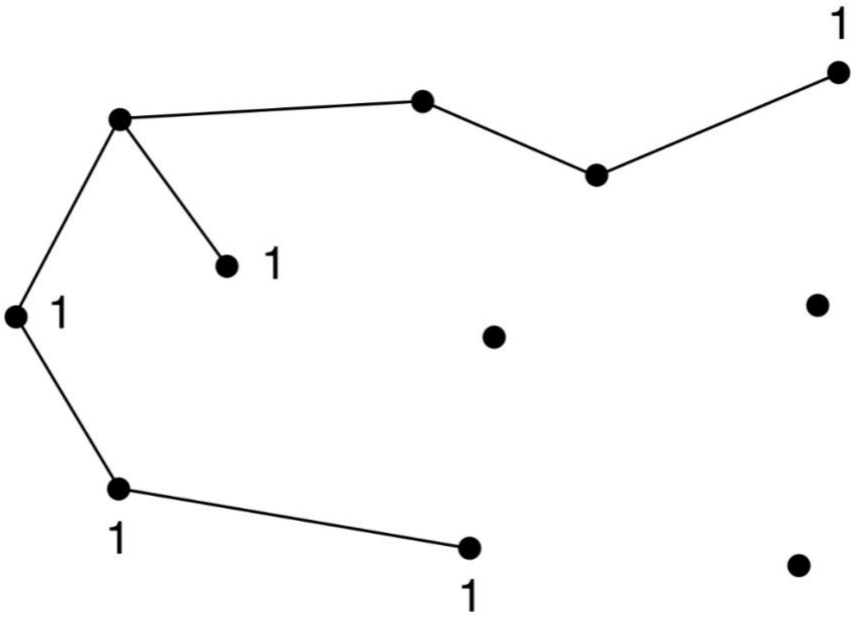
- Some routers are attached to hosts that belong to one or both of these groups (groups 1 & 2)

- A spanning tree for the leftmost router is shown in the below figure:



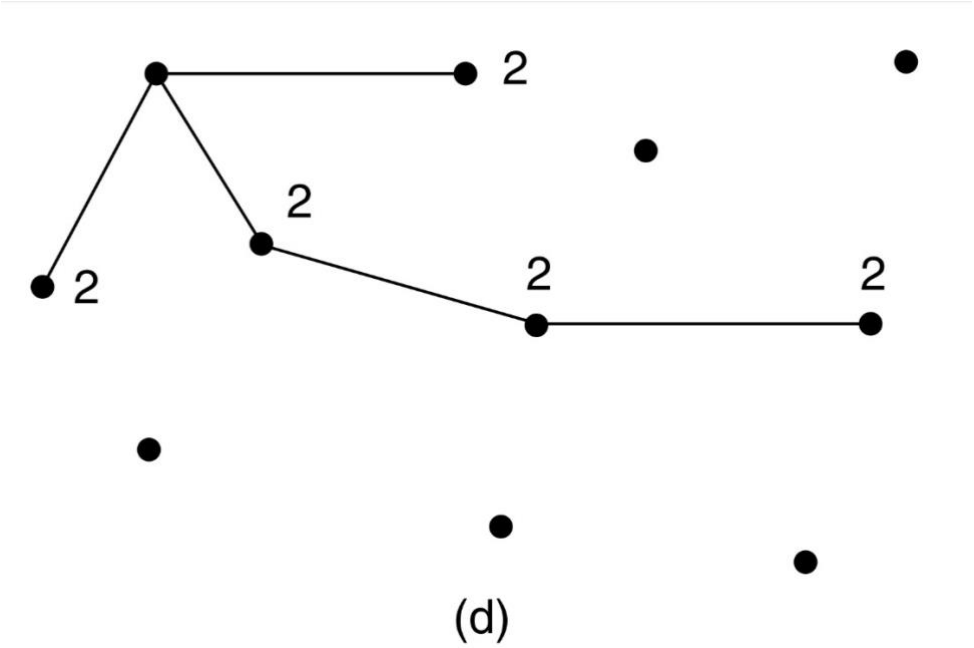

(b)

- This tree can be used for broadcast but is excess for multicast, as can be seen from the two pruned versions that are shown next.




(c)

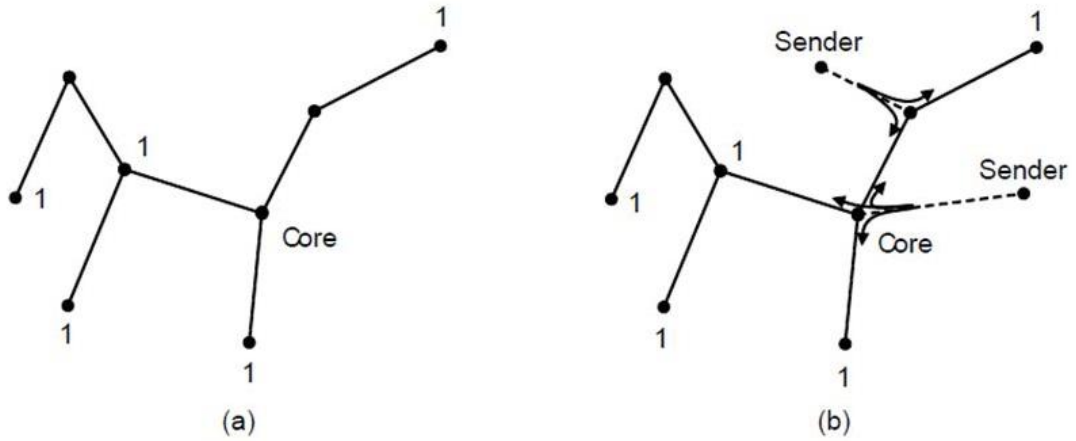
- In the above figure (a multicast tree for group 1), all the links that don't lead to hosts that are members of group 1 have been removed
- The result is the multicast spanning tree for the leftmost router to send to group 1
- Packets are forwarded only along this spanning tree, which is more efficient than the broadcast tree because there are 7 links instead of 10
- The below graph shows the multicast spanning tree after pruning for group 2:




- It is efficient, with only 5 links this time. It also shows that different multicast groups have different spanning trees.
- Various ways of pruning the spanning tree are possible.
- The simplest one can be used if **link state routing** is used & each router is aware of the complete topology, including which hosts belong to which groups. Each router can then construct its own pruned spanning tree for each sender to the group in question by constructing a sink tree for the sender as usual & then removing all links that don't connect group members to the sink node. **MOSPF (Multicast OSPF)** is an example of a link state protocol that works in this way.
- With **distance vector routing**, a different pruning strategy can be followed. The basic algorithm is **reverse path forwarding**. But, whenever a router with no hosts interested in a particular group & no connections to other routers receives a multicast message for that group, it responds with a PRUNE (trim) message, telling the neighbour that sent the message not to send it any more multicasts from the sender for that group. When a router with no group

members among its own hosts has received such messages on all the lines to which it sends the multicast, it, too, can respond with a PRUNE message. In this way, the spanning tree is recursively pruned. **DVMRP (Distance Vector Multicast Routing Protocol)** is an example of a multicast routing protocol that works this way.

- Pruning results in efficient spanning trees that use only the links that are actually needed to reach members of the group
- One disadvantage is that it is lots of work for routers, especially for large networks. Suppose that a network has ***n*** groups, each with an average of ***m*** nodes. At each router and for each group, ***m*** pruned spanning trees must be stored, for a total of ***mn*** trees.
- Consider the above example, the graph showing spanning tree for the leftmost router to send to group 1. The spanning tree for the rightmost router to send to group 1 (not shown) will look quite different, as packets will head directly for group members rather than via the left side of the graph. This in turn means that routers must forward packets destined to group 1 in different directions depending on which node is sending to the group. When many large groups with many senders exist, considerable storage is needed to store all the trees.
- An alternative design uses **core-based trees** to compute a single spanning tree for the group. All of the routers agree on a **roo**t (called the **core or rendezvous point**) & build the tree by sending a packet from each member to the root. The tree is the union of the paths traced by these packets.



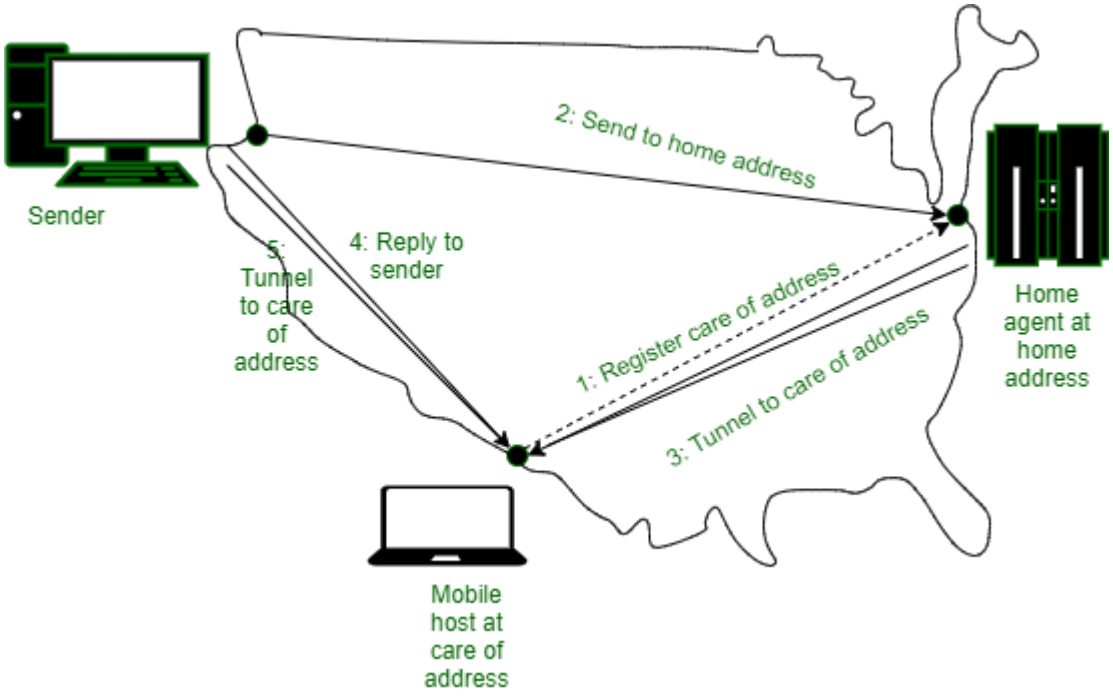

(a) Core-based tree for group 1.
(b) Sending to group 1.

- The above figure shows a core-based tree for group 1. To send to this group, a sender sends a packet to the core. When the packet reaches the core, it is forwarded down the tree. This is shown in (b) for the sender on the righthand side of the network. As a performance optimization, packets destined for the group don't need to reach the core before they are multicast. As soon as a packet reaches the tree, it can be forwarded up toward the root, as well as down all the other branches. This is the case for the sender at the top of figure (b).
- Having a shared tree is not optimal for all sources. For example, in the above figure (b), the packet from the sender on the righthand side reaches the top-right group member via the core in three hops, instead of directly. The inefficiency depends on where the core & senders are located, but often it is reasonable when the core is in the middle of the senders. When there is only a single sender, as in a video that is streamed to a group, using the sender as the core is optimal.
- Also, shared trees can be a major savings in storage costs, messages sent, & computation. Instead of ***m*** trees, each router has to keep only one tree per group.
- Also, routers that are not part of the tree need not work at all to support the group. For this reason, shared tree approaches like core-based trees are used for multicasting to sparse groups in the Internet as part of popular protocols such as PIM (Protocol Independent Multicast).

## Routing for Mobile Hosts

- Millions of people use computers, from mobile situations with wireless devices in moving cars, to nomadic situations in which laptop computers are used in a series of different locations
- We use the term mobile hosts to mean a category, as that is distinct from stationary hosts that never move
- People want to stay connected wherever in the world they may be, as easily as if they were at home
- These mobile hosts introduce a new complication: **to route a packet to a mobile host, the network first has to find it**
- The model of the world that we will consider is one in which **all hosts** are assumed to **have a permanent home location** that never changes. **Each host** also **has a permanent home address** that can be used to determine its home location (analogous (similar) to the way the telephone number 1-212-5551212 indicates the United States (country code 1) & Manhattan (212)).

- The goal of routing in mobile hosts is to make it possible to send packets to mobile hosts using their fixed home addresses & have the packets efficiently reach them wherever they may be.
- The trick, is to find them. A different model would be to recompute routes as the mobile host moves & the topology changes.
- We can use the routing schemes we have learnt earlier. But, with a growing number of mobile hosts, this would soon lead to the entire network endlessly computing new routes.
- Using the home addresses greatly reduces this burden. Another alternative would be to provide mobility above the network layer, which is what typically happens with laptops today. **When they are moved to new Internet locations, laptops acquire new network addresses**. There is no association between the old & new addresses; the network doesn't know that they belonged to the same laptop.
- In this model, a laptop can be used to browse the Web, but other hosts can't send packets to it (for example, for an incoming call), without building a higher layer location service, for example, signing into Skype again after moving.
- Besides, connections can't be maintained while the host is moving; new connections must be started up instead
- Network-layer mobility is useful to fix these problems
- **The basic idea used for mobile routing for a mobile host in Internet & cellular networks is to tell a host at the home location where it is now**.
- This host, which acts on behalf of the mobile host, is called the **home agent.** Once it knows where the mobile host is currently located, it can forward packets so that they are delivered.
- The below figure (packet routing for mobile hosts) shows mobile routing in action. A sender in the northwest city of Seattle wants to send a packet to a host normally located across the United States in New York. The case of interest to us is when the mobile host is not at home. Instead, it is temporarily in San Diego.
- The mobile host in San Diego must acquire a local network address before it can use the network. This happens in normal way that hosts obtain network addresses.
- The local address is called a **care of address**. Once the mobile host has this address, it can tell its home agent where it is now. It does this by sending a registration message to the home agent (step 1) with the care of address. The message is shown with a dashed line in the figure to indicate that it is a control message, not a data message.

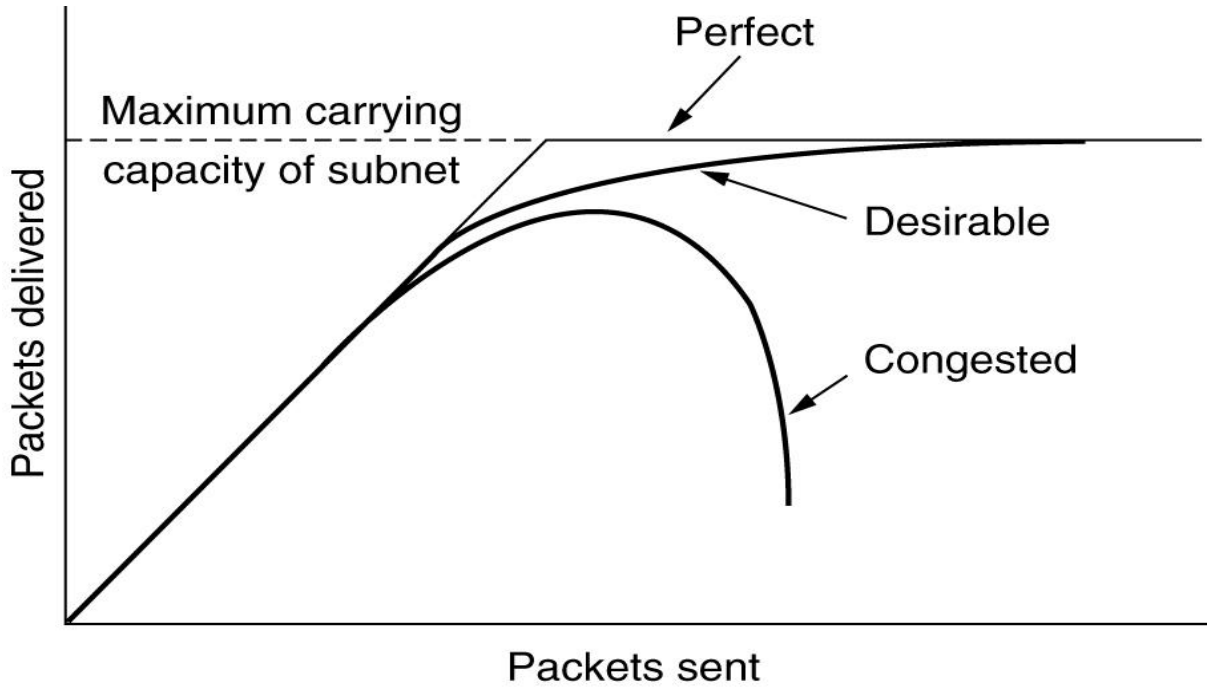



- Next, the sender sends a data packet to the mobile host using its permanent address (step 2). This packet is routed by the network to the host's home location because that is where the home address belongs. In New York, the home agent intercepts this packet because the mobile host is away from home. It then wraps or **encapsulates** the packet with a new header & sends this bundle to the **care of address** (step 3). This mechanism is called **tunnellin**g.
- When the encapsulated packet arrives at the care of address, the mobile host unwraps it & retrieves the packet from the sender. The mobile host then sends its reply packet directly to the sender (step 4).
- The overall route is called **triangle routing** because it may be circuitous if the remote location is far from the home location. As part of step 4, the sender may learn the current care of address. Subsequent packets can be routed directly to the mobile host by tunnelling them to the care of address (step 5), bypassing the home location entirely.
- If connectivity is lost for any reason as the mobile moves, the home address can always be used to reach the mobile
- An important aspect that we have omitted from this description is **security**
- Security information is included in the messages so that their validity can be checked with cryptographic protocols (a protocol that performs security-related function & applies cryptographic methods)
- There are many variations on mobile routing

- The scheme above is modelled on **IPv6 mobility**, the form of mobility used in the Internet & as part of IP-based cellular networks such as **UMTS (Universal Mobile Telecommunications System)**
- Here, the sender is a stationary node for simplicity, but the designs let both nodes be mobile hosts
- Alternatively, the host may be part of a mobile network, for example a computer in a plane
- Some schemes make use of a foreign (i.e., remote) agent, similar to home agent but at foreign location, or analogous to the VLR (Visitor Location Register) in cellular networks. But, in more recent schemes, foreign agent is not needed; mobile hosts act as their own foreign agents.
- In either case, knowledge of the temporary location of the mobile host is limited to a small number of hosts (e.g., the mobile, home agent, & senders) so that the many routers in a large network don't need to recompute routes.

## CONGESTION CONTROL ALGORITHMS

- When too many packets are present in the subnet, performance degrades. This situation is called **congestion**




- The above graph shows that, when too much traffic is offered, congestion sets in & performance degrades sharply

- When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered & the number delivered is proportional to the number sent.
- But, as traffic increases too far, the routers are no longer able to cope & they begin losing packets. This tends to make matters worse. At very high traffic, performance collapses completely & almost no packets are delivered.
- **Factor Causing Congestion:**
  Congestion can be brought on by several factors.
  - If all of a sudden, streams of packets begin arriving on three or four input lines & all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packets will **be lost**.
  - **Slow processors** can also cause congestion. If the routers' CPUs are slow at performing the bookkeeping tasks required of them (queuing buffers, updating tables, etc.), queues can build up, even though there is excess line capacity.
  - **Low-bandwidth lines** can also cause congestion. Upgrading the lines but not changing the processors, or vice versa, often helps a little, but frequently just shifts the bottleneck.

## General principles of congestion control

- Many problems in computer networks can be viewed from a control theory point of view. This approach leads to dividing all solutions into 2 groups:
  - **Open loop**
  - **Closed loop**

**Open Loop:**

- Solve the problem by good design, to make sure it doesn't occur in the first place
- Once the system is up & running, corrections are not made in between
- Tools for doing open-loop control include:
  - Deciding when to accept new traffic;
  - Deciding when to discard packets & which ones to discard; &
  - Making scheduling decisions at various points in the network
- All of these, in common, have the fact that they make decisions without regard to the current state of the network.

**Closed loop:**

- Closed loop solutions are based on the concept of a **feedback loop**.
- This approach has **three parts when applied to congestion control**:

1. Monitor the system to detect when & where congestion occurs.
2. Pass this information to places where action can be taken.
3. Adjust system operation to correct the problem.

- A variety of metrics can be used to monitor the subnet for congestion. Some important metrics are the percentage of all packets discarded for lack of buffer space, the average queue lengths, the number of packets that time out & are retransmitted, the average packet delay, & the standard deviation of packet delay.
- In all cases, rising numbers indicate growing congestion.
- **The second step** in feedback loop is to transfer information about congestion from the point where it is detected to the point where something can be done about it.
- The solution is to detect the congestion by sending a packet to the traffic source or sources, announcing the problem.

**Alternate possibility:**

- A bit or field can be reserved in every packet for routers to fill in whenever congestion gets above some threshold level.
- When a router detects this congested state, it fills in the field in all outgoing packets, to warn the neighbours.
- Another approach is to have hosts or routers periodically send probe packets out to explicitly ask about congestion. This information can then be used to route traffic around problem areas.

**Categories of Closed loop algorithm:**

- The closed loop algorithms are also divided into two subcategories:
  - **Explicit feedback**
  - **Implicit feedback**
- In **explicit feedback algorithms**, packets are sent back from the point of congestion to warn the source.
- In **implicit algorithms**, the source deduces the existence of congestion by making local observations, such as the time needed for acknowledgements to come back.

## Congestion prevention policies

- These are methods to control congestion by looking at open loop systems by using appropriate policies at various levels. Given below are some policies that affect congestion:

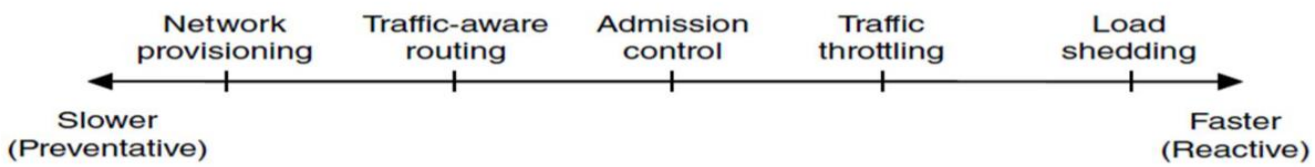
| Layer | Policies |
|---|---|
| Transport | • Retransmission policy<br>• Out-of-order caching policy<br>• Acknowledgement policy<br>• Flow control policy<br>• Timeout determination |
| Network | • Virtual circuits versus datagram inside the subnet<br>• Packet queueing and service policy<br>• Packet discard policy<br>• Routing algorithm<br>• Packet lifetime management |
| Data link | • Retransmission policy<br>• Out-of-order caching policy<br>• Acknowledgement policy<br>• Flow control policy |

- **Policies considered in the Data link layer**
  - Retransmission policy:
    - How fast a sender times out & what it transmits upon timeout
    - 'Go back N' will put a heavier load on the system than 'selective repeat'
  - Out-of-order caching policy:
    - 'Selective repeat' is clearly better than 'go back N'
  - Acknowledgement policy:
    - Piggybacking onto reverse traffic may help
    - But extra timeouts & retransmissions may happen.
  - Flow control Policy:
    - A small window reduces the data rate & thus helps fight congestion
- **Policies considered in the Network layer**
  - Choice of virtual circuits vs datagrams:
    - Affects congestion since many congestion control algorithms work only with virtual-circuit subnets
  - Packet queuing & service policy:

        - Relates to whether routers have one queue per input line, one queue per output line, or both
        - Also relates to the order in which packets are processed (e.g., round robin or priority based)
    - Packet Discard policy:
        - Rule telling which packet is dropped when there is no space
    - Routing algorithm:
        - Good algorithm can help avoid congestion by spreading the traffic over all the lines
    - Packet Lifetime Management
        - Deals with how long a packet may live before being discarded
        - If it is too long, lost packets may block up the works for a long time
        - If it is too short, packets may sometimes time out before reaching their destination, thus inducing retransmissions
- **Policies considered in the Transport layer**
    - Same issues as in data link layer plus
    - Timeout determination:
        - Determining if the timeout interval is harder because the transit time across the network is less predictable
        - If the timeout interval is too short, extra packets will be sent unnecessarily
        - If it is too long, congestion will be reduced but the response time will suffer whenever a packet is lost.

## Approaches to Congestion Control

- The presence of congestion means the load is (temporarily) greater than the resources (in a part of the network) can handle. Two solutions are:
    - Increase the resources; or
    - Decrease the load
- As shown here (timescales of approaches to congestion control), these solutions are usually applied on different time scales to either prevent congestion or react to it once it has occurred.



- **Network Provisioning**: The most basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If there is a low-bandwidth link on the path along which most traffic is directed, congestion is likely to happen.
    - Sometimes resources can be added dynamically when there is serious congestion, for example, turning on spare routers or enabling lines that are normally used only as backups (to make the system fault tolerant) or purchasing bandwidth on the open market.
    - More often, links & routers that are regularly heavily utilized are upgraded at the earliest opportunity. This is called **provisioning** & happens on a time scale of months, driven by long-term traffic trends.
- **Traffic-aware Routing:** To make most of the existing network capacity, routes can be tailored to traffic patterns that change during the day as network users wake & sleep in different time zones.
    - For example, routes may be changed to shift traffic away from heavily used paths by changing shortest path weights. Some local radio stations have helicopters flying around their cities to report on road congestion to make it possible for their mobile listeners to route their packets (cars) around hotspots. This is called **traffic-aware routing**.
    - Splitting traffic across multiple paths is also helpful.
- **Admission Control**: Sometimes it's not possible to increase capacity. The only way to beat the congestion is to decrease the load
    - In a **virtual-circuit network**, new connections can be refused if they would cause the network to become congested. This is called **admission control**.
- **Traffic Throttling**: At a finer granularity, when congestion is imminent the network can deliver feedback to the sources whose traffic flows are responsible for the problem. The network can request these sources to throttle their traffic, or it can slow down the traffic itself.
    - Two difficulties with this approach are **how to identify the onset of congestion, & how to inform the source that needs to slow down**.
    - To tackle the first issue, routers can monitor the average load, queueing delay, or packet loss. In all cases, rising numbers indicate growing congestion.
    - To tackle the second issue, routers must participate in a feedback loop with the sources. For a scheme to work correctly, the time scale must be adjusted carefully. If every time two packets arrive in a row, a router yells STOP & every time a router is idle for 20 μsec, it yells GO, the system will oscillate wildly & never converge. On the other

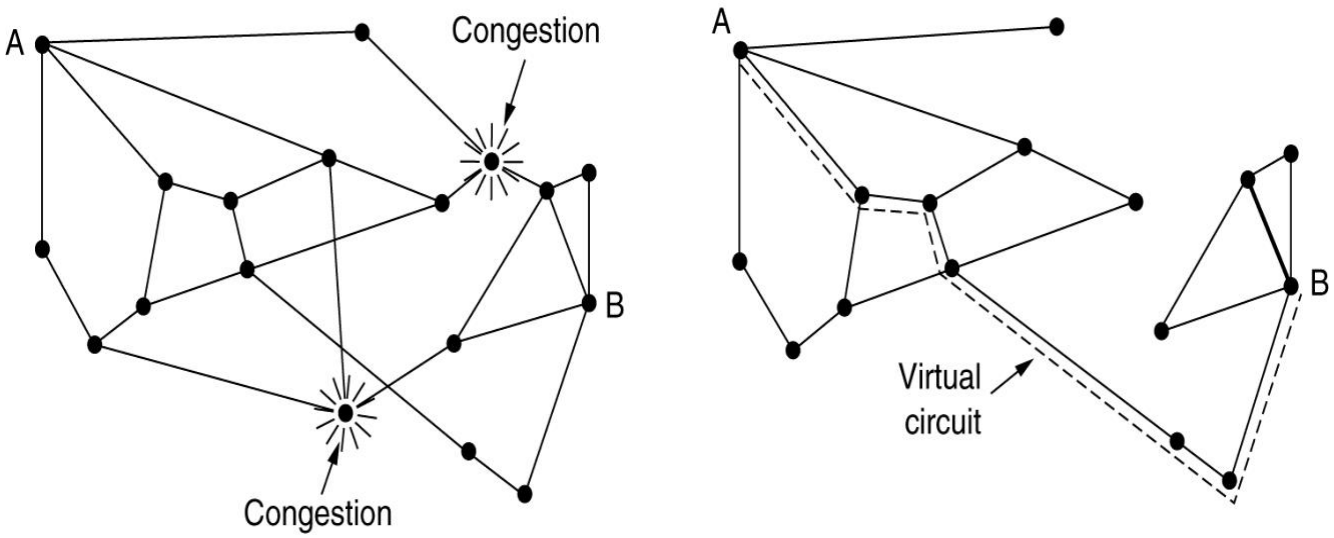
hand, if it waits 30 minutes to make sure before saying anything, the congestion-control mechanism will react too slowly to be of any use.
    - Delivering timely feedback is a significant matter. An added concern is having routers send more messages when the network is already congested.
- **Load Shedding**: When all the methods fail, the network is forced to discard packets that it can't deliver. The general name for this is **load shedding**.
    - A good policy for choosing which packets to discard can help to prevent congestion collapse.

## Congestion control in virtual circuit subnets

- One technique that is widely used in virtual-circuit networks to keep congestion at bay is **admission control**
- The idea is, do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested. Thus, attempts to set up a virtual circuit may fail.
- This is better than the alternative, as letting more people in when the network is busy just makes matters worse
- For example, in the telephone system, when a switch gets overloaded, it practices admission control by not giving dial tones. The trick with this approach is working out when a new virtual circuit will lead to congestion. The task is straightforward in the telephone network because of the fixed bandwidth of calls (64 kbps for uncompressed audio).
- But, virtual circuits in computer networks come in all shapes & sizes. Thus, the circuit must come with some characterization of its traffic if we are to apply admission control.
- Traffic is often described in terms of its rate & shape. The problem of how to describe it in a simple yet meaningful way is difficult because traffic is typically burst—the average rate is only half the story.
- For example, traffic that varies while browsing the Web is more difficult to handle than a streaming movie with the same long-term throughput because the bursts of Web traffic are more likely to congest routers in the network.
- A commonly used descriptor that captures this effect is the **leaky bucket** or **token bucket.** A **leaky bucket** has two parameters that bound the average rate & the instantaneous burst size of traffic.
- With traffic descriptions, the network can decide whether to admit the new virtual circuit. One possibility for the network is to reserve enough capacity along the paths of each of its virtual circuits that congestion will not occur. In

this case, the traffic description is a service agreement for what the network will guarantee its users.

- Even without making guarantees, the network can use traffic descriptions for admission control. The task is to estimate how many circuits will fit within the carrying capacity of the network without congestion.
- Suppose the virtual circuits that may blast traffic at rates up to 10 Mbps all pass through the same 100- Mbps physical link. How many circuits should be admitted? Clearly, 10 circuits can be admitted without risking congestion, but this is wasteful in the normal case since it may rarely happen that all 10 are transmitting full blast at the same time.
- In real networks, measurements of past behaviour that capture the statistics of transmissions can be used to estimate the number of circuits to admit, to trade better performance for acceptable risk.
- **Admission control** can be combined with **traffic-aware routing** by considering routes around traffic hotspots as part of the setup procedure
- For example, consider the network illustrated here (a congested network & the portion of the network that is not congested; a virtual circuit from A to B is also shown).




(a) (b)

- Suppose that a host attached to router A wants to set up a connection to a host attached to router B. Normally, this connection would pass through one of the congested routers. To avoid this situation, we can redraw the network as shown in figure (b), omitting the congested routers & all of their lines. The dashed line shows a possible route for the virtual circuit that avoids the congested routers.

## Congestion control in Datagram subnets

- In Internet & many other computer networks, senders adjust their transmissions to send as much traffic as the network can readily deliver. In this setting, the network operates just before the start of congestion.
- When congestion is forthcoming, it tells the senders to throttle back (reduce) their transmissions & slow down. This feedback is business as usual rather than an exceptional situation.
- Let's consider some approaches to throttle traffic that can be used in both **datagram networks** & virtual-circuit networks
- Each approach must solve 2 problems:
  - First, **routers must determine when congestion is approaching**, ideally before it has arrived
    - Each router can continuously monitor the resources it is using
    - Three possibilities are: utilization of the output links, **the buffering of queued packets inside the router**, & the number of packets that are lost due to insufficient buffering. Of these possibilities, the second one is the most useful.
    - A utilization of 50% may be low for smooth traffic & too high for highly variable traffic. Counts of packet losses come too late. Congestion has already set in by the time that packets are lost.
    - The queueing delay inside routers directly captures any congestion experienced by packets. It should be low most of time, but will jump when there is a burst of traffic that generates a backlog. To maintain a good estimate of the queueing delay, ***d***, a sample of the instantaneous queue length, ***s***, can be made periodically & ***d*** updated according to

      $$d_{new} = \alpha d_{old} + (1 - \alpha)\, s$$

      where the constant $\alpha$ **determines how fast the router forgets recent history**. This is called an **EWMA (Exponentially Weighted Moving Average).**

      It soothes out fluctuations & is equivalent to a low-pass filter. Whenever ***d*** moves above the threshold, the router notes the onset of congestion.

- The second problem is that **routers must deliver timely feedback to the senders that are causing the congestion**. Congestion is experienced in the network, but relieving congestion requires action on behalf of the senders that are using the network. To deliver feedback, the router must identify the appropriate senders. It must then warn them carefully, without sending many more packets into the already congested network. Different schemes use different feedback mechanisms:

## The Warning Bit

- Old DECNET architecture & frame relay signaled the warning state by setting a special bit in the packet's header
- When the packet arrived at its destination, the transport entity copied the bit into the next acknowledgement sent back to the source
- The source then cut back on traffic
- As long as the router was in the warning state, it continued to set the warning bit, which meant that the source continued to get acknowledgements with it set.
- The source monitored the fraction of acknowledgements with the bit set & adjusted its transmission rate accordingly.
- As long as the warning bits continued to flow in, the source continued to decrease its transmission rate.
- When they slowed to a trickle, it increases its transmission rate.
- Since every router along the path could set the warning bit, traffic is increased only when no router was in trouble.

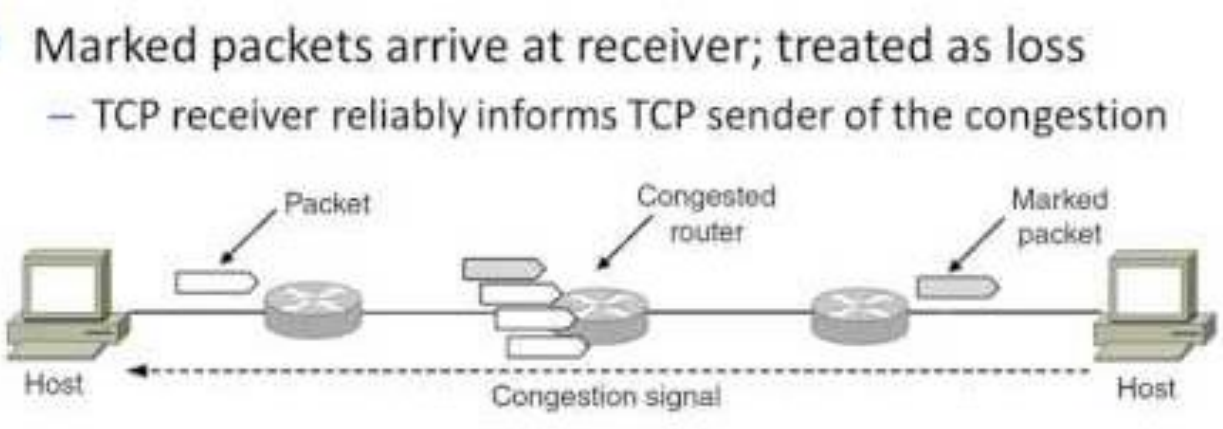
## Choke Packets

- Direct way to notify a sender of congestion is to tell it directly
- In this approach, the router selects a congested packet & sends a choke packet back to the source host, giving it the destination found in the packet
- The original packet may be tagged (a header bit is turned on) so that it won't generate any more choke packets farther along the path & then forwarded in the usual way

- To avoid increasing load on the network during a time of congestion, the router may only send choke packets at a low rate.
- When the source host gets the choke packet, it reduces the traffic sent to the specified destination, for example, by 50%
- In a datagram network, simply picking packets at random when there is congestion is likely to cause choke packets to be sent to fast senders, because they will have the most packets in the queue
- The feedback implicit in this protocol can help prevent congestion but not throttle any sender unless it causes trouble
- So, it is likely that multiple choke packets will be sent to a given host & destination. The host should ignore these additional chokes for the fixed time interval until its reduction in traffic takes effect. After that period, further choke packets indicate that the network is still congested.
- An example of a choke packet used in the early Internet is the SOURCEQUENCH message.
- The modern Internet uses an alternative notification design.

**Explicit Congestion Notification**

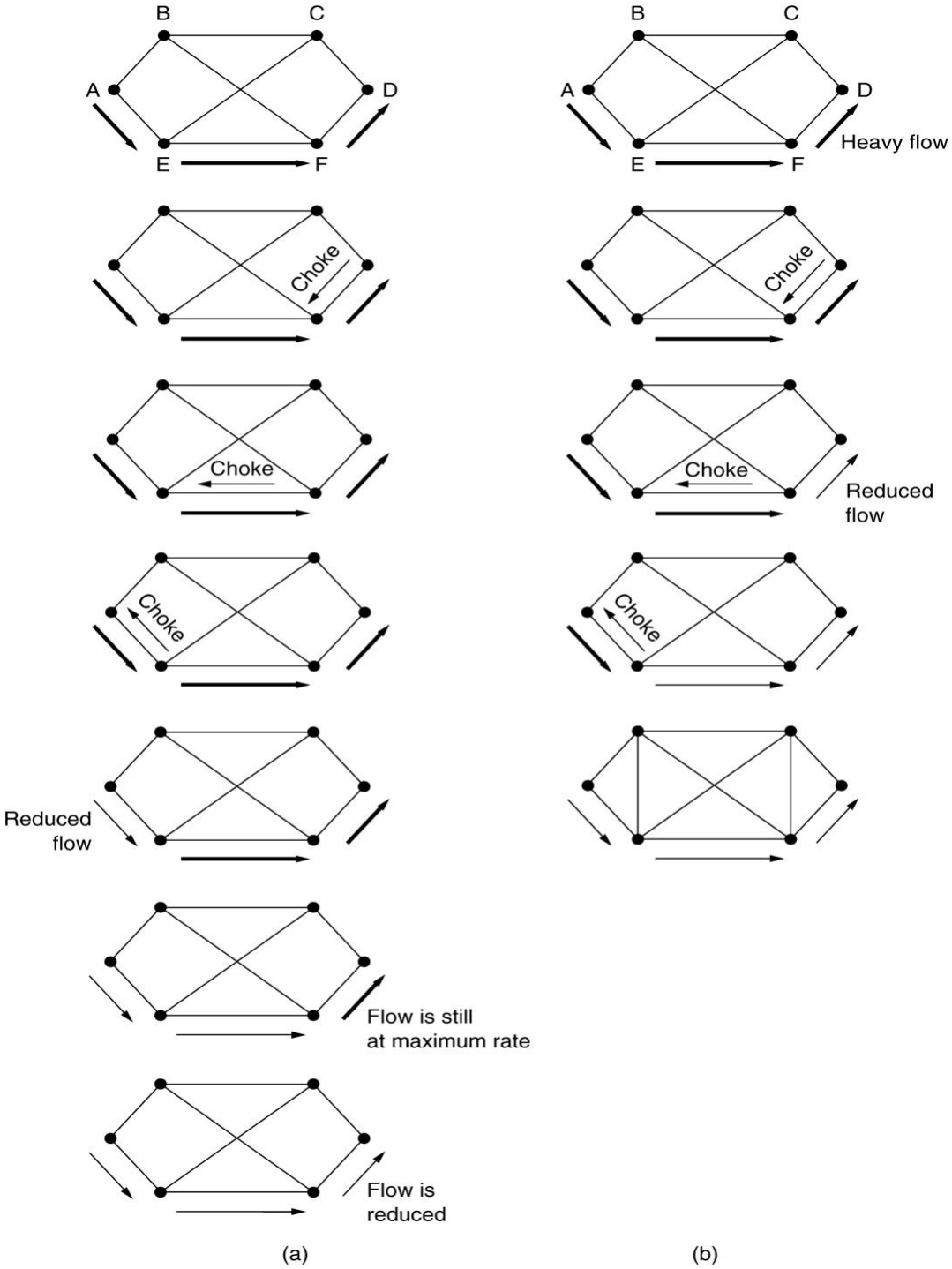
- Instead of generating additional packets (choke packets) to warn congestion, a router can tag any packet it forwards (by setting a bit in the packet's header) to signal that it is experiencing congestion
- When the network delivers a packet, the destination can note that there is congestion & inform the sender when it sends a reply packet. The sender can then throttle its transmissions as before.
- This design is called **ECN (Explicit Congestion Notification)** & is used in the Internet
- It is a refinement of early congestion signalling protocols, such as, the binary feedback scheme of Ramakrishnan & Jain (1988) that was used in the DECNET architecture
- Two bits in the IP packet header are used to record whether the packet has experienced congestion. Packets are unmarked when they are sent, as illustrated here (Explicit congestion notification):




- If the packets pass through any of the congested routers, that router will mark the packet as having experienced congestion as it is forwarded.
- The destination will then echo any marks back to the sender as an explicit congestion signal in its next reply packet. This is shown with a dashed line in the figure to indicate that it happens above the IP level (e.g., in TCP).
- The sender must then throttle its transmissions, as in the case of choke packets.

**Hop-by-Hop Backpressure**

- At high speeds or over long distances, sending a choke packet to the source hosts doesn't work well because the reaction is so slow.
- Refer to figure (a) (a choke packet that affects only the source) below:
  - An alternative approach is to have the choke packet take effect at every hop it passes through
- As soon as the choke packet reaches F, F is required to reduce the flow to D.
- Doing so will require F to devote more buffers to the flow, since the source is still sending away at full blast




- This is done till the choke packet reaches the source

Net effect is to provide quick relief at the point of congestion at the price of using up more buffers upstream

## Load Shedding

- When none of the above methods make the congestion disappear, routers can choose **load shedding**

- **Load shedding** is a fancy way of saying that **when routers are being flooded by packets that they can't handle, they just throw them away**
- The term comes from the world of electrical power generation, where it refers to the practice of utilities intentionally blacking out certain areas to save the entire grid from collapsing on hot summer days when the demand for electricity greatly exceeds the supply
- The key question for a router drowning in packets is which packets to drop
- The preferred choice may depend on the type of applications that use the network
- For a file transfer, an old packet is worth more than a new one. This is because dropping packet 6 & keeping packets 7 through 10, will only force the receiver to do more work to buffer data that it cannot yet use.
- In contrast, for real-time media, a new packet is worth more than an old one. This is because packets become useless if they are delayed & miss the time at which they must be played out to the user.
- The former policy (old is better than new) is often called ***wine*** & the latter (new is better than old) is often called ***milk*** because most people would rather drink new milk & old wine than the alternative
- More intelligent load shedding requires cooperation from the senders. An example is packets that carry routing information. These packets are more important than regular data packets because they establish routes; if they are lost, the network may lose connectivity.
- Another example is that algorithms for compressing video, like MPEG, periodically transmit an entire frame & then send subsequent frames as differences from the last full frame. In this case, dropping a packet that is part of a difference is preferable to dropping one that is part of a full frame because future packets depend on the full frame.
- To implement an intelligent discard policy, applications must mark their packets to indicate to the network how important they are. Then, when packets have to be discarded, routers can first drop packets from the least important class, then the next most important class, & so on.
- Of course, unless there is some significant incentive to avoid marking every packet as VERY IMPORTANT—NEVER, EVER DISCARD, nobody will do it
- The network might let senders send faster than the service they purchased allows if they mark excess packets as low priority. Such a strategy is actually not a bad idea because it makes more efficient use of

idle resources, allowing hosts to use them as long as nobody else is interested, but without establishing a right to them when times get tough.

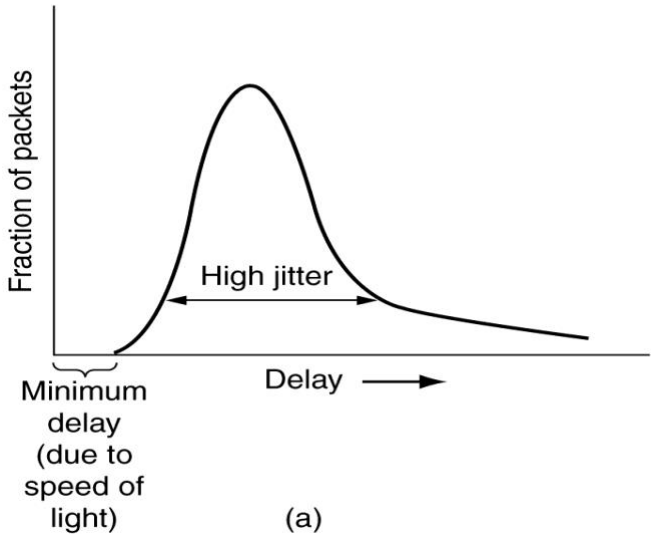
**Random Early Detection**

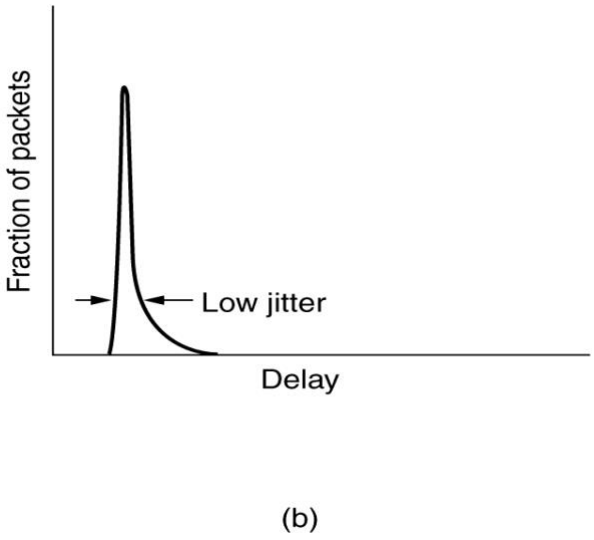
- Dealing with congestion when it first starts is more effective than letting it mixed with the works & then trying to deal with it
- This leads to an interesting twist on load shedding, which is to discard packets before all the buffer space is really exhausted
- The motivation for this idea is that most Internet hosts don't yet get congestion signals from routers in the form of ECN. Instead, the only reliable indication of congestion that hosts get from the network is packet loss.
- After all, it is difficult to build a router that doesn't drop packets when it is overloaded. Transport protocols such as TCP are thus hardwired to react to loss as congestion, slowing down the source in response.
- The reasoning behind this logic is that TCP was designed for wired networks & wired networks are very reliable, so lost packets are mostly due to buffer overruns rather than transmission errors.
- Wireless links must recover transmission errors at the link layer (so they are not seen at the network layer) to work well with TCP. This situation can be exploited to help reduce congestion.
- By having routers drop packets early, before the situation has become hopeless, there is time for the source to take action before it is too late. A popular algorithm for doing this is called **RED (Random Early Detection).**
- To determine when to start discarding, routers maintain a running average of their queue lengths. When the average queue length on some link exceeds a threshold, the link is said to be congested & a small fraction of the packets are dropped at random. Picking packets at random makes it more likely that the fastest senders will see a packet drop; this is the best option since the router can't tell which source is causing the most trouble in a datagram network. The affected sender will notice the loss when there is no acknowledgement, & then the transport protocol will slow down. The lost packet is thus delivering the same message as a choke packet, but implicitly, without the router sending any explicit signal.
- RED routers improve performance compared to routers that drop packets only when their buffers are full, though they may require tuning to work well. For example, the ideal number of packets to drop depends

on how many senders need to be notified of congestion. However, ECN is the preferred option if it is available. It works in exactly the same manner, but delivers a congestion signal explicitly rather than as a loss; **RED is used when hosts cannot receive explicit signals**.

## Jitter Control

- For applications such as audio & video streaming, it doesn't matter much if the packets take 20 msec or 30 msec to be delivered, as long as the transit time is constant.
- The variation (i.e., standard deviation) in the packet arrival times is called **jitter.**
- High jitter, for example, having some packets taking 20 msec & others taking 30 msec to arrive will give an uneven quality to the sound or movie.
- The jitter can be controlled by computing the expected transit time for each hop along the path.
- When a packet arrives at a router, the router checks to see how much the packet is behind or ahead of its schedule. This information is stored in the packet & updated at each hop. If the packet is ahead of schedule, it is held just long enough to get it back on schedule.
- If it is behind schedule, the router tries to send it out quickly. Both cases reduce the amount of jitter
- In applications such as video on demand, jitter can be eliminated by buffering at the receiver & then fetching data for display from the buffer instead from the network in real time.
- In real time applications like Internet telephony & videoconferencing, the delay inherent in buffering is not acceptable.



# QUALITY OF SERVICE

- QoS is defined as something a flow seeks to attain. A stream of packets from a source to a destination is called a **flow**.
- Ina connection-oriented network, all the packets belonging to a flow follow the same route; in a connectionless network, they may follow different routes.
- The needs of each flow can be characterized by 4 primary parameters: reliability, delay, jitter, & bandwidth. Together these determine the QoS (Quality of Service) the flow requires.
- QoS defines a set of attributes related to the performance of the connection. For each connection, the user can request a particular attribute.
- Traditionally, **4 types of flow characteristics** are attributed to a flow, which is given below:
  - **Reliability**
    - A characteristic that a flow need
    - Lack of reliability means losing a packet or acknowledgement, which entails retransmission
    - But, the sensitivity of application programmes to reliability is not the same
  - **Delay**
    - Source-to-destination delay
    - Application can tolerate delay in different degrees.
    - Telephony, audio conferencing, video conferencing & remote log-in need minimum delay, while delay in file transfer or mail is less important.
  - **Jitter**
    - Variation in delay for packets belonging to the same flow
    - If 4 packets depart at time 0, 1, 2, & 3, & arrive at 20, 21, 22, & 23, all have the same delay, 20 units of time.
    - High jitter means the difference between delays is large, low jitter means the variation is small.
  - **Bandwidth**
    - Different application needs different bandwidths
    - In video conferencing, one needs millions of bits per second to refresh a colour screen while the total number of bits in an e-mail may not reach even a million

**QoS Attributes**

- QoS attributes can be classified into 2 major categories as given below:
    1. User-Related Attributes
        - Related to the end user
        - Defines how fast a user wants to send/receive data
        - These attributes are negotiated & defined at the time of contract between the user & the network service provider

| **Attribute** | **Description** |
|---|---|
| Sustained Cell Rate (SCR) | This is the average cell rate over a period of time, which could be more or less the actual transmission rates, as long as the average is maintained. |
| Peak Cell Rate (PCR) | Maximum transmission rate at a point of time. |
| Minimum Cell Rate (MCR) | Minimum Cell Rate that network guarantees a user. |
| Cell Variation Delay Tolerance (CVDT) | A unit to measure changes in cell transmission time (i.e., what is the maximum & minimum delay between the delivery of any two cells). |

    2. Network-Related Attributes
        - Define the characteristics of a network

| **Attribute** | **Description** |
|---|---|
| Cell Lose Ratio (CLR) | Fraction of the cells lost/delivered too late during transmission |
| Cell Transfer Delay (CTD) | Average time required for a cell to travel from the source to the destination |
| Cell Delay Variation (CDV) | Difference between the maximum & minimum values of CTD |
| Cell Error Ratio (CER) | Fraction of cells that contain errors |

## Requirements

- Several common applications & the harshness of their requirements are listed here (how strict the QoS requirements are):

| Application | Reliability | Delay | Jitter | Bandwidth |
|---|---|---|---|---|
| E-mail | High | Low | Low | Low |
| File transfer | High | Low | Low | Medium |
| Web access | High | Medium | Low | Medium |
| Remote login | High | Medium | Medium | Low |
| Audio on demand | Low | Low | High | Medium |
| Video on demand | Low | Low | High | High |
| Telephony | Low | High | High | Low |
| Videoconferencing | Low | High | High | High |

- The first four applications have stringent requirements on reliability. No bits may be delivered incorrectly. The four final (audio/video) applications can tolerate errors, so no checksums are computed or verified.
- File transfer applications, including e-mail & video, are not delay sensitive. If all packets are delayed uniformly by a few seconds, no harm is done. Interactive applications, such as Web surfing & remote login, are more delay sensitive. Real-time applications, such as telephony & videoconferencing have strict delay requirements. If all the words in a telephone call are each delayed by exactly 2.000 seconds, the users will find the connection unacceptable. On the other hand, playing audio or video files from a server doesn't require low delay.
- Finally, the applications differ in their bandwidth needs, with e-mail & remote login not needing much, but video in all forms needing a great deal.
- **ATM (Asynchronous Transfer Mode) networks** classify flows in four broad categories with respect to their QoS demands as follows:

1. Constant bit rate (e.g., telephony).
2. Real-time variable bit rate (e.g., compressed videoconferencing).
3. Non-real-time variable bit rate (e.g., watching a movie over the Internet).
4. Available bit rate (e.g., file transfer).

- These categories are also useful for other purposes & other networks. Constant bit rate is an attempt to simulate a wire by providing a uniform bandwidth & a uniform delay. Variable bit rate occurs when video is compressed, some frames compressing more than others.

## Techniques for achieving good Quality of Service

Some of the techniques, system designers use to achieve QoS:

**Overprovisioning**

- Provide greater router capacity, buffer space & bandwidth
- An expensive technique as the resources is costly
- E.g.: Telephone System.

**Buffering**

- Flows can be **buffered** on the receiving side before being delivered.
- Buffering them doesn't affect the reliability or bandwidth, & increases the delay, but it smooths out the jitter.
- For audio & video on demand, jitter is the main problem, so this technique helps a lot.

**Traffic Shaping**

- Regulating the average rate of data transmission
- Smooths the traffic on server side other than client side. When a connection is set up, the user machine & subnet agree on a certain traffic pattern for that circuit called as *Service Level Agreement*. It reduces congestion & thus helps the carrier to deliver the packets in the agreed pattern.

# Module - 4 (Network Layer on the Internet)

IP protocol, IP addresses, Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP). Open Shortest Path First (OSPF) Protocol, Border Gateway Protocol (BGP), Internet multicasting, IPv6, ICMPv6.

# Network Layer on the Internet

Principles that helped in the design of Network Layer on the Internet:

1. **Make sure it works** (Don't finalize the design or standard until multiple prototypes have successfully communicated with each other)

2. **Keep it simple** (When in doubt, use the simplest solution)

3. **Make clear choices** (If there are several ways of doing the same thing, choose one. Having 2 or more ways to do the same thing is looking for trouble.)

4. **Exploit modularity** (This principle leads directly to the idea of having protocol stacks, each of whose layers is independent of all the other ones. In this way, if circumstances require one module or layer to be changed, the other ones won't be affected.)

5. **Expect heterogeneity** (Different types of hardware, transmission facilities, & applications will occur on any large network. To handle them, the network design must be simple, general, & flexible)

6. **Avoid static options and parameters** (If parameters are unavoidable (e.g., maximum packet size), it is best to have the sender & receiver negotiate a value rather than defining fixed choices.)

7. **Look for a good design; it need not be perfect** (Often, the designers have a good design, but it cannot handle some weird special case.)

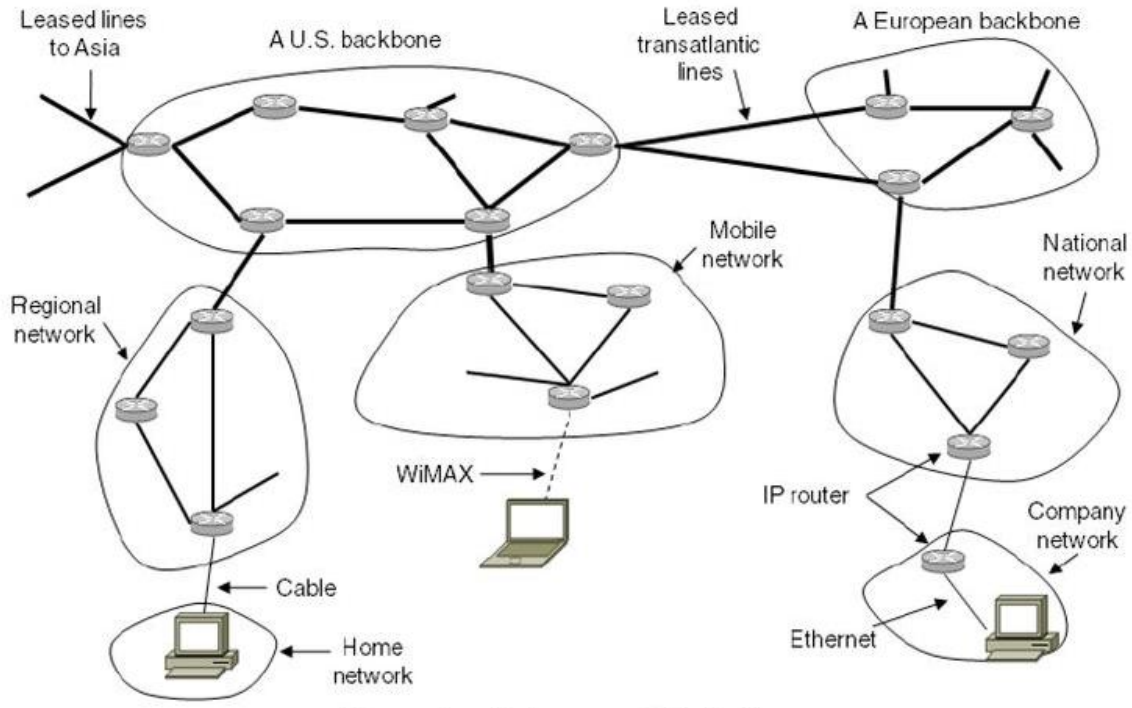
8. **Be strict when sending and tolerant when receiving** (Send only packets that rigorously comply with the standards but expect incoming packets that may not be fully conformant & try to deal with them.)

9. **Think about scalability** (If the system is to handle millions of hosts & billions of users effectively, no centralized databases of any kind are

tolerable, & load must be spread as evenly as possible over the available resources.)

10. **Consider performance & cost** (If a network has poor performance or outrageous costs, nobody will use it.)

- Internet can be viewed as a collection of networks or ASes (Autonomous Systems) that are interconnected (in the network layer).
- There is no real structure, but several major backbones exist. These are constructed from high-bandwidth lines & fast routers.
- The biggest of these backbones, to which everyone else connects to reach the rest of the Internet, are called Tier 1 networks
- Attached to the backbones are ISPs (Internet Service Providers) that provide Internet access to homes & businesses, data centres & colocation facilities full of server machines, & regional (mid-level) networks. The data centres serve much of the content that is sent over the Internet.
- Attached to the regional networks are more ISPs, LANs at many universities & companies, & other edge networks. A sketch of this quasihierarchical organization is given in the below figure.



# Internet Protocol (IP)

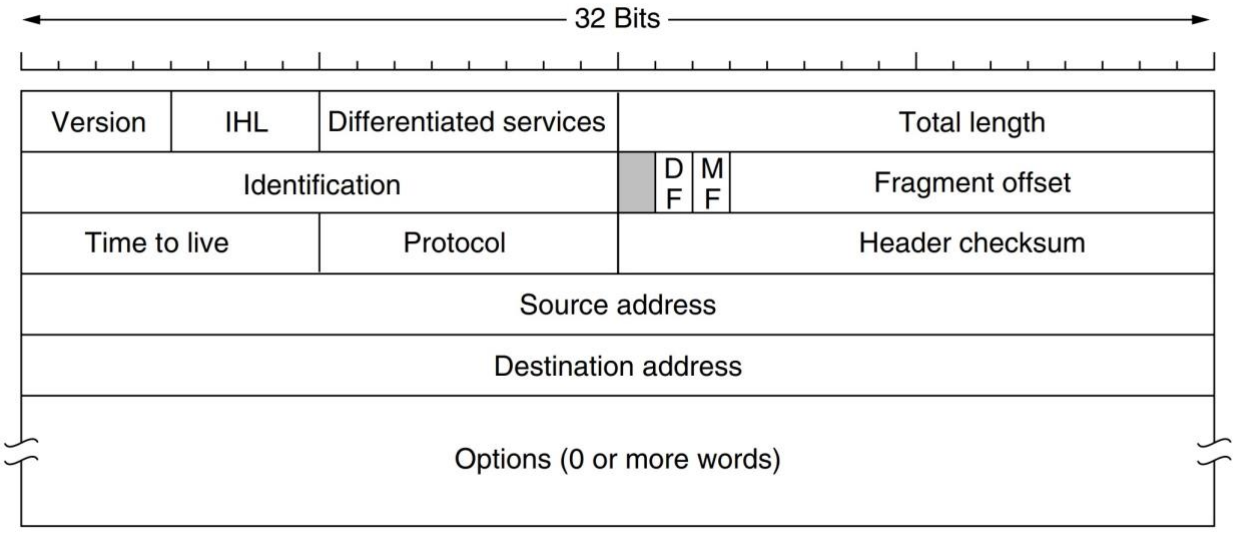
- The glue that holds the whole Internet together is the network layer protocol, **IP (Internet Protocol)**.
- Unlike most older network layer protocols, IP was designed from the beginning with internetworking in mind.
- A good way to think of the network layer is this: its job is to provide a best-effort (i.e., not guaranteed) way to transport packets from source to destination, without regard to whether these machines are on the same network or whether there are other networks in between them.

Communication on the Internet works as follows:

- The transport layer takes data streams & breaks them up so that they may be sent as IP packets
- In theory, packets can be up to 64 KB each, but in practice they are usually not more than 1500 bytes (so they fit in one Ethernet frame)
- IP routers forward each packet through the Internet, along a path from one router to the next, until the destination is reached
- At the destination, the network layer hands the data to the transport layer, which gives it to the receiving process
- When all the pieces finally get to the destination machine, they are reassembled by the network layer into the original datagram. This datagram is then handed to the transport layer.

- In the example given in the figure above, a packet originating at a host on the home network must traverse 4 networks & many IP routers before even getting to the company network on which the destination host is located.
- This is not unusual in practice, and there are many longer paths. There is also much redundant connectivity on the Internet, with backbones and ISPs connecting to each other in multiple locations.
- This means that there are many possible paths between two hosts. It is the job of the IP routing protocols to decide which paths to use.

# The IP Version 4 Protocol




Let's learn the format of IP datagrams

- An IPv4 datagram consists of **a header part** & **a body or payload part**
- The **header has a 20-byte fixed part & a variable-length optional part**. The **header format is shown in the figure mentioned** above.
- The bits are transmitted from left to right & top to bottom, with the high-order bit of the Version field going 1st
- The ***Version field*** **keeps track of which version of the protocol the datagram belongs to**
    - Version 4 dominates the Internet today
    - By including the version at the start of each datagram, it becomes possible to have a transition between versions over a long period of time
    - In fact, **IPv6,** the next version of IP**, was defined more than a decade ago, yet is only just beginning to be deployed**. Its use will eventually be forced when each of China's almost 231 people has a desktop PC, a laptop, & an IP phone. As an aside on numbering, IPv5 was an experimental real-time stream protocol that was never widely used.
- The **header length** is not constant, so, a field in the header, **IHL (Internet Header length), tells how long the header is**, in 32-bit words.
    - The minimum value is 5, which applies when no options are present
    - The maximum value of this 4-bit field is 15, which limits the header to 60 bytes, & thus the Options field to 40 bytes. For some options,

such as one that records the route a packet has taken, 40 bytes is far too small, making those options useless.

- The **Differentiated services field** was originally called the **Type of service** field. It **distinguishes between different classes of service**.
    - Various combinations of reliability and speed are possible
    - For digitized voice, fast delivery beats accurate delivery
    - For file transfer, error-free transmission is more important than fast transmission
    - The Type of service field provided **3 bits to signal priority** and **3 bits to signal whether a host cared more about delay, throughput, or reliability**
    - **The top 6 bits are used to mark the packet with its service class; the bottom 2 bits are used to carry explicit congestion notification information**, such as whether the packet has experienced congestion.
- The **Total length** includes everything in the datagram—both header & data. The maximum length is 65,535 bytes. Now, this upper limit is tolerable, but with future networks, larger datagrams may be needed.
- The **Identification field allows the destination host to determine which packet a newly arrived fragment belongs to.** All the fragments of a packet contain the same Identification value.
- Next comes **an unused bit**, that was proposed by Bellovin (2003) to detect malicious traffic. This would greatly **simplify security**, as packets with the ‘‘evil’’ bit set would be known to have been sent by attackers and could just be discarded. Unfortunately, network security is not this simple.
- Then come two **1-bit fields** related to fragmentation:
- **DF** stands for **Don’t Fragment**. It is an **order to the routers not to fragment** the packet.
    - It was intended to support hosts incapable of putting the pieces back together again
    - Now, it is used as part of the process to discover the path MTU (Maximum Transmission Unit), which is the largest packet that can travel along a path without being fragmented
    - By marking the datagram with the DF bit, the sender knows it will either arrive in one piece, or an error message will be returned to the sender
- **MF** stands for **More Fragments**. All fragments except the last one has this bit set. It is needed to **know when all fragments of a datagram have arrived**.

- The **Fragment offset** tells where this fragment belongs in the current packet.
    - All fragments except the last one in a datagram must be a multiple of 8 bytes, the elementary fragment unit
    - Since 13 bits are provided, there is a maximum of 8192 fragments per datagram, supporting a maximum packet length up to the limit of the Total length field.
- Working together, the **Identification, MF,** and **Fragment offset fields** are used to **implement fragmentation.**
- The **TTL (Time to live) field** is a counter used to limit packet lifetimes.
    - It was supposed to count time in seconds, allowing a maximum lifetime of 255 sec. It must be decremented on each hop and is supposed to be decremented multiple times when a packet is queued for a long time in a router.
    - In practice, it just counts hops. When it hits zero, the packet is discarded, and a warning packet is sent back to the source host. This feature prevents packets from wandering around forever, something that otherwise might happen if the routing tables ever become corrupted. When the network layer has assembled a complete packet, it needs to know what to do with it.
- The **Protocol field** tells it which transport process to give the packet to.
    - TCP is one possibility, but so are UDP and some others.
    - The numbering of protocols is global across the entire Internet.
    - Protocols and other assigned numbers were formerly listed in RFC 1700, but nowadays they are contained in an online database located at www.iana.org.
- Since the header carries vital information, such as addresses, it rates its own checksum for protection, the **Header checksum**.
    - The algorithm is to add up all the 16-bit halfwords of the header as they arrive, using one's complement arithmetic, and then take the one's complement of the result.
    - For purposes of this algorithm, the Header checksum is assumed to be zero upon arrival. Such a checksum is **useful for detecting errors** while the packet travels through the network. Note that it must be recomputed at each hop because at least one field always changes (the Time to live field), but tricks can be used to speed up the computation.
- The **Source address** and **Destination address** indicate the IP address of the source and destination network interfaces.

- The **Options field** was designed
    - to provide an escape to allow subsequent versions of the protocol
    - to include information that is not present in the original design,
    - to permit experimenters to try out new ideas, &
    - to avoid allocating header bits to information that is rarely needed.
  - The options are of variable length. Each begins with a 1-byte code identifying the option. Some options are followed by a 1-byte option length field, and then one or more data bytes. The Options field is padded out to a multiple of 4 bytes.
  - 5 options are listed in the table below:

| Option | Description |
|---|---|
| Security | Specifies how secret the datagram is |
| Strict source routing | Gives the complete path to be followed |
| Loose source routing | Gives a list of routers not to be missed |
| Record route | Makes each router append its IP address |
| Timestamp | Makes each router append its address and timestamp |

- The *Security option* tells how secret the information is. A military router might use this field to specify not to route packets through certain countries the military considers to be ‘‘bad guys.’’ In practice, all routers ignore it, so its only practical function is to help spies find the good stuff more easily.
- The *Strict source routing option* gives the complete path from source to destination as a sequence of IP addresses. The datagram is required to follow that exact route. It is most useful for system managers who need to send emergency packets when the routing tables have been corrupted, or for making timing measurements.
- The *Loose source routing option* requires the packet to traverse the list of routers specified, in the order specified, but it is allowed to pass through other routers on the way. Normally, this option will provide only a few routers, to force a particular path. For example, to force a packet from London to Sydney to go west instead of east, this option might specify routers in New York,

Los Angeles, and Honolulu. This option is most useful when political or economic considerations dictate passing through or avoiding certain countries.

- The ***Record route option*** tells each router along the path to append its IP address to the Options field. This allows system managers to track down bugs in the routing algorithms (‘‘Why are packets from Houston to Dallas visiting Tokyo first?’’). When the ARPANET was first set up, no packet ever passed through more than nine routers, so 40 bytes of options was plenty. As mentioned above, now it is too small.
- Finally, the ***Timestamp option*** is like the Record route option, except that in addition to recording its 32-bit IP address, each router also records a 32-bit timestamp. This option, too, is mostly useful for network measurement.

- Today, IP options have fallen out of favour. Many routers ignore them or do not process them efficiently, shunting them to the side as an uncommon case. That is, they are only partly supported, and they are rarely used.

# IP Addresses

- A defining feature of IPv4 is its 32-bit addresses
- Every host and router on the Internet have an IP address that can be used in the Source address and Destination address fields of IP packets
- An IP address doesn’t refer to a host. It refers to a network interface, so if a host is on two networks, it must have two IP addresses. But, in practice, most hosts are on one network and thus have one IP address. In contrast, routers have multiple interfaces and thus multiple IP addresses.

# Prefixes (not in the syllabus)

- IP addresses are hierarchical, unlike Ethernet addresses
- Each 32-bit address is comprised of a variable-length network portion in the top bits and a host portion in the bottom bits
- For all the hosts on a single network (like an Ethernet LAN), the network portion has the same value. This means that a network corresponds to a contiguous block of IP address space. This block is called a **prefix**.
- IP addresses are written in dotted decimal notation. In this format, each of the 4 bytes is written in decimal, from 0 to 255.

- For example, the 32-bit hexadecimal address 80D00297 is written as 128.208.2.151.
- Prefixes are written by giving the lowest IP address in the block and the size of the block. The size is determined by the number of bits in the network portion; the remaining bits in the host portion can vary. This means that the size must be a power of two. By convention, it is written after the prefix IP address as a slash followed by the length in bits of the network portion. In our example, if the prefix contains 28 addresses and so leaves 24 bits for the network portion, it is written as 128.208.0.0/24. Since the prefix length cannot be inferred from the IP address alone, routing protocols must carry the prefixes to routers.
- Sometimes prefixes are simply described by their length, as in a ‘‘/16’’ which is pronounced “slash 16”. The length of the prefix corresponds to a binary mask of 1s in the network portion.
- When written out this way, it is called a subnet mask. It can be ANDed with the IP address to extract only the network portion. For our example, the subnet mask is 255.255.255.0. The below figure shows a prefix and a subnet mask.

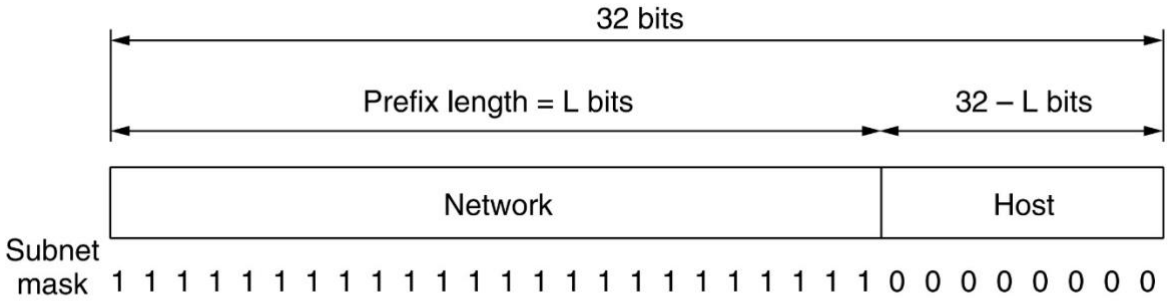



**Figure 5-48.** An IP prefix and a subnet mask.

- Hierarchical addresses have significant advantages and disadvantages. The key **advantage** of prefixes is that **routers can forward packets based on only the network portion of the address, if each of the networks has a unique address block**. The host portion does not matter to the routers because all hosts on the same network will be sent in the same direction. It is only when the packets reach the network for which they are destined that they are forwarded to the correct host. This makes the routing tables much smaller than they would otherwise be. Consider that the number of hosts on the Internet is approaching one billion. That would be a very large table for every router to keep. However, by using a hierarchy, routers need to keep routes for only around 300,000 prefixes.

- While using a hierarchy lets Internet routing scale, it has two **disadvantages**:
    - First, the IP address of a host depends on where it is in the network. An Ethernet address can be used anywhere in the world, but every IP address belongs to a specific network, and routers will only be able to deliver packets destined to that address to the network. Designs such as mobile IP are needed to support hosts that move between networks but want to keep the same IP addresses.
    - The second disadvantage is that unless it is carefully managed, hierarchy is wasteful. If addresses are assigned to networks in (too) large blocks, there will be (many) addresses that are allocated but not in use. This allocation would not matter much if there were plenty of addresses to go around. But it was realized more than two decades ago that the tremendous growth of the Internet was rapidly depleting the free address space. IPv6 is the solution to this shortage, but until it is widely deployed, there will be great pressure to allocate IP addresses so that they are used very efficiently.

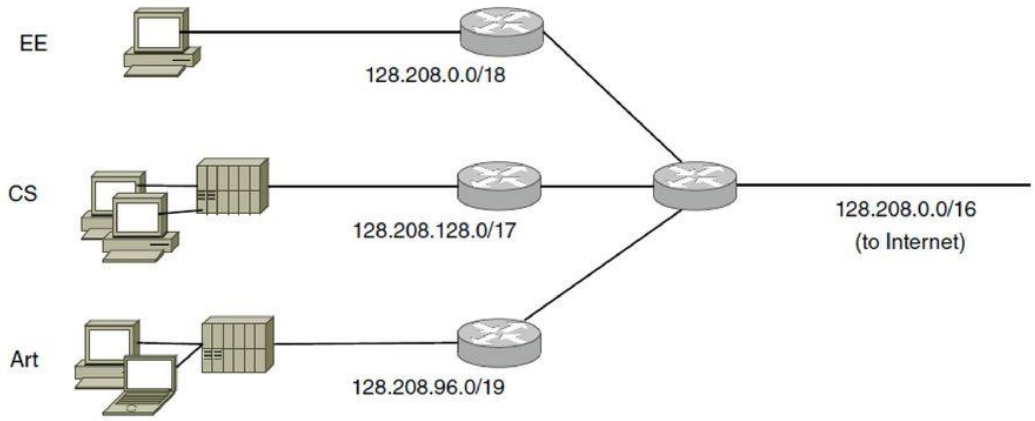
# Subnets

- Network numbers are managed by a non-profit corporation called **ICANN (Internet Corporation for Assigned Names and Numbers)**, to avoid conflicts
- In turn, ICANN has assigned parts of the address space to various regional authorities, which allot IP addresses to ISPs and other companies. This is the process by which a company is allocated a block of IP addresses.
- But this process is only the start of the story, as IP address assignment is ongoing as companies grow.
- Since routing by prefix requires all the hosts in a network to have the same network number, this can cause problems as networks grow.
- For example, consider a university that started out with our example /16 prefix for use by the Computer Science Dept. for the computers on its Ethernet. A year later, the Electrical Engineering Dept. wants to get on the Internet. The Art Dept. soon follows suit. What IP addresses should these departments use? Getting further blocks requires going outside the university and may be expensive or inconvenient. Moreover, the /16 already allocated has enough addresses for over 60,000 hosts. It might be intended to allow for significant growth, but until

that happens, it is wasteful to allocate further blocks of IP addresses to the same university. A different organization is required.

- The **solution** is to allow the block of addresses to be split into several parts for internal use as multiple networks, while still acting like a single network to the outside world. This is called **subnetting** and the networks (such as Ethernet LANs) that result from dividing up a larger network are called **subnets**.
- The term "**subnet**" means set of all routers and communication lines in a network. The below figure shows how subnets can help with our example. The single /16 has been split into pieces.

# IP Addresses (2)




Splitting an IP prefix into separate networks with subnetting.

- This split doesn't need to be even, but each piece must be aligned so that any bits can be used in the lower host portion. In this case, half of the block (a /17) is allocated to the Computer Science Dept, a quarter is allocated to the Electrical Engineering Dept. (a /18), and one eighth (a /19) to the Art Dept. The remaining eighth is unallocated.
- A different way to see how the block was divided is to look at the resulting prefixes when written in binary notation:

  Computer Science: 10000000 11010000 1|xxxxxxx xxxxxxxx
  Electrical Eng. : 10000000 11010000 00|xxxxxx xxxxxxxx

Art : 10000000 11010000 011|xxxxx xxxxxxxx

- Here, the **vertical bar (|)** shows the **boundary between the subnet number & the host portion**. When a packet comes into the main router, how does the router know which subnet to give it to?
- The routers simply need to know the subnet masks for the networks on campus
- When a packet arrives, the router looks at the destination address of the packet and checks which subnet it belongs to. The router can do this by ANDing the destination address with the mask for each subnet and checking to see if the result is the corresponding prefix.
- For example, consider a packet destined for IP address 128.208.2.151. To see if it is for the Computer Science Dept., we AND with 255.255.128.0 to take the first 17 bits (which is 128.208.0.0) and see if they match the prefix address (which is 128.208.128.0). They do not match.
- Checking the first 18 bits for the Electrical Engineering Dept., we get 128.208.0.0 when ANDing with the subnet mask. This does match the prefix address, so the packet is forwarded onto the interface which leads to the Electrical Engineering network.
- The subnet divisions can be changed later, by updating all subnet masks at routers inside the university. Outside the network, the subnetting is not visible, so allocating a new subnet doesn't require contacting ICANN or changing any external databases.

# CIDR—Classless Interdomain Routing

- Also known as **Classless addressing**

In **Classful** addressing, no of Hosts within a network always remains the same depending upon the class of the Network.

Class A network contains $2^{24}$ Hosts,

Class B network contains $2^{16}$ Hosts,

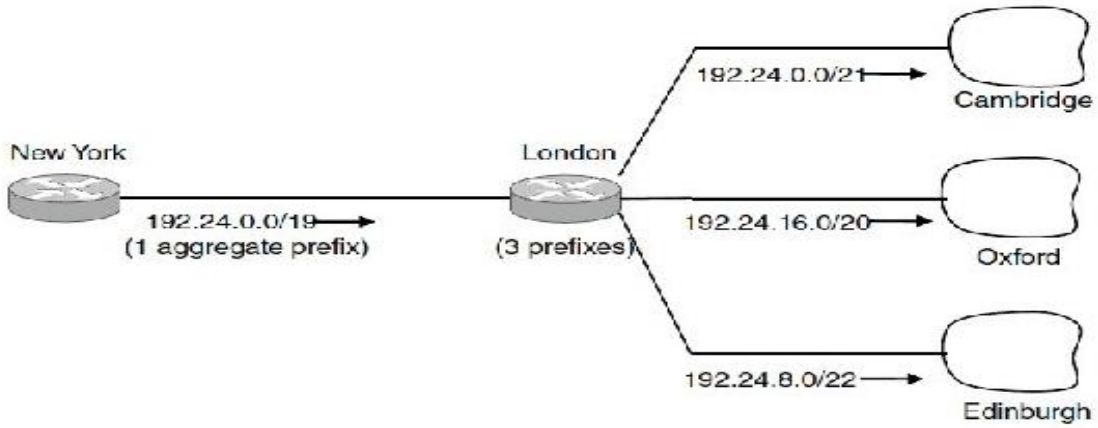
Class C network contains $2^{8}$ Hosts

- Now, let's suppose an organization requires $2^{14}$ hosts, then it must have to purchase a Class B network. In this case, 49152 Hosts will be wasted. This is the major **drawback of Classful Addressing**.
- In order to reduce the wastage of IP addresses a new concept of **Classless Inter-Domain Routing** is introduced

- IANA (Internet Assigned Numbers Authority) uses this technique to provide IP addresses. Whenever any user asks for IP addresses, IANA assigns that many IP addresses to the User.
- Consider an example in which a block of 8192 IP addresses is available, starting at 194.24.0.0.
  - Suppose Cambridge University needs 2048 addresses & is assigned the addresses 194.24.0.0 through 194.24.7.255, along with mask 255.255.248.0. This is a /21 prefix.
  - Next, Oxford University asks for 4096 addresses. Since a block of 4096 addresses must lie on a 4096-byte boundary, Oxford can't be given addresses starting at 194.24.8.0. Instead, it gets 194.24.16.0 through 194.24.31.255, along with subnet mask 255.255.240.0.
  - Finally, the University of Edinburgh asks for 1024 addresses & is assigned addresses 194.24.8.0 through 194.24.11.255 & mask 255.255.252.0. These assignments are summarized here:

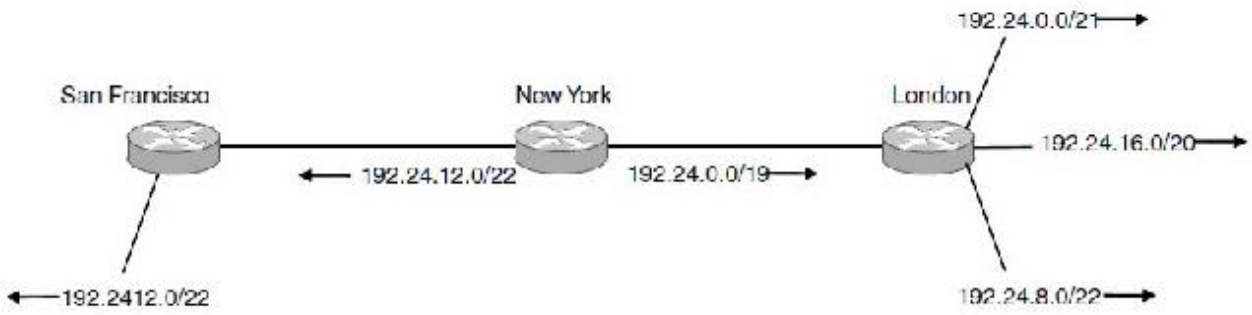
| University | First address | Last address | How many | Prefix |
|---|---|---|---|---|
| Cambridge | 194.24.0.0 | 194.24.7.255 | 2048 | 194.24.0.0/21 |
| Edinburgh | 194.24.8.0 | 194.24.11.255 | 1024 | 194.24.8.0/22 |
| (Available) | 194.24.12.0 | 194.24.15.255 | 1024 | 194.24.12/22 |
| Oxford | 194.24.16.0 | 194.24.31.255 | 4096 | 194.24.16.0/20 |

- All routers in the default-free zone are now told about IP addresses in the three networks. Routers close to the universities may need to send on a different outgoing line for each of the prefixes, so they need an entry for each of the prefixes in their routing tables. An example of the router in London is shown below:




Aggregation of IP prefixes

- Let’s look at these three universities from the point of view of a distant router in New York.
  - All of the IP addresses in the three prefixes should be sent from New York (or the U.S. in general) to London. The routing process in London notices this & combines the three prefixes into a single aggregate entry for the prefix 194.24.0.0/19 that it passes to the New York router.
  - This prefix contains 8K addresses & covers the three universities & the otherwise unallocated 1024 addresses.
  - By using aggregation, three prefixes have been reduced to one, reducing the prefixes that the New York router must be told about & the routing table entries in the New York router.
- When aggregation is turned on, it is an automatic process. It depends on which prefixes are located where in the Internet, not on the actions of an administrator assigning addresses to networks. Aggregation is heavily used throughout the Internet & can reduce the size of router tables to around 200,000 prefixes.
- Prefixes are allowed to overlap.
  - The rule is that packets are sent in the direction of the most specific route, or the longest matching prefix that has the fewest IP addresses.
  - **Longest matching prefix routing** provides a useful degree of flexibility, as seen in the behaviour of the router at New York in below figure.
  - This router uses a single aggregate prefix to send traffic for the three universities to London. But, the previously available block of addresses within this prefix has now been allocated to a network in San Francisco.
  - One possibility for the New York router is to keep four prefixes, sending packets for three of them to London & packets for the fourth to San Francisco.
  - Instead, longest matching prefix routing can handle this forwarding with two prefixes that are shown.
  - One overall prefix is used to direct traffic for the entire block to London. One more specific prefix is also used to direct a portion of the larger prefix to San Francisco.
  - With the longest matching prefix rule, IP addresses within the San Francisco network will be sent on the outgoing line to San Francisco, & all other IP addresses in the larger prefix will be sent to London.




Longest matching prefix routing at the New York router.

- **Conceptually, CIDR works as follows:**
  - **When a packet comes in, the routing table is scanned to determine if the destination lies within the prefix.**
  - **It is possible that multiple entries with different prefix lengths will match, in which case the entry with the longest prefix is used. Thus, if there is a match for a /20 mask & a /24 mask, the /24 entry is used to look up the outgoing line for the packet.**
  - **But this process would be tedious if the table were really scanned entry by entry.**
  - **Instead, complex algorithms have been devised to speed up the address matching process. Commercial routers use custom VLSI chips with these algorithms embedded in hardware.**

# Network Address Translation (NAT)

- To access the Internet, one public IP address is needed, but we can use a private IP address in our private network.
- **The idea of NAT is to allow multiple devices to access the Internet through a single public address.**
- To achieve this, the translation of a private IP address to a public IP address is required.
- **Network Address Translation (NAT)** is a process in which one or more local IP address is translated into one or more Global IP address & vice versa in order to provide Internet access to the local hosts.

- Also, it does the translation of port numbers, i.e., masks the port number of the host with another port number, in the packet that will be routed to the destination.
- It then makes the corresponding entries of IP address & port number in the NAT table. NAT generally operates on a router or firewall.

**Network Address Translation (NAT) working:**

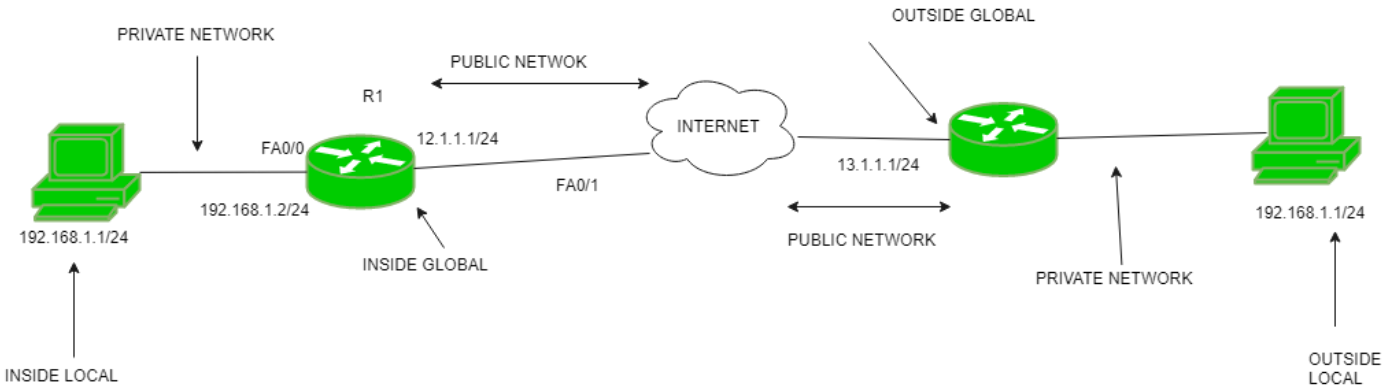
- Generally, the border router is configured for NAT, i.e., the router which has one interface in the local (inside) network & one interface in the global (outside) network.
- When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address.
- When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.
- If NAT runs out of addresses, i.e., no address is left in the pool configured, then the packets will be dropped & an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

**Why mask port numbers?**

- Suppose, in a network, two hosts A & B are connected.
- Both of them request for the same destination, on the same port number, say 1000, on the host side, at the same time.
- If NAT does only translation of IP addresses, then when their packets will arrive at the NAT, both of their IP addresses would be masked by the public IP address of the network & sent to the destination.
- Destination will send replies to the public IP address of the router.
- Thus, on receiving a reply, it will be unclear to NAT as to which reply belongs to which host (because source port numbers for both A & B are the same).
- Hence, to avoid such a problem, NAT masks the source port number as well & makes an entry in the NAT table.

# NAT inside and outside addresses

Inside refers to the addresses which must be translated.
Outside refers to the addresses which are not in control of an organization.
These are the network Addresses in which the translation of the addresses will be done.




- **Inside local address:**
  - An IP address that is assigned to a host on the Inside (local) network.
  - The address is probably not an IP address assigned by the service provider i.e., these are private IP addresses.
  - This is the inside host seen from the inside network.
- **Inside global address:**
  - IP address that represents one or more inside local IP addresses to the outside world.
  - This is the inside host as seen from the outside network.
- **Outside local address:**
  - This is the actual IP address of the destination host in the local network after translation.
- **Outside global address:**
  - This is the outside host as seen from the outside network.
  - It is the IP address of the outside destination host before translation.

# Network Address Translation (NAT) Types:

There are **3 ways to configure NAT**:

1. **Static NAT**
   - A single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e., one-to-one mapping between local & global addresses.
   - This is generally used for Web hosting.
   - These are not used in organizations as there are many devices that will need Internet access & to provide Internet access, a public IP address is needed.

Suppose, if there are 3000 devices that need access to the Internet, the organization has to buy 3000 public addresses that will be very costly.

2. **Dynamic NAT**

- An unregistered IP address is translated into a registered (Public) IP address from a pool of public IP addresses.
- If the IP address of the pool is not free, then the packet will be dropped as only a fixed number of private IP addresses can be translated to public addresses.

Suppose, if there is a pool of 2 public IP addresses then only 2 private IP addresses can be translated at a given time. If 3rd private IP address wants to access the Internet, then the packet will be dropped, therefore, many private IP addresses are mapped to a pool of public IP addresses. NAT is used when the number of users who want to access the Internet is fixed. This is also very costly as the organization has to buy many global IP addresses to make a pool.

3. **Port Address Translation (PAT)**

    - Also known as NAT overload
    - Many local (private) IP addresses can be translated to a single registered IP address.
    - Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address.
    - This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

## Advantages of NAT

- NAT conserves legally registered IP addresses.
- It provides privacy as the device's IP address, sending & receiving the traffic, will be hidden.
- Eliminates address renumbering when a network evolves.

**Disadvantage of NAT**

- Translation results in switching path delays.
- Certain applications won't function while NAT is enabled.
- Complicates tunnelling protocols such as IPsec.
- Also, the router being a network layer device, shouldn't tamper with port numbers (transport layer) but it has to do so because of NAT.

# Internet Control Message Protocol (ICMP)

- Operation of Internet is monitored closely by routers.
- When something unexpected occurs at a router during packet processing, the event is reported to the sender by **ICMP (Internet Control Message Protocol)**.
- ICMP is also used to test the Internet. About a dozen types of ICMP messages are defined. Each ICMP message type is carried & encapsulated in an IP packet. The most important ones are listed here:

| Message type | Description |
|---|---|
| Destination unreachable | Packet could not be delivered |
| Time exceeded | Time to live field hit 0 |
| Parameter problem | Invalid header field |
| Source quench | Choke packet |
| Redirect | Teach a router about geography |
| Echo and Echo reply | Check if a machine is alive |
| Timestamp request/reply | Same as Echo, but with timestamp |
| Router advertisement/solicitation | Find a nearby router |

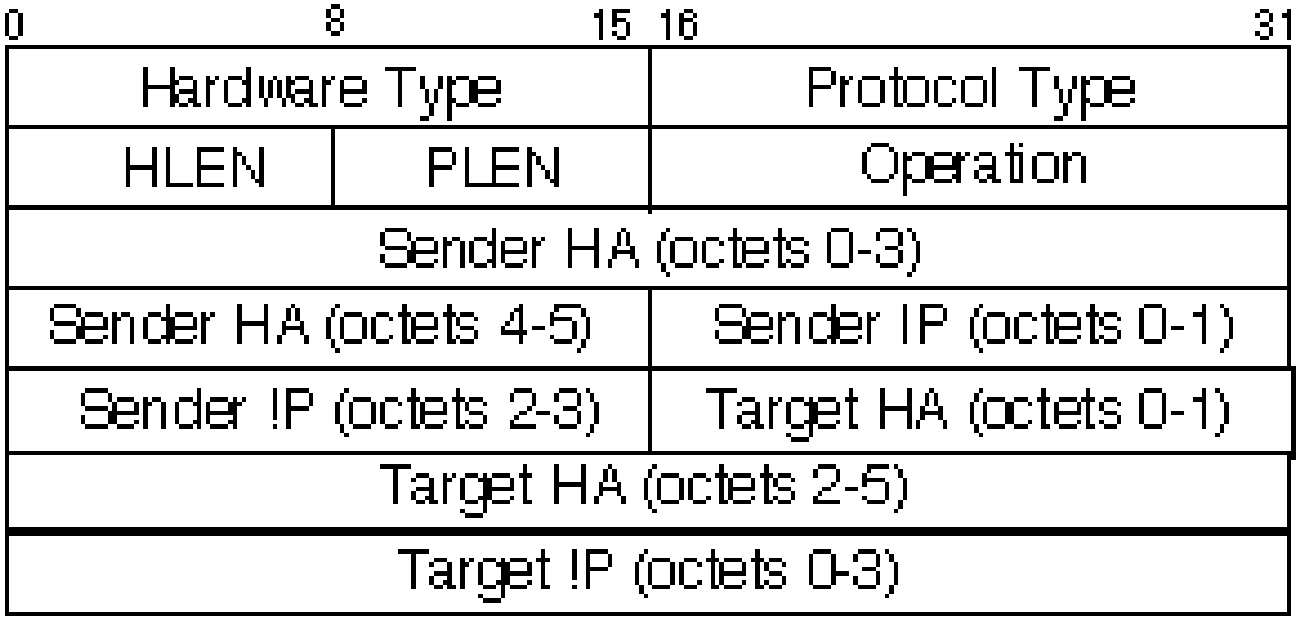
- The **DESTINATION UNREACHABLE** message is used when the router can't locate the destination or when a packet with DF bit can't be delivered because a "small-packet" network stands in the way.
- The **TIME EXCEEDED** message is sent when a packet is dropped because its TtL (Time to live) counter has reached zero. This event is a symptom that packets are looping, or that the counter values are being set too low.
- The **PARAMETER PROBLEM** message indicates that an illegal value has been detected in a header field. This problem indicates a bug in the sending host's IP software or possibly in the software of a router transited.
- The **SOURCE QUENCH** message was long ago used to throttle hosts that were sending too many packets.
    - When a host received this message, it was expected to slow down.
    - It is rarely used anymore because when congestion occurs, these packets tend to add more fuel to the fire & it is unclear how to respond to them.

- The **REDIRECT** message is used when a router notices that a packet seems to be routed incorrectly. It is used by the router to tell the sending host to update to a better route.
- The **ECHO & ECHO REPLY** messages are sent by hosts to see if a given destination is reachable & currently alive.
    - Upon receiving the ECHO message, the destination is expected to send back an ECHO REPLY message. These messages are used in the ping utility that checks if a host is up & on the Internet.
- The **TIMESTAMP REQUEST & TIMESTAMP REPLY** messages are similar, except that the arrival time of the message & the departure time of the reply are recorded in the reply. This facility can be used to measure network performance.
- The **ROUTER ADVERTISEMENT & ROUTER SOLICITATION** messages are used to let hosts find nearby routers. A host needs to learn the IP address of at least one router to be able to send packets off the local network.

# Address Resolution Protocol (ARP)

- The **Address Resolution Protocol (ARP)** is a protocol used by the Internet Protocol (IP) [RFC826], specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol.
- The protocol operates below the network layer as a part of the interface between the OSI network & OSI link layer. It is used when IPv4 is used over Ethernet.
- The term address resolution refers to the process of finding an address of a computer in a network.
- The address is “resolved” using a protocol in which a piece of information is sent by a client process executing on the local computer to a server process executing on a remote computer.
- The information received by the server allows the server to uniquely identify the network system for which the address was required & therefore to provide the required address. The address resolution procedure is completed when the client receives a response from the server containing the required address.
- An Ethernet network uses two hardware addresses which identify the source & destination of each frame sent by the Ethernet.
- The destination address (all 1's) may also identify a broadcast packet (to be sent to all connected computers).
- The hardware address is also known as the **Medium Access Control (MAC) address**, in reference to the standards which define Ethernet.

- Each computer network interface card is allocated a globally unique 6-byte link address when the factory manufactures the card (stored in a PROM). This is the normal link source address used by an interface.
- A computer sends all packets which it creates with its own hardware source link address, & receives all packets which match the same hardware address in the destination field or one (or more) pre-selected broadcast/multicast addresses.
- The Ethernet address is a link layer address & is dependent on the interface card which is used.
- IP operates at the network layer & is not concerned with the link addresses of individual nodes which are to be used. The Address Resolution Protocol (ARP) is therefore used to translate between the two types of address. The ARP client & server processes operate on all computers using IP over Ethernet.
- The processes are normally implemented as part of the software driver that drives the network interface card.
- There are four types of ARP messages that may be sent by the ARP protocol. These are identified by four values in the "operation" field of an ARP message. The types of messages are:
    - ARP-Request (Broadcast, source IP address of the requester)
    - ARP-Reply (Unicast to requester, the target)
- The format of an ARP message is shown below:



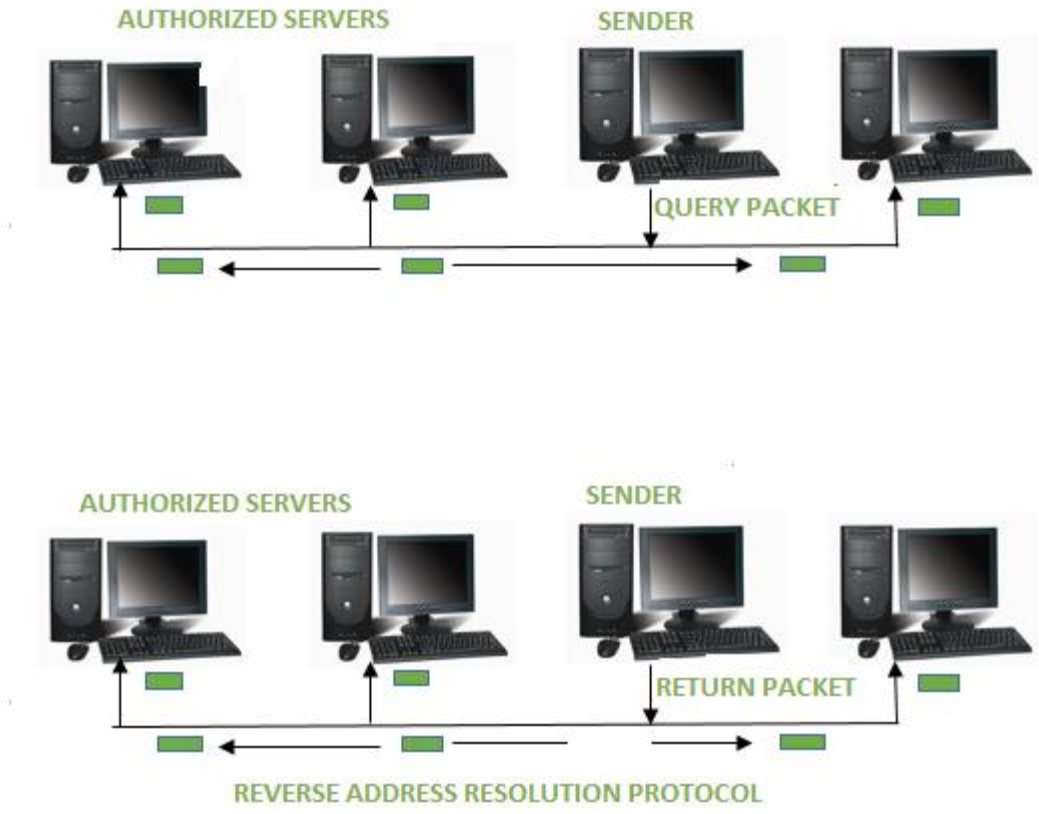

- To reduce the number of address resolution requests, a client normally **caches** resolved addresses for a (short) period of time.

- The ARP cache is of a finite size, & would become full of incomplete & obsolete entries for computers that are not in use if it was allowed to grow without check.
- The ARP cache is therefore periodically flushed of all entries. This deletes unused entries & frees space in the cache. It also removes any unsuccessful attempts to contact computers which are not currently running.

If a host changes the MAC address it is using, this can be detected by other hosts when the cache entry is deleted & a fresh ARP message is sent to establish the new association. The use of gratuitous ARP (e.g., triggered when the new NIC interface is enabled with an IP address) provides a more rapid update of this information.

# Reverse Address Resolution Protocol (RARP)

- A protocol based on computer networking
- Employed by a client computer to request its IP address from a gateway server's Address Resolution Protocol table or cache.
- The network administrator creates a table in gateway-router, which is used to map the MAC address to corresponding IP address.
- This protocol is used to communicate data between two points in a server. The client doesn't necessarily need prior knowledge of the server identities capable of serving its request.
- Medium Access Control (MAC) addresses requires individual configuration on the servers done by an administrator.
- RARP limits to the serving of IP addresses only.
- When a replacement machine is set up, the machine may or might not have an attached disk that may permanently store the IP Address so the RARP client program requests IP Address from the RARP server on the router.
- The RARP server will return the IP address to the machine under the belief that an entry has been setup within the router table.




- RARP was proposed in 1984 by the university Network group. This protocol provided the IP Address to the workstation. These diskless workstations were also the platform for the primary workstations from Sun Microsystems.

## Working of RARP

- The RARP is on the Network Access Layer & is employed to send data between two points in a network.
- Each network participant has two unique addresses:
  - IP address (a logical address)
  - MAC address (the physical address)
- The IP address gets assigned by software & after that the MAC address is constructed into the hardware.
- The RARP server that responds to RARP requests, can even be any normal computer within the network. But it must hold the data of all the MAC addresses with their assigned IP addresses.
- If a RARP request is received by the network, only these RARP servers can reply to it. The info packet needs to be sent on very cheap layers of the network. This implies that the packet is transferred to all the participants at the identical time.

- The client broadcasts a RARP request with an Ethernet broadcast address & with its own physical address. The server responds by informing the client its IP address.

| RARP | ARP |
|---|---|
| RARP stands for Reverse Address Resolution Protocol | ARP stands for Address Resolution Protocol |
| In RARP, we find our own IP address | In ARP, we find the IP address of a remote machine |
| The MAC address is known and the IP address is requested | The IP address is known, and the MAC address is being requested |
| It uses the value 3 for requests and 4 for responses | It uses the value 1 for requests and 2 for responses |

## Uses of RARP:

- RARP is used to convert the Ethernet address to an IP address.
- It is available for the LAN technologies like FDDI, token ring LANs, etc.

## Disadvantages of RARP:

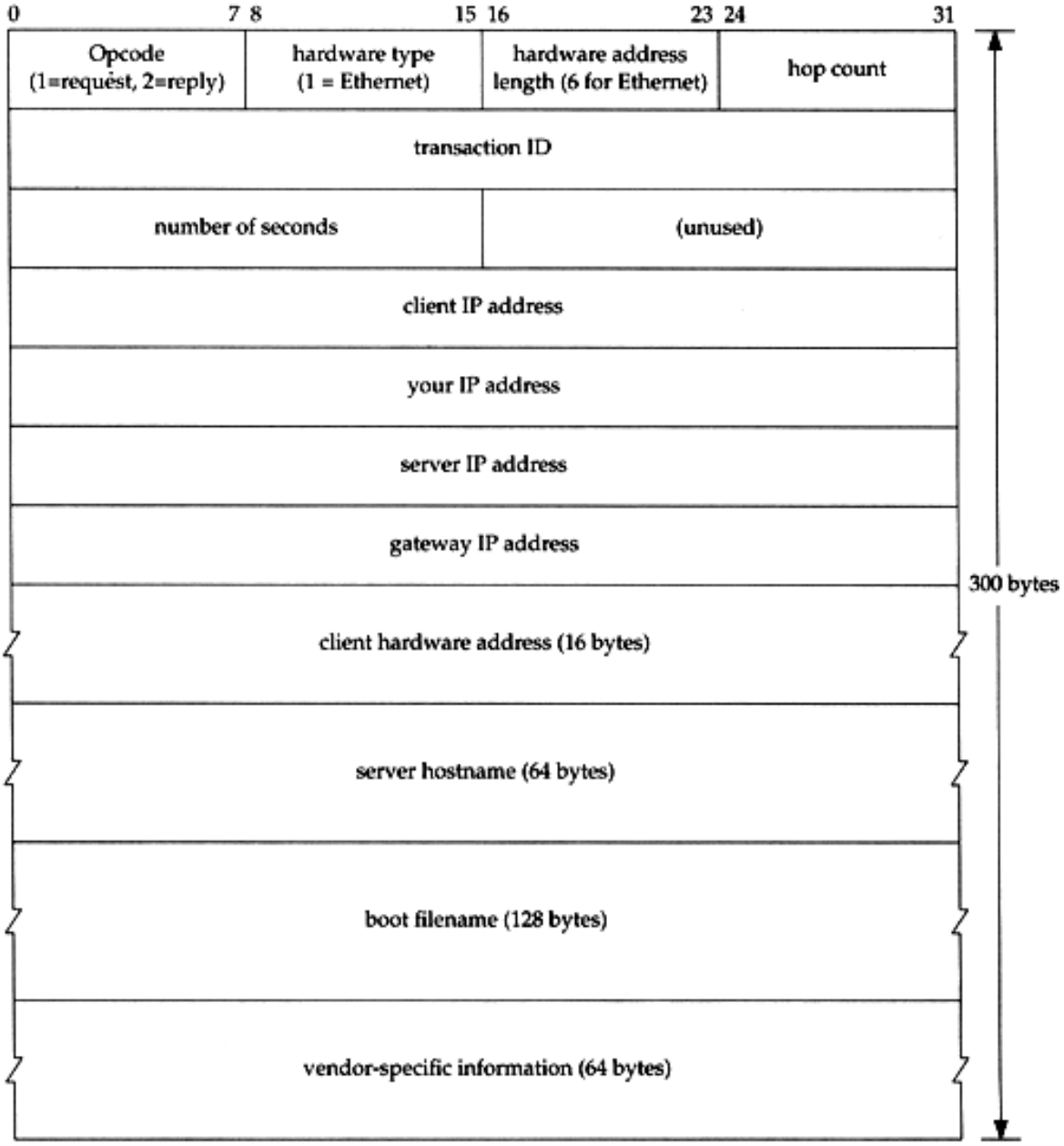
The Reverse Address Resolution Protocol had few disadvantages which eventually led to its replacement by BOOTP & DHCP. Some of the disadvantages are listed below:

- The RARP server must be located within the same physical network.
- The computer sends the RARP request on very cheap layer of the network. Thus, it's unattainable for a router to forward the packet because the computer sends the RARP request on very cheap layer of the network.
- The RARP can't handle the subnetting process because no subnet masks are sent. If the network is split into multiple subnets, a RARP server must be available with each of them.
- It isn't possible to configure the PC in a very modern network.
- It doesn't fully utilize the potential of a network like Ethernet.

RARP has now become an outdated protocol since it operates at low level. Due to this, it requires direct address to the network which makes it difficult to build a server.

# Bootstrap Protocol (BOOTP)

- The Bootstrap protocol (BOOTP) is a client/server protocol designed to provide physical address to logical address mapping.
- BOOTP is an application layer protocol, administrator may put the client & server on the same network or on different network.
- BOOTP message are encapsulated in a UDP packet, & the UDP packet itself is encapsulated in an IP packet.
- The client may be unknown about IP address, but it needs to send IP datagram.
- The client simply uses all 0's as the source address & all 1's as the destination address.
- One of the advantages of BOOTP over RARP is that the client & server are application layer processes.
- In client & server on different network:
    - The BOOTP request is broadcast because the client doesn't know the IP address of server.
    - A broadcast IP datagram can't pass through any router. So, there is a need for an intermediary.
    - One of the hosts can be used as a relay (Relay agent). The relay agent knows the unicast address of BOOTP server.
    - When relay agent receives BOOTP request packet, it encapsulates the message in a unicast datagram & send the request to the BOOTP server.
    - BOOTP server knows that the message comes from a relay agent because one of the fields in the request message define the IP address of relay agent.
    - The relay agent, after receiving reply, send it to BOOTP client.




- The disadvantage of RARP is that it uses a destination address of all 1's (limited broadcasting) to reach the RARP server. But such broadcasts are not forwarded by routers, so a RARP server is needed on each network.
- To get around the problems associated with RARP, an alternative bootstrap protocol called BOOTP has been developed, which is defined in RFCs 951, 1048, & 1084.
- Unlike RARP, BOOTP uses User Datagram Protocol (UDP) messages for connectionless transmission, which are forwarded over routers. It also provides a system with additional information, including the IP address of the file server holding the memory image, the IP address of the default router, & the subnet mask to use.

- BOOTP (Bootstrap Protocol) is a protocol that lets a network user be automatically configured (receive an IP address) & have an operating system booted (initiated) without user involvement. The BOOTP server, managed by a network administrator, automatically assigns the IP address from a pool of addresses for a certain duration of time.

- BOOTP is the basis for a more advanced network manager protocol, the Dynamic Host Configuration Protocol (DHCP). BOOTP is implemented using the User Datagram Protocol (UDP) as transport protocol, port number 67 is used by the (DHCP) server to receive client requests & port number 68 is used by the client to receive (DHCP) server responses. BOOTP operates only on IPv4 networks.

- Some parts of BOOTP have been effectively superseded by the Dynamic Host Configuration Protocol (DHCP), which adds the feature of leases, parts of BOOTP are used to provide service to the DHCP protocol. DHCP servers also provide the legacy BOOTP functionality

# Dynamic Host Configuration Protocol (DHCP)

- BOOTP is not a dynamic configuration protocol.
- When a client requests it IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address & the IP address of client already exist. The binding is predetermined.
- The **Dynamic Host Configuration Protocol (DHCP)** has been devised to provide static & dynamic address allocation that can be done by manual or automatic.

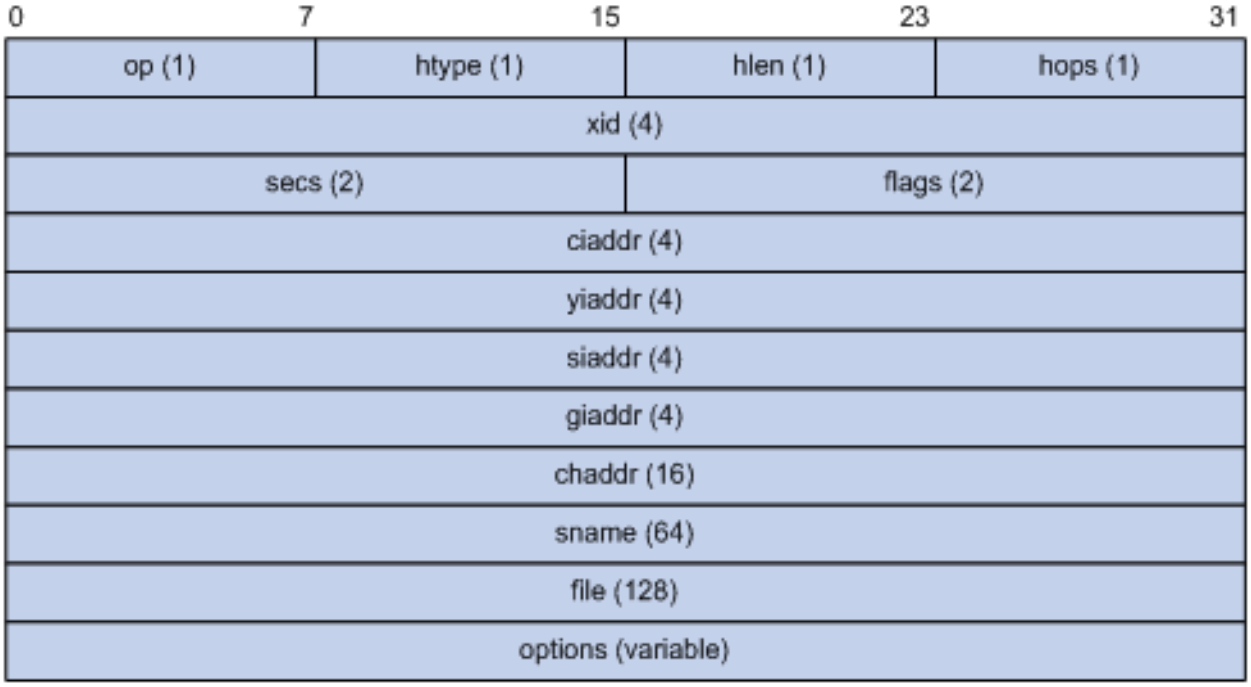
**Static address allocation**:

- In this case DHCP acts same as BOOTP does
- A host running client can request a static address from a DHCP server.
- A DHCP server has a database that statically binds the physical address to IP addresses.

**Dynamic address allocation:**

- DHCP has a second database with a pool of available IP addresses. This second data base make DHCP dynamic.

- When a DHCP client request a temporary IP address, the DHCP server goes the pool of available(unused)IP addresses & assign an IP address for a negotiable period of time.
- DHCP provide temporary IP addresses for a limited time.
- The address assigned from the pool are temporary addresses the DHCP server issue lease for a specific time.




- When the Lease expires, the client must either stop using the IP address or renew the lease.

# Open Shortest Path First (OSPF) protocol

- An Interior Gateway Routing Protocol
- Internet is made up of a large number of independent networks or ASes (Autonomous Systems) that are operated by different organizations, usually a company, university, or ISP. Inside of its own network, an organization can use its own algorithm for internal routing, or **intradomain routing**, as it is more commonly known.
- But there are only a handful of standard protocols that are popular
- Now, we discuss the problem of intradomain routing & the OSPF protocol that is widely used in practice.
- An **intradomain routing protocol** is also called an **interior gateway protocol**.

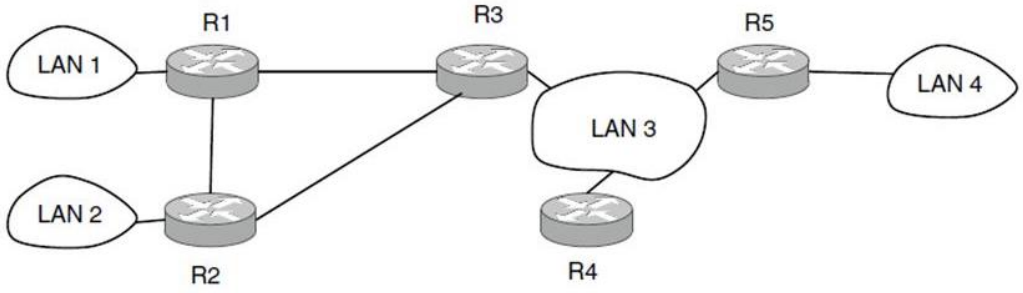
- Early intradomain routing protocols used a distance vector design, based on the distributed Bellman-Ford algorithm inherited from the ARPANET.
- RIP (Routing Information Protocol) is the main example that is used to this day.
  - It works well in small systems, but less well as networks get larger.
  - It also suffers from the count-to-infinity problem & generally slow convergence.
- Due to these problems, ARPANET switched over to a link state protocol in May 1979, & in 1988 IETF (Internet Engineering Task Force) began to work on a link state protocol for intradomain routing. That protocol, called **OSPF (Open Shortest Path First)**, became a standard in 1990.
  - It drew on a protocol called **IS-IS (Intermediate-System to Intermediate-System)**, which became an ISO standard. Because of their shared heritage, the two protocols are much more alike than different.
  - They are the dominant intradomain routing protocols, & most router vendors now support both of them.
  - **OSPF is more widely used in company networks, & IS-IS is more widely used in ISP networks.**

## Requirements while designing OSPF:

1. The algorithm had to be published in the open literature (publicly available literature) hence the ‘‘O’’ in OSPF. A proprietary solution owned by one company would not do.
2. The new protocol had to support a variety of distance metrics, including physical distance, delay, & so on.
3. It had to be a dynamic algorithm, one that adapted to changes in the topology automatically & quickly.
4. It had to support routing based on type of service. The new protocol had to be able to route real-time traffic one way & other traffic a different way. At the time, IP had a Type of service field, but no existing routing protocol used it.
5. OSPF had to do load balancing, splitting the load over multiple lines. Most previous protocols sent all packets over a single best route, even if there were two routes that were equally good. The other route was not used at all. In many cases, splitting the load over multiple routes gives better performance.

6. Support for hierarchical systems was needed. By 1988, some networks had grown so large that no router could be expected to know the entire topology. OSPF had to be designed so that no router would have to.
7. Some measure of security was required to prevent fun-loving students from spoofing routers by sending them false routing information.
8. Provision was needed for dealing with routers that were connected to the Internet via a tunnel. Previous protocols didn't handle this well.

- **OSPF supports both point-to-point links (e.g., SONET) & broadcast networks (e.g., most LANs).**
- It is able to support networks with multiple routers, that can communicate directly with the others (called multiaccess networks) even if they don't have broadcast capability. Earlier protocols didn't handle this.
- An example of an autonomous system network is given in the below figure (An autonomous system). Hosts are omitted because they don't generally play a role in OSPF, while routers & networks (which may contain hosts) do.
- Most of the routers in the below figure are connected to other routers by point-to-point links, & to networks to reach the hosts on those networks. But, routers R3, R4, & R5 are connected by a broadcast LAN such as switched Ethernet.

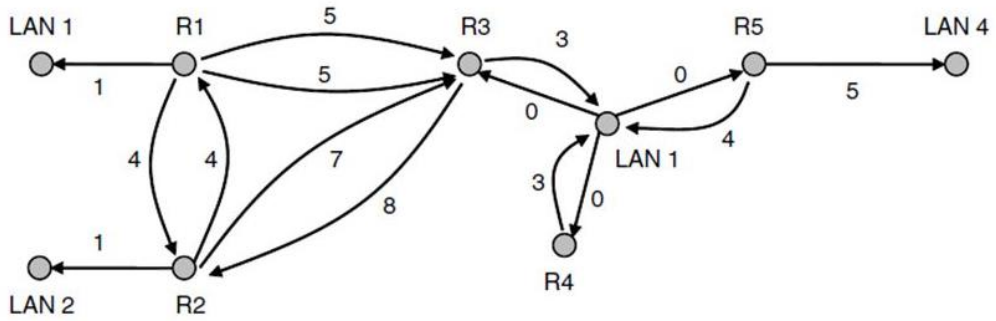



An autonomous system

- OSPF operates by abstracting the collection of actual networks, routers, & links into a directed graph in which each arc is assigned a weight (distance, delay, etc.).
- A point-to-point connection between two routers is represented by a pair of arcs, one in each direction. Their weights may be different. A broadcast network is represented by a node for the network itself, plus a

node for each router. The arcs from that network node to the routers have weight 0. They are important even so, as without them there is no path through the network. Other networks, which have only hosts, have only an arc reaching them & not one returning. This structure gives routes to hosts, but not through them.

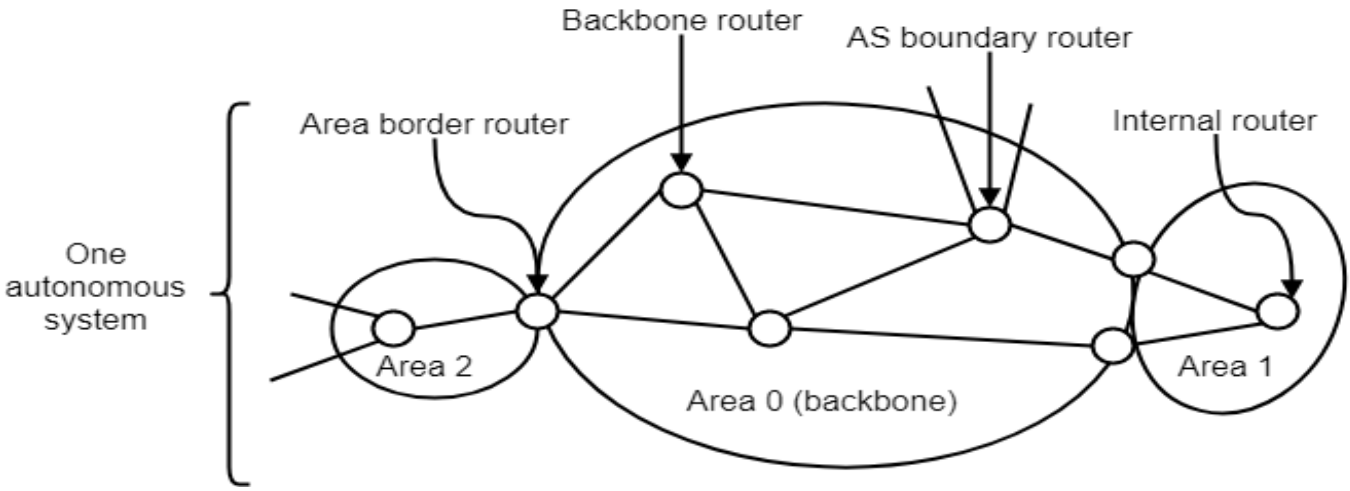
- The below figure shows the graph representation of the network of the above figure (An autonomous system). What OSPF fundamentally does is represent the actual network as a graph like this & then use the link state method to have every router compute the shortest path from itself to all other nodes.
- Multiple paths may be found that are equally short. In this case, OSPF remembers the set of shortest paths & during packet forwarding, traffic is split across them. This helps to balance load. It is called **ECMP (Equal Cost Multipath).**




- Many of the ASes in the Internet are large & significant to manage. So, OSPF allows an AS to be divided into numbered areas, where an area is a network or a set of contiguous networks.
  - Areas don't overlap but need not be complete, that is, some routers may belong to no area.
  - Routers that lie wholly within an area are called **internal routers**.
  - An area is a generalization of an individual network.
  - Outside an area, its destinations are visible but not its topology. This characteristic helps routing to scale.
- Every AS has a backbone area, called area 0.
  - The routers in this area are called **backbone routers**.
  - All areas are connected to the backbone, possibly by tunnels, so it is possible to go from any area in the AS to any other area in the AS via the backbone.
  - A **tunnel** is represented in the graph as just another arc with a cost.

- As with other areas, the topology of the backbone is not visible outside the backbone.

- Each router that is connected to two or more areas is called an **area border router**.
  - It must also be part of the backbone.
  - The job of an area border router is to summarize the destinations in one area & to inject this summary into the other areas to which it is connected.
  - This summary includes cost information but not all the details of the topology within an area.
  - Passing cost information allows hosts in other areas to find the best area border router to use to enter an area.
  - Not passing topology information reduces traffic & simplifies the shortest-path computations of routers in other areas.
  - But, if there is only one border router out of an area, even the summary doesn't need to be passed.
  - Routes to destinations out of the area always start with the instruction ***"Go to the border router"***. This kind of area is called a **stub area**.
- The last kind of router is the **AS boundary router**.
  - It injects routes to external destinations on other ASes into the area.
  - The external routes then appear as destinations that can be reached via the AS boundary router with some cost.
  - An external route can be injected at one or more AS boundary routers.
  - The relationship between ASes, areas, & the various kinds of routers is shown below. One router may play multiple roles, for example, a border router is also a backbone router.



- During normal operation, each router within an area has same link state database & runs same shortest path algorithm.
    - Its main job is to calculate the shortest path from itself to every other router & network in the entire AS.
    - An area border router needs the databases for all the areas to which it is connected & must run the shortest path algorithm for each area separately.
- For a source & destination in the same area, the best intra-area route (that lies wholly within the area) is chosen.
    - For a source & destination in different areas, the inter-area route must go from the source to the backbone, across the backbone to the destination area, & then to the destination.
    - This algorithm forces a star configuration on OSPF, with the backbone being the hub & the other areas being spokes.
    - Since the route with the lowest cost is chosen, routers in different parts of the network may use different area border routers to enter the backbone & destination area.
    - Packets are routed from source to destination ***"as is"***, (they are not encapsulated or tunnelled, unless going to an area whose only connection to the backbone is a tunnel).
    - Also, routes to external destinations may include the external cost from the AS boundary router over the external path, or just the cost internal to the AS.
- When a router boots, it sends HELLO messages on all of its point-to-point lines & multicasts them on LANs to the group consisting of all the other routers. From the responses, each router learns who its neighbours are. Routers on the same LAN are all neighbours.
- OSPF works by exchanging information between adjacent routers, which is not the same as between neighbouring routers.
    - It is inefficient to have every router on a LAN talk to every other router on the LAN.
    - To avoid this situation, one router is elected as the designated router.
    - It is said to be adjacent to all the other routers on its LAN, & exchanges information with them.
    - In effect, it is acting as the single node that represents the LAN.
    - Neighbouring routers that are not adjacent don't exchange information with each other.

- A backup designated router is always kept up to date to ease the transition when the primary designated router crash & need to be replaced immediately.

- During normal operation, each router periodically floods **LINK STATE UPDATE** messages to each of its adjacent routers.
  - These messages give its state & provide the costs used in the topological database.
  - The flooding messages are acknowledged, to make them reliable.
  - Each message has a sequence number, so a router can see whether an incoming LINK STATE UPDATE is older or newer than what it currently has.
  - Routers also send these messages when a link goes up or down or its cost changes.
- **DATABASE DESCRIPTION** messages give the sequence numbers of all the link state entries currently held by the sender.
  - By comparing its own values with those of the sender, the receiver can determine who has the most recent values.
  - These messages are used when a link is brought up.
  - Either partner can request link state information from the other one by using LINK STATE REQUEST messages.
  - The result of this algorithm is that each pair of adjacent routers checks to see who has the most recent data, & new information is spread throughout the area this way.
  - All these messages are sent directly in IP packets.
  - The five kinds of messages are summarized here:

| Message type | Description |
|---|---|
| Hello | Used to discover who the neighbors are |
| Link state update | Provides the sender's costs to its neighbors |
| Link state ack | Acknowledges link state update |
| Database description | Announces which updates the sender has |
| Link state request | Requests information from the partner |

- Finally, we can put all the pieces together.
  - Using flooding, each router informs all the other routers in its area of its links to other routers & networks & the cost of these links.

- This information allows each router to construct the graph for its area(s) & compute the shortest paths.
- The backbone area also does this work.
- Also, the backbone routers accept information from the area border routers in order to compute the best route from each backbone router to every other router.
- This information is propagated back to the area border routers, which advertise it within their areas.
- Using this information, internal routers can select the best route to a destination outside their area, including the best exit router to the backbone.

## OSPF supports/provides/advantages –

- Both IPv4 & IPv6 routed protocols
- Load balancing with equal-cost routes for the same destination
- VLSM (Variable Length Subnet mask) & route summarization
- Unlimited hop counts
- Trigger updates for fast convergence
- A loop-free topology using SPF (Shortest Path First) algorithm
- Run-on most routers
- Classless protocol

There are some **disadvantages of OSPF** like:

- It requires an extra CPU process to run the SPF algorithm
- Requiring more RAM to store adjacency topology
- Being more complex to set up & hard to troubleshoot

# Border Gateway Protocol (BGP)

- Now, we will discuss the problem of routing between independently operated networks, or interdomain routing.
- For that case, all networks must use the same **interdomain routing protocol** or **exterior gateway protocol**. The protocol that is used in the Internet is **BGP (Border Gateway Protocol).**
- Within a single autonomous system (AS), OSPF & IS-IS are the protocols that are commonly used.
    - Between ASes, a different protocol, called BGP (Border Gateway Protocol), is used.

    - A different protocol is needed because the goals of an intradomain protocol & an interdomain protocol are not the same.
    - All an intradomain protocol has to do is move packets as efficiently as possible from the source to the destination. It doesn't have to worry about politics.
- In contrast, interdomain routing protocols have to worry about politics.
    - For example, a corporate AS might want the ability to send packets to any Internet site & receive packets from any Internet site.
    - But it might be unwilling to carry transit packets originating in a foreign AS & ending in a different foreign AS, even if its own AS is on the shortest path between the two foreign ASes.
    - On the other hand, it might be willing to carry transit traffic for its neighbours, or even for specific other ASes that paid it for this service.
    - Telephone companies, for example, might be happy to act as carriers for their customers, but not for others.
    - Exterior gateway protocols in general, & BGP in particular, have been designed to allow many kinds of routing policies to be enforced in the interAS traffic.
- Typical policies involve political, security, or economic considerations.
- A few examples of possible routing constraints are:
    1. Don't carry commercial traffic on the educational network.
    2. Never send traffic from the Pentagon on a route through Iraq.
    3. Use TeliaSonera instead of Verizon because it is cheaper.
    4. Don't use AT&T (American Telephone & Telegraph) in Australia because performance is poor.
    5. Traffic starting or ending at Apple shouldn't transit Google.
- A routing policy is implemented by deciding what traffic can flow over which of the links between ASes.
    - One common policy is that a customer ISP pays another provider ISP to deliver packets to any other destination on the Internet & receive packets sent from any other destination.
    - The customer ISP is said to buy transit service from the provider ISP.
    - This is just like a customer at home buying Internet access service from an ISP.
    - To make it work, the provider should advertise routes to all destinations on the Internet to the customer over the link that connects them.

    - In this way, the customer will have a route to use to send packets anywhere.
    - Conversely, the customer should advertise routes only to the destinations on its network to the provider.
    - This will let the provider send traffic to the customer only for those addresses; the customer doesn't want to handle traffic intended for other destinations.

- **Border Gateway Protocol (BGP)** is used to Exchange routing information for the internet & is the protocol used between ISP which are different ASes.
- The protocol can connect together any internetwork of autonomous system using an arbitrary topology.
- The only requirement is that each AS have at least one router that is able to run BGP & that is router connect to at least one other AS's BGP router.
- BGP's main function is to exchange network reachability information with other BGP systems.
- Border Gateway Protocol constructs an autonomous systems' graph based on the information exchanged between BGP routers.

## Characteristics of Border Gateway Protocol (BGP):

- **Inter-Autonomous System Configuration:** The main role of BGP is to provide communication between two autonomous systems.
- BGP supports Next-Hop Paradigm.
- Coordination among multiple BGP speakers within the AS (Autonomous System).
- **Path Information:** BGP advertisement also include path information, along with the reachable destination & next destination pair.
- **Policy Support:**
    - BGP can implement policies that can be configured by the administrator.
    - For ex: - a router running BGP can be configured to distinguish between the routes that are known within the AS & that which are known from outside the AS.
- Runs Over TCP.
- BGP conserve network Bandwidth.
- BGP supports CIDR.
- BGP also supports Security.

## Functionality of Border Gateway Protocol (BGP):

BGP peers performs 3 functions, which are given below.

- **Initial peer acquisition & authentication**: Both the peers established a TCP connection & perform message exchange that guarantees both sides have agreed to communicate.
- **Sending negative or positive reachability information.**
- **Verifies that the peers & the network connection between them are functioning correctly.**

## BGP Route Information Management Functions:

- **Route Storage:** Each BGP stores information about how to reach other networks.
- **Route Update:** In this task, Special techniques are used to determine when & how to use the information received from peers to properly update the routes.
- **Route Selection:** Each BGP uses the information in its route databases to select good routes to each network on the internet network.
- **Route advertisement:** Each BGP speaker regularly tells its peer what is knows about various networks & methods to reach them.

# Internet multicasting

The IP protocol can be involved in two types of communication

1. **Unicasting**: Communication between one sender & one receiver (one-to-one communication)
2. **Multicasting**: The same message is sent to a large number of receivers simultaneously (One to many communication)

- Normal IP communication is between one sender & one receiver. But for some applications, it is useful for a process to be able to send to a large number of receivers simultaneously.
- Examples are:
    - Streaming a live sports event to many viewers
    - Delivering program updates to a pool of replicated servers
    - Handling digital conference (i.e., multiparty) telephone calls.
- IP supports one-to-many communication, or multicasting, using class D IP addresses.

- Each class D address identifies a group of hosts.
- 28 bits are available for identifying groups, so over 250 million groups can exist at the same time.
- When a process sends a packet to a class D address, a best-effort attempt is made to deliver it to all the members of the group addressed, but no guarantees are given.
- Some members may not get the packet.

- The range of IP addresses 224.0.0.0/24 is reserved for multicast on the local network.
  - In this case, no routing protocol is needed.
  - The packets are multicast by simply broadcasting them on the LAN with a multicast address.
  - All hosts on the LAN receive the broadcasts, & hosts that are members of the group process the packet.
  - Routers don't forward the packet off the LAN.
  - Some examples of local multicast addresses are:

    224.0.0.1 All systems on a LAN
    224.0.0.2 All routers on a LAN
    224.0.0.5 All OSPF routers on a LAN
    224.0.0.251 All DNS servers on a LAN
- Other multicast addresses may have members on different networks.
  - A routing protocol is needed in this case.
  - But first the multicast routers need to know which hosts are members of a group.
  - A process asks its host to join in a specific group.
  - It can also ask its host to leave the group.
  - Each host keeps track of which groups its processes currently belong to.
  - When the last process on a host leaves a group, the host is no longer a member of that group.
  - About once a minute, each multicast router sends a query packet to all the hosts on its LAN (using the local multicast address of 224.0.0.1), asking them to report back on the groups to which they currently belong.
  - The multicast routers may or may not be collocated with the standard routers.
  - Each host sends back responses for all the class D addresses it is interested in.
  - These query & response packets use a protocol called **IGMP (Internet Group Management Protocol)**.

## Internet Group Management Protocol (IGMP)

- **The Internet Group Management Protocol (IGMP)** is a communication protocol used by host & adjacent routers on IPv4 networks to establish multicast group membership.
- IGMP is an integral part of IP multicast.
- IGMP can be used for one-to-many networking applications such as online streaming video & gaming, & allows more efficient use of resources when supporting these types of applications.
- IGMP is used on IPv4 networks.
- Multicast management IGMP is used on IPv4 networks.
- Multicast management on IPv6 network is handled by **Multicast listener Discovery (MLD)** which is a part of ICMPv6 in contrast to IGMP's bare IP encapsulation.
- IGMP operates between a host & a local multicast router.
- Switches featuring IGMP snooping derive useful information by observing these IGMP transactions.
- **Protocol Independent Multicast (PIM)** is used between the local & remote multicast routers, to direct multicast traffic from hosts sending multicasts to hosts that have registered through IGMP to receive them.
    - In **Dense Mode PIM**, a pruned reverse path forwarding tree is created. This is suited to situations in which members are everywhere in the network, such as distributing files to many servers within a data centre network.
    - In **Sparse Mode PIM**, spanning trees that are built are similar to core-based trees. This is suited to situations such as a content provider multicasting TV to subscribers on its IP network.
    - A variant of this design, called **Source-Specific Multicast PIM**, is optimized for the case that there is only one sender to the group.
- Finally, multicast extensions to BGP or tunnels need to be used to create multicast routes when the group members are in more than one AS.

# IPv6

- IPv4 is close to running out of addresses.
- Even with CIDR (Classless Inter Domain Routing) & NAT (Network Address translation) using addresses more economically, the last IPv4 addresses are expected to be assigned by ICANN (Internet Corporation for Assigned Names and Numbers) before the end of 2012.

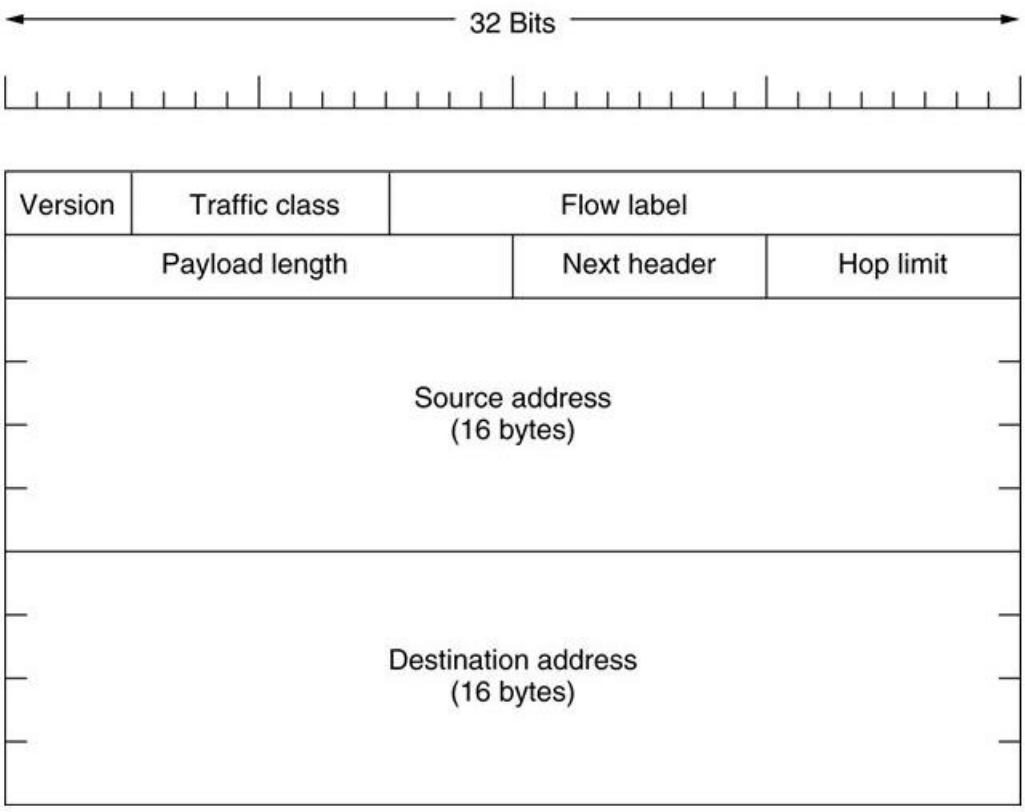
- The long-term solution is to move to larger addresses. **IPv6 (IP version 6**) is a replacement design that does just that.
    - It uses 128-bit addresses; a shortage of these addresses is not likely any time in the foreseeable future.
    - But IPv6 is very difficult to deploy.
    - It is a different network layer protocol that doesn't really interwork with IPv4, despite many similarities.
- In addition to the address problems, other issues are:
    - In the early years, the Internet was largely used by universities, high-tech industries, & the U.S. Government (especially the Dept. of Defence).
    - But now, it is used by a different group of people, often with different requirements.
    - Numerous people with smart phones use it to keep in contact with their home bases.
    - With the invention of the computer, communication, & entertainment industries, every telephone & television set in the world is an Internet node, resulting in a billion machines being used for audio & video on demand.
    - Under these circumstances, it became apparent that IP had to evolve & become more flexible.
- In 1990, IETF (Internet Engineering Task Force) worked on a new version of IP, IPv6 that will never run out of addresses, & solve a variety of other problems, & be more flexible & efficient as well.
- Its **major goals** were:
    1. Support billions of hosts, even with inefficient address allocation.
    2. Reduce the size of the routing tables.
    3. Simplify the protocol, to allow routers to process packets faster.
    4. Provide better security (authentication and privacy).
    5. Pay more attention to the type of service, particularly for real-time data.
    6. Aid multicasting by allowing scopes to be specified.
    7. Make it possible for a host to roam without changing its address.
    8. Allow the protocol to evolve in the future.
    9. Permit the old & new protocols to coexist for years.
- IPv6 meets IETF's goals fairly well.
    - It maintains the good features of IP, discards or deemphasizes the bad ones, & adds new ones where needed.
    - IPv6 is not compatible with IPv4, but it is compatible with the other auxiliary Internet protocols, including TCP, UDP, ICMP, IGMP, OSPF,

BGP, & DNS, with small modifications being required to deal with longer addresses.

- The main **features of IPv6** are discussed below:
    1. IPv6 has longer addresses than IPv4
        - They are 128 bits long.
        - Provide an effectively unlimited supply of Internet addresses
    2. Simplification of the header
        - It contains only seven fields (versus 13 in IPv4).
        - This allows routers to process packets faster
        - Improves throughput & delay
    3. Better support for options
        - Fields that previously required are now optional (since they are not used so often).
        - Also, the way options are represented is different, making it simple for routers to skip over options not intended for them.
        - This feature speeds up packet processing time.
    4. Security
        - Authentication & privacy are key features of the new IP.
        - These were later retrofitted to IPv4, but in the area of security the differences are not so great any more.
    5. Quality of service
        - Various half-hearted efforts to improve QoS have been made in the past, but now, with the growth of multimedia on the Internet, the sense of urgency is greater.

# The Main IPv6 Header

- The IPv6 header is shown below:
- The ***Version*** field is always 6 for IPv6 (& 4 for IPv4).




The IPv6 fixed header (required).

- The ***Differentiated services*** field (originally called ***Traffic class***) is used to distinguish the class of service for packets with different real-time delivery requirements.
    - It is used with differentiated service architecture for quality of service in the same manner as the field of same name in the IPv4 packet.
    - Also, the low-order 2 bits are used to signal explicit congestion indications, again in the same way as with IPv4.
- The ***Flow label*** field provides a way for a source & destination to mark groups of packets that have same requirements & should be treated in the same way by the network, forming a pseudo connection.
- The ***Payload length field*** tells how many bytes follow the 40-byte header of the above figure.
    - The name was changed from the ***IPv4 Total length field*** because the meaning was changed slightly:
        - the 40 header bytes are no longer counted as part of the length.
        - This change means the payload can now be 65,535 bytes instead of a mere 65,515 bytes.
- The ***Next header*** field tells which of the (currently) six extension headers, if any, follow this one.

- If this header is the last IP header, the Next header field tells which transport protocol handler (e.g., TCP, UDP) to pass the packet to.

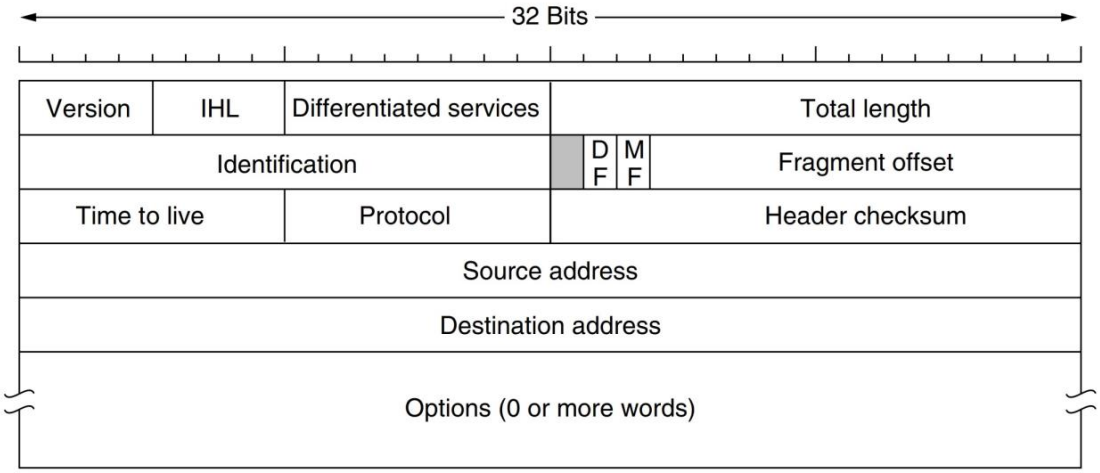
- The ***Hop limit*** field is used to keep packets from living forever.
  - It is, the same as the ***Time to live field in IPv4***, a field that is decremented on each hop.
- Next come the ***Source address*** & ***Destination address*** fields.
  - With 8-byte addresses IPv6 would run out of addresses within a few decades, whereas with **16-byte addre**sses it would never run out.
- A new notation has been devised for writing 16-byte addresses. They are written as 8 groups of 4 hexadecimal digits with colons between the groups, like this:

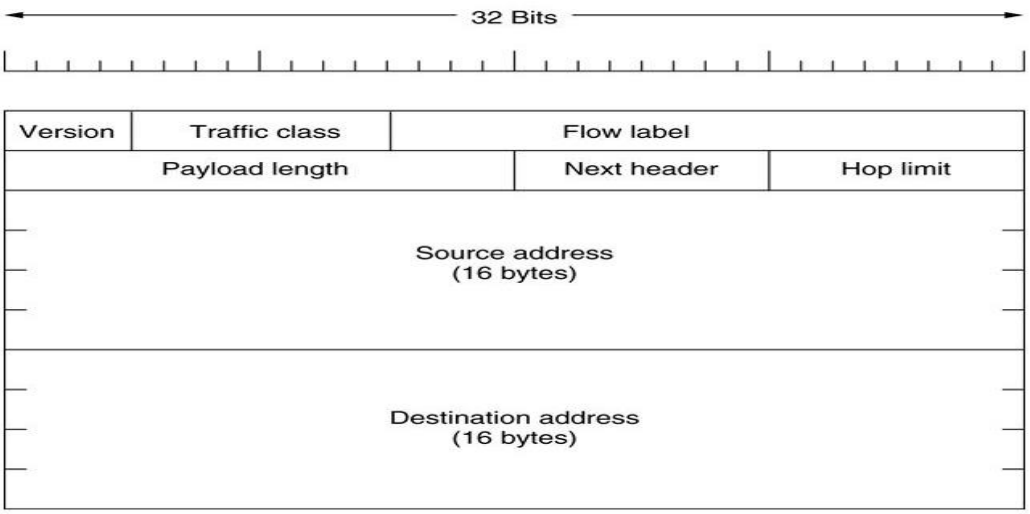
  8000:0000:0000: 0000:0123:4567:89AB: CDEF
- Since many addresses will have many zeros inside them, three optimizations have been authorized.
  1. Leading zeros within a group can be omitted, so 0123 can be written as 123.
  2. One or more groups of 16 zero bits can be replaced by a pair of colons. Thus, the above address now becomes

     8000::123:4567:89AB: CDEF
  3. IPv4 addresses can be written as a pair of colons & an old dotted decimal number, for example:

     ::192.31.20.46

- Let's compare IPv4 header (the 1st figure below) with the IPv6 header (the 2nd figure below) to see what has been left out in IPv6.
  - The ***IHL*** field is gone because the IPv6 header has a fixed length.
  - The ***Protocol*** field was taken out because the ***Next header field*** tells what follows the last IP header (e.g., a UDP or TCP segment).






The IPv6 fixed header (required).

- All the ***fields relating to fragmentation*** were removed because IPv6 takes a different approach to fragmentation.
  - When a host sends an IPv6 packet that is too large, instead of fragmenting it, the router that is unable to forward it drops the packet & sends an error message back to the sending host.
  - This message tells the host to break up all future packets to that destination.
  - Having the host send packets that are the right size in the first place is ultimately much more efficient than having the routers fragment them on the fly.
  - Also, the minimum-size packet that routers must be able to forward has been raised from 576 to 1280 bytes to allow 1024 bytes of data & many headers.
- Finally, the ***Checksum*** field is gone because calculating it greatly reduces performance.
  - With the reliable networks now used, combined with the fact that the data link layer & transport layers normally have their own checksums, the value of yet another checksum is not worth the performance.

- Removing all these features has resulted in a lean & mean network layer protocol.
- Thus, the goal of IPv6—a fast, yet flexible, protocol with plenty of address space—is met by this design.

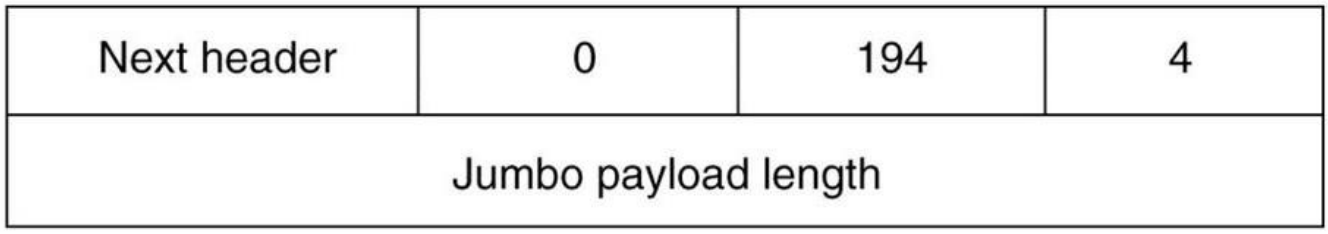
# IPv6 Extension headers

- IPv6 introduces the concept of (optional) extension headers.
- These headers can be supplied to provide extra information, but encoded in an efficient way.
- Six kinds of extension headers are defined at present, as listed below:
- Each one is optional, but if more than one is present, they must appear directly after the fixed header, & preferably in the order listed.

| Extension header | Description |
|---|---|
| Hop-by-hop options | Miscellaneous information for routers |
| Destination options | Additional information for the destination |
| Routing | Loose list of routers to visit |
| Fragmentation | Management of datagram fragments |
| Authentication | Verification of the sender's identity |
| Encrypted security payload | Information about the encrypted contents |

- Some of the headers have a fixed format; others contain a variable number of variable-length options.
    - So, each item is encoded as a (Type, Length, Value) tuple.
    - The ***Type*** is a 1-byte field telling which option this is.
        - The Type values have been chosen so that the first 2 bits tell routers that don't know how to process the option what to do.
        - The choices are:
            - Skip the option
            - discard the packet
            - Discard the packet & send back an ICMP packet
            - Discard the packet but don't send ICMP packets for multicast addresses
    - The ***Length*** is also a 1-byte field.
        - It tells how long the value is (0 to 255 bytes).
    - The ***Value*** is any information required, up to 255 bytes.
- The ***hop-by-hop*** header is used for information that all routers along the path must examine.
    - So far, one option has been defined: support of datagrams exceeding 64 KB.
    - The format of this header is shown below.

- When it is used, the ***Payload length*** field in the fixed header is set to 0.




  - Starts with a byte telling what kind of header comes next.
  - This byte is followed by one telling how long the hop-by-hop header is in bytes, excluding the first 8 bytes, which are mandatory. All extensions begin this way.
  - The next 2 bytes indicate that this option defines the ***datagram size*** (code 194) & that the size is a 4-byte number.
  - The last 4 bytes give the ***size of the datagram***.
  - Sizes less than 65,536 bytes are not permitted & will result in the first router discarding the packet & sending back an ICMP error message.
  - Datagrams using this header extension are called ***jumbo grams***.
  - The use of jumbo grams is important for supercomputer applications that must transfer gigabytes of data efficiently across the Internet.
- The ***destination options*** header is for fields that need to be interpreted only at the destination host.
  - In the initial version of IPv6, the only options defined are null options for padding this header out to a multiple of 8 bytes, so initially it won't be used.
  - It was included to make sure that new routing & host software can handle it, in case someone thinks of a destination option someday.
- The ***routing*** header lists one or more routers that must be visited on the way to the destination.
- The ***fragment*** header deals with ***fragmentation*** similarly to the way IPv4 does.
  - The header holds the datagram identifier, fragment number, & a bit telling whether more fragments will follow.
  - In IPv6, unlike in IPv4, only the source host can fragment a packet. Routers along the way may not do this.
- The ***authentication*** header provides a mechanism by which the receiver of a packet can be sure of who sent it.
  - The encrypted security payload makes it possible to encrypt the contents of a packet so that only the intended recipient can read it.

- These headers use cryptographic techniques.

# Internet Control Message Protocol version 6 (ICMPv6)

- **Internet Control Message Protocol version 6 (ICMPv6)** is the implementation of Internet Control Message Protocol (ICMP) for Internet Protocol version 6 (IPv6).
- ICMPv6 is an integral part of IPv6 & performs error reporting & diagnostic functions (e.g., ping), & has a framework for extensions to implement future changes.
- Several extensions have been published, defining new ICMPv6 message types as well as new options for existing ICMPv6 message types.

## Neighbour Discovery Protocol (NDP)

- NDP is a node discovery protocol in IPv6 which replaces & enhances functions of ARP.
- Secure Neighbour Discovery (SEND) is an extension of NDP with extra security.
- Multicast Listener Discovery (MLD) is used by IPv6 routers for discovering multicast listeners on a directly attached link, much like Internet Group Management Protocol (IGMP) used in IPv4.
- Multicast Router Discovery (MRD) allows discovery of multicast routers.

## Message types & formats

- ICMPv6 messages are classified as:
  - Error messages
  - Information messages
- ICMPv6 messages are transported by IPv6 packets in which the **IPv6 Next Header value** for ICMPv6 is set to the value **58**.
- The ICMPv6 message consists of:
  - A header
  - Protocol payload
- The **header** contains only three fields:
  - Type (8 bits)
  - Code (8 bits)
  - Checksum (16 bits)

- ***Type*** specifies the type of the message
  - Values in the range from 0 to 127 (high-order bit is 0) indicate an error message
  - Values in the range from 128 to 255 (high-order bit is 1) indicate an information message.
- The ***code*** field value depends on the message type & provides an additional level of message granularity.
- The ***checksum*** field provides a minimal level of integrity verification for the ICMP message.


## Checksum

- ICMPv6 provides a minimal level of message integrity verification by the inclusion of a ***16-bit checksum*** in its header.
- The ***checksum*** is calculated starting with ***a pseudo-header*** (below diagram) of IPv6 header fields according to the IPv6 standard, which consists of:
  - Source & destination addresses
  - Packet length
  - Next header field, which is set to a value 58


- Following this pseudo header, the checksum is continued with the ICMPv6 message.
- The **checksum computation is performed** according to Internet protocol standards **using 16-bit ones' complement summation**, followed by a final ones' complement of the checksum itself & inserting it into the checksum field.
- This differs from the way it is calculated for IPv4 in ICMP, but is similar to the calculation done in TCP.

# ICMPv6 messages

ICMPv6 messages are subdivided into two classes:

1)error messages

2)information messages.

## Error Messages

The Internet Control Message Protocol Version 6 (ICMPv6) error messages belong to four different categories:

- Destination Unreachable
- Time Exceeded
- Packet Too Big
- Parameter Problems

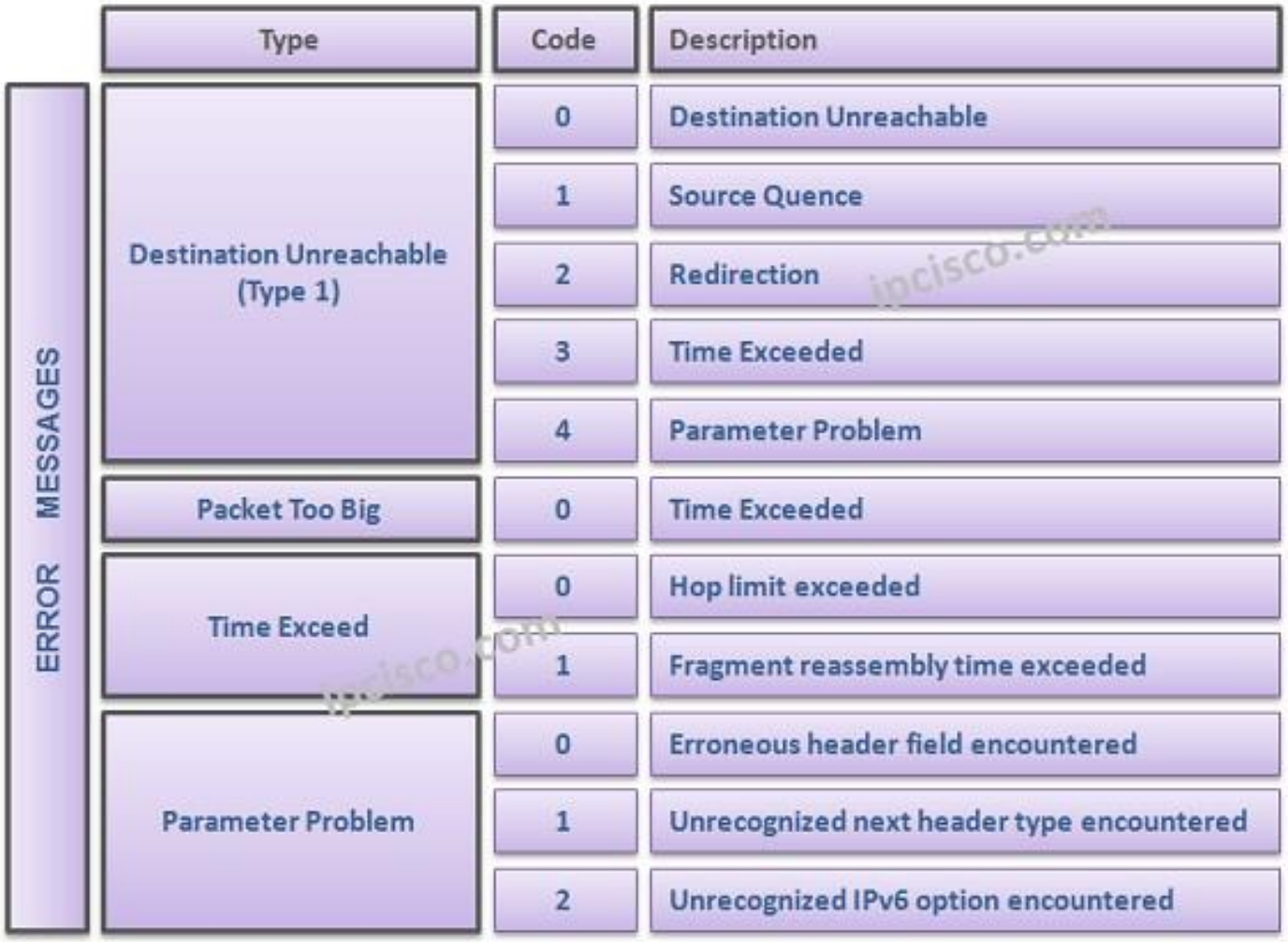
## Information Messages

The Internet Control Message Protocol Version 6 (ICMPv6) information messages are subdivided into three groups:

- Diagnostic messages
- Neighbour Discovery messages
- Messages for the management of multicast groups

## Types

- ***Control messages*** are identified by the value in the ***type*** field.
- The code field gives additional context information for the message.
- Some messages serve the same purpose as the correspondingly named ICMP message types.




## ICMPv6-Advantages & Disadvantages

### ADVANTAGES

- Provides more address space, which is being needed in larger business scales. Example: Comcast
- More powerful internet (128 bit versus IPv4's current 32 bit)
- Offers an overall larger scale internet, which, again will be needed in the future

- Address allocation is done by the device itself
- Support for security using IPsec, (Internet Protocol Security)

## DISADVANTAGES

- It will be much harder to remember IP addresses, compared to the addresses now
- Creating a smooth transition from IPv4 to IPv6 is hard
- IPv6 is not available to machines that run IPv4
- Consumer costs to replace IPv4 machine
- Time to convert over to IPv6

# Module – 5 (Transport Layer & Application Layer)

Transport service – Services provided to the upper layers, Transport service primitives. User Datagram Protocol (UDP). Transmission Control Protocol (TCP) – Overview of TCP, TCP segment header, Connection establishment &release, Connection management modelling, TCP retransmission policy, TCP congestion control.

Application Layer –File Transfer Protocol (FTP), Domain Name System (DNS), Electronic mail, Multipurpose Internet Mail Extension (MIME), Simple Network Management Protocol (SNMP), World Wide Web (WWW) – Architectural overview

# Transport Layer

- Together with network layer, transport layer is the heart of the protocol hierarchy.
- Network layer provides end-to-end packet delivery using datagrams or virtual circuits.
- *The transport layer builds on the network layer, to provide data transport from a process on a source machine to a process on a destination machine with a desired level of reliability that is independent of the physical networks currently in use.*
- It provides the abstractions that applications need to use the network.
- Without transport layer, the whole concept of layered protocols would make little sense.

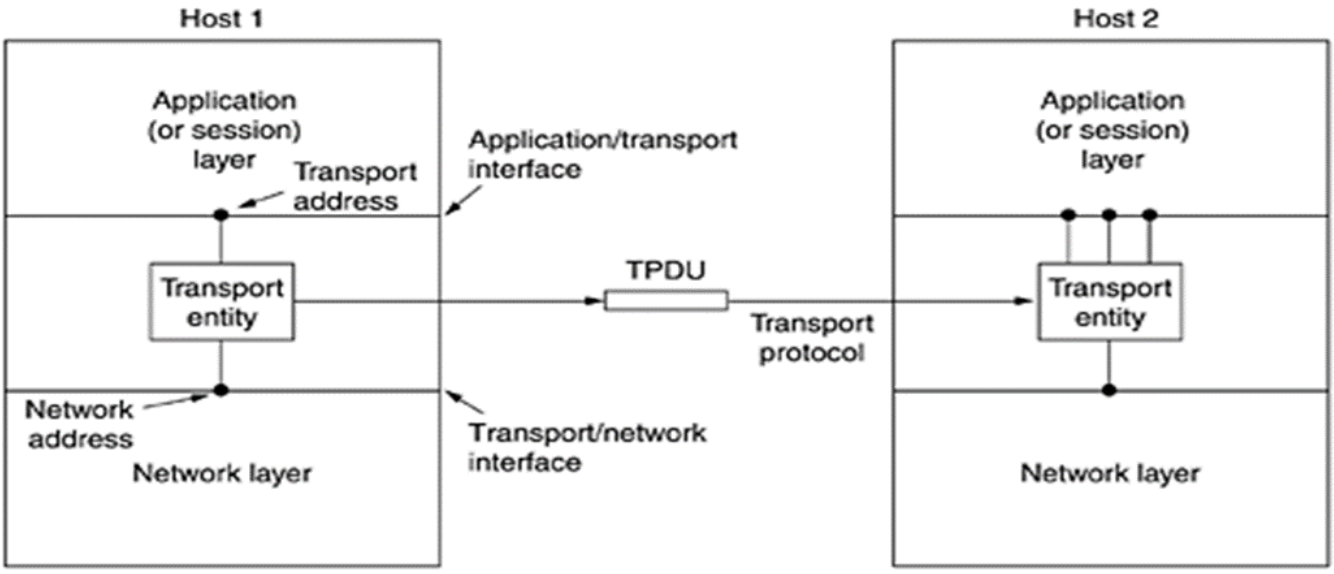
# Transport Service

- There are 2 sets of **transport layer primitives**:
    1. A simple (but hypothetical) one to show the basic ideas
    2. The interface commonly used in the Internet
- Two **types of transport services**:
    1. Connection-oriented transport service
    2. Connectionless transport service
        - Example:

        - Client-server computing
        - Streaming multimedia
- Both are similar to the network layer services. But the difference is:
    - The transport code runs entirely on the users' machines,
    - The network layer mostly runs on the routers, which are operated by the carrier
- Transport layer forms the major boundary between service provider & service user of a reliable data transmission service.

## Services provided to the upper layers

- The ultimate goal of the transport layer is to provide efficient, reliable, & cost-effective data transmission service to its users, normally processes in the application layer (upper layer).
- To achieve this, transport layer makes use of services provided by the network layer.
- The software and/or hardware within the transport layer that does the work is called the ***transport entity***.
- The ***transport entity*** can be located anywhere, such as:
    - in the operating system kernel;
    - in a library package bound into network applications;
    - in a separate user process; or
    - on the network interface card
- The first two options are most common on the Internet.
- The (logical) relationship of the network, transport, & application layer is illustrated below:




There are two types of transport service

- **Connection-oriented transport service**
  - Similar to connection-oriented network service
  - Connections have 3 phases:
    - Establishment
    - Data transfer
    - Release
  - Addressing & flow control are also similar in both layers (transport layer & network layer).
- **Connectionless transport service**
  - Similar to connectionless network service.
  - But it can be difficult to provide a connectionless transport service on top of a connection-oriented network service, since it is inefficient to set up a connection to send a single packet & then tear it down immediately afterwards.

## Difference between network layer & transport layer

- The transport code runs entirely on users' machines, but network layer mostly runs on the routers, which are operated by the carrier (at least for a wide area network).
- The users have no control over network layer, so they can't solve the problem of poor service by using better routers or putting

more error handling in the data link layer because they don't own the routers.

- The only possibility is to put another layer on top of the network layer that improves the quality of the service.
- In a connectionless network, if the packets are lost or distorted, the transport entity can detect the problem & compensate for it by using retransmissions.
- In a connection-oriented network, if a transport entity is informed halfway through a long transmission that its network connection has been abruptly terminated, with no indication of what has happened to the data currently in transit, it can set up a new network connection to the remote transport entity.
- Using this new network connection, it can send a query to its peer asking which data arrived & which didn't, & knowing where it was, pick up from where it left off.

- The transport layer makes it possible for transport service to be more reliable than underlying network.
- Also, transport primitives can be implemented as calls to library procedures to make them independent of the network primitives.
- The network service calls may vary from one network to another (e.g., calls based on a connectionless Ethernet may be quite different from calls on a connection-oriented WiMAX network).
- With the help of transport layer, application programmers can write code according to a standard set of primitives & have these programs work on a wide variety of networks, without having to worry about dealing with different network interfaces & levels of reliability.
- If all real networks were flawless & all had the same service primitives & were guaranteed no change, the transport layer might not be needed.
- But, in the real-world **transport layer** fulfils the key function of isolating the upper layers from the technology, design, & imperfections of the network.

- So, there is a qualitative distinction between layers 1 through 4 on the one hand & layer(s) above 4 on the other.
- The bottom four layers can be seen as the transport service provider, whereas the upper layer(s) are the transport service user.
- This distinction of provider versus user has a considerable impact on design of the layers & puts the transport layer in a key position, since it forms the major boundary between the provider & user of reliable data transmission service.
- It is the only level that applications see.

# Transport service primitives

- To allow users to access transport service, the transport layer must provide some operations to application programs, that is, a transport service interface.
- Each transport service has its own interface.
- The transport service is similar to network service, but there are some important differences.
- The main difference is that network service is intended to model service offered by real networks.
  - Real networks can lose packets, so the network service is generally unreliable.
- The connection-oriented transport service, is reliable. Of course, real networks are not error-free, but that is precisely the purpose of the transport layer—to provide a reliable service on top of an unreliable network.
- As an example, consider two processes on a single machine connected by a pipe in UNIX (or any other interprocess communication facility).
  - They assume the connection between them is 100% perfect.
  - They don’t want to know about acknowledgements, lost packets, congestion, or anything at all like that.
  - They want a 100% reliable connection.
  - Process A puts data into one end of the pipe, & process B takes it out of the other.

- This is what the connection-oriented transport service is all about—hiding imperfections of the network service so that user processes can just assume the existence of an error-free bit stream even when they are on different machines.

- Transport layer can also provide unreliable (datagram) service.
  - Some applications, such as client-server computing & streaming multimedia, build on a connectionless transport service.
- A second difference between network service & transport service is whom the services are intended for.
  - The network service is only used by the transport entities.
  - Few users write their own transport entities, & thus few users or programs ever see the bare network service.
  - In contrast, many programs (& thus programmers) see the transport primitives.
  - So, transport service must be convenient & easy to use.

- To get an idea of what a transport service might be like, consider the five primitives listed below:
  - This transport interface gives the essential flavour of what a connection-oriented transport interface has to do.
  - It allows application programs to establish, use, & then release connections, which is sufficient for many applications.

| Primitive | Packet sent | Meaning |
|---|---|---|
| LISTEN | (none) | Block until some process tries to connect |
| CONNECT | CONNECTION REQ. | Actively attempt to establish a connection |
| SEND | DATA | Send information |
| RECEIVE | (none) | Block until a DATA packet arrives |
| DISCONNECT | DISCONNECTION REQ. | This side wants to release the connection |

- To see how these primitives might be used, consider an application with a server & a number of remote clients.
  - To start with, the server executes a ***LISTEN*** primitive, typically by calling a library procedure that makes a system call that blocks the server until a client turns up.
  - When a client wants to talk to the server, it executes a ***CONNECT*** primitive. The transport entity carries out this primitive by blocking the caller & sending a packet to the server. A transport layer message is encapsulated in the payload of this packet for server's transport entity.
  - The client's ***CONNECT*** call causes a ***CONNECTION REQUEST*** segment to be sent to the server.
  - When it arrives, the transport entity checks to see that the server is blocked on a ***LISTEN*** (i.e., is interested in handling requests).
  - If so, it then unblocks the server & sends a ***CONNECTION ACCEPTED*** segment back to the client.
  - When this segment arrives, the client is unblocked & the connection is established.
  - Data can now be exchanged using the ***SEND*** & ***RECEIVE*** primitives.
  - Either party can do a (blocking) ***RECEIVE*** to wait for the other party to do a ***SEND***.
  - When the segment arrives, the receiver is unblocked.
  - It can then process the segment & send a reply.
  - As long as both sides can keep track of whose turn it is to send, this scheme works fine.

- In transport layer, even a simple unidirectional data exchange is more complicated than at network layer.
- Every data packet sent will also be acknowledged (eventually).
- The packets bearing control segments are also acknowledged, implicitly or explicitly.

- These acknowledgements are managed by transport entities, using network layer protocol, & are not visible to transport users.
- Similarly, transport entities need to worry about timers & retransmissions. None of this machinery is visible to transport users.
- To transport users, a connection is a reliable bit pipe: one user stuffs bits in & they magically appear in the same order at the other end. This ability to hide complexity is the reason that layered protocols are such a powerful tool.

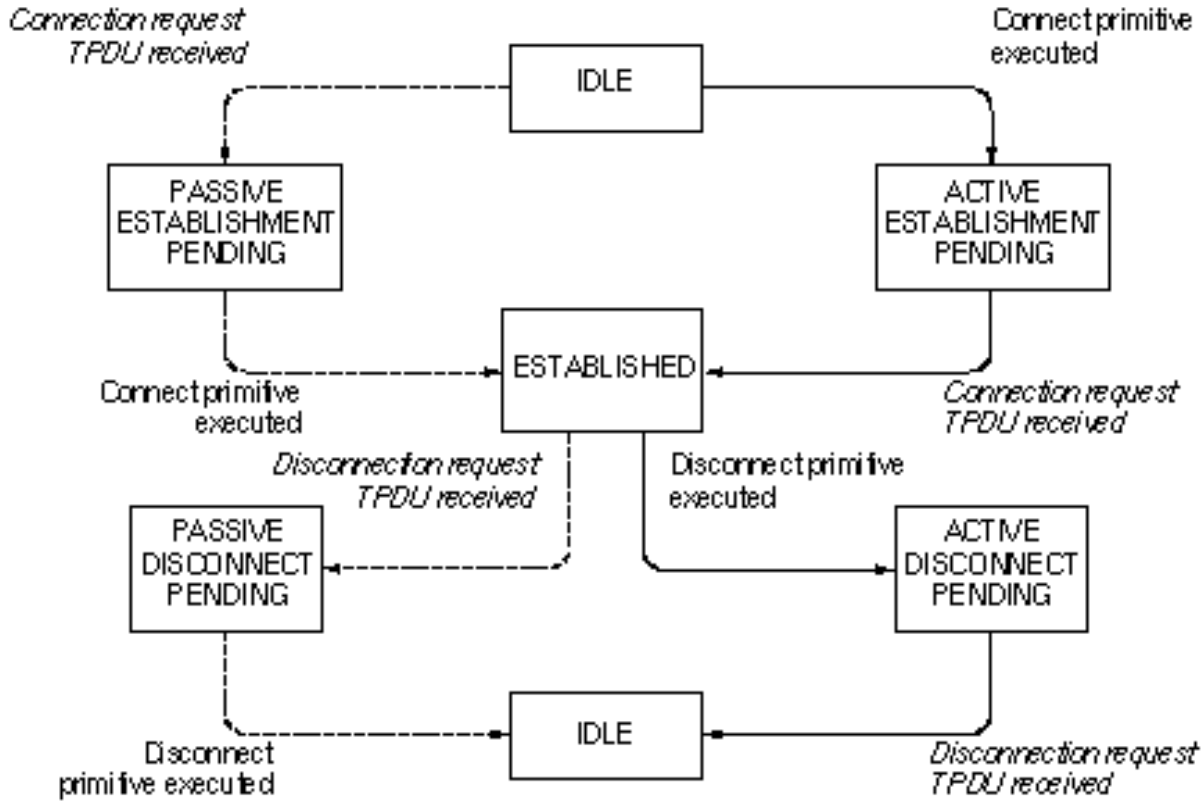
- When a connection is no longer needed, it must be released to free up table space within two transport entities.

- Disconnection has two variants:

  - Asymmetric
  - Symmetric.

- In ***asymmetric*** variant, either transport user can issue a ***DISCONNECT*** primitive, which results in a ***DISCONNECT*** segment being sent to the remote transport entity. Upon its arrival, the connection is released.
- In the ***symmetric*** variant, each direction is closed separately, independently of the other one. When one side does a ***DISCONNECT***, that means it has no more data to send but it is still willing to accept data from its partner. In this model, a connection is released when both sides have done a ***DISCONNECT***.
- A state diagram for connection establishment & release for these simple primitives is given below.
  - Each transition is triggered by some event, either a primitive executed by the local transport user or an incoming packet.

- Assume that each segment is separately acknowledged. Also assume that a symmetric disconnection model is used, with client going first.

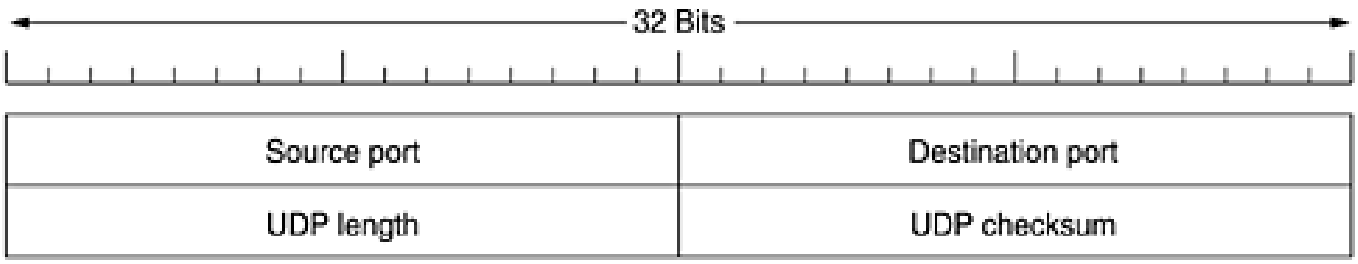



## User Datagram Protocol (UDP)

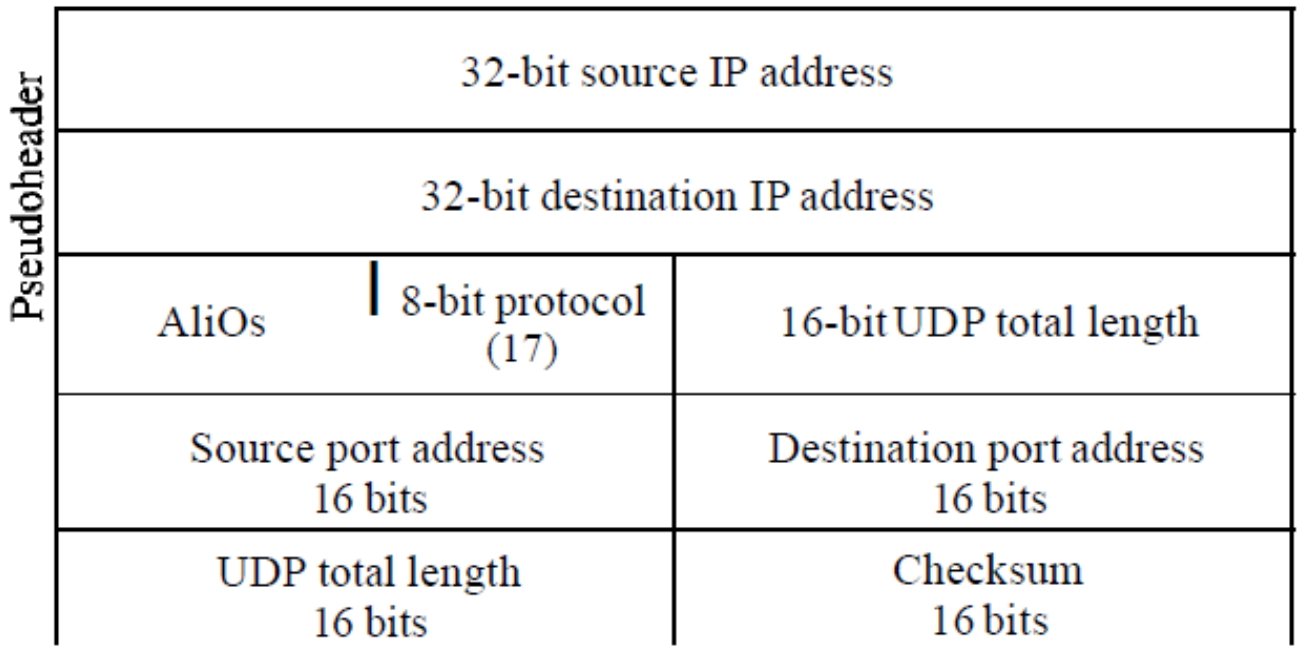
- Internet protocol suite supports a connectionless transport protocol, **UDP (User Datagram Protocol).**
- UDP provides a way for applications to send encapsulated IP datagrams & send them without having to establish a connection.
- UDP is described in RFC (Request for Comments) 768.
- UDP transmits segments consisting of an 8-byte header followed by the payload.
- The header is shown below.
- The two ports serve to identify end points within the source & destination machines.

- When a UDP packet arrives, its payload is handed to the process attached to the destination port. This attachment occurs when ***BIND*** primitive or something similar is used.
- The main intention of having UDP over using raw IP is the addition of source & destination ports.
- Without port fields, the transport layer wouldn't know what to do with the packet. With them, it delivers segments correctly.

## The UDP header




- **Source Port:** Source Port is 2 Byte long field used to identify port number of source.
- **Destination Port:** It is 2 Byte long field, used to identify the port of destined packet.
- **Length:** Length is the length of UDP including header & the data. It is 16-bits field.
- **Checksum:**
    - Checksum is 2 Bytes long field.
    - It is the 16-bit one's complement of one's complement sum of UDP header, pseudo header of information from IP header & the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.
    - UDP checksum calculation is different from IP & ICMP.
    - Here checksum includes three sections:
        - A pseudo header
        - UDP header
        - Data coming from application layer
- Pseudo header is the part of header of IP packet in which the user datagram is to be encapsulated with some fields filled with 0's. It is shown in the following figure:




- If the checksum doesn't include pseudo header, a user datagram may arrive safe & sound. But, if IP header is corrupted, it may be delivered to the wrong host.
- The ***protocol field*** is added to ensure that the packet belongs to UDP, & not to other transport-layer protocols. The value of the protocol field for UDP is 17. If this value is changed during transmission, the checksum calculation at receiver will detect it & UDP drops the packet. It is not delivered to the wrong protocol.
- The calculation of the checksum & its inclusion in a user datagram are optional. If checksum is not calculated, the field is filled with 1s.

## Applications of UDP

- Used for simple request response communication when size of data is less & hence there is lesser concern about flow & error control.
- Suitable protocol for multicasting as UDP supports packet switching.
- UDP is used for some routing update protocols like RIP (Routing Information Protocol).
- Used for real time applications which can't tolerate uneven delays between sections of a received message.

## **Following implementations uses UDP as a transport layer protocol:**

- NTP (Network Time Protocol)
- DNS (Domain Name Service)
- BOOTP, DHCP.
- NNP (Network News Protocol)
- Quote of the day protocol
- TFTP, RTSP, RIP, OSPF.

Application layer can do some of the tasks through UDP-

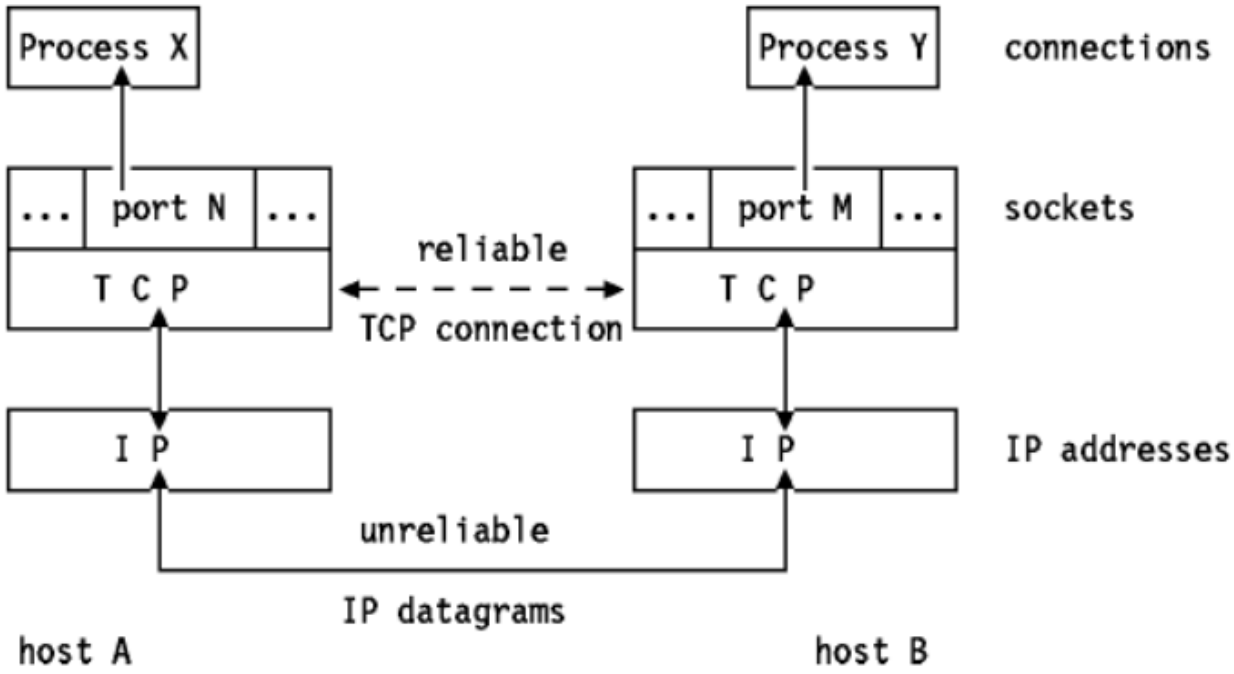
- Trace Route
- Record Route
- Time stamp

**UDP takes datagram from Network Layer, attach its header & send it to user. So, it works fast.**

# Transmission Control Protocol (TCP)

- TCP provides a connection oriented, & a reliable, byte stream service.
- The term connection-oriented means the ***two applications using TCP must establish a TCP connection with each other before they can exchange data***.
- It is a full duplex protocol, meaning that each TCP connection supports a pair of byte streams, one flowing in each direction.
- TCP includes a flow-control mechanism for each of these byte streams that allows the receiver to limit how much data the sender can transmit.
- TCP also implements a congestion-control mechanism.
- Consider the below figure:
    - Two processes communicate via TCP sockets.

- Each side of a TCP connection has a socket which can be identified by the pair < *IP_address, port_number* >.
- Two processes communicating over TCP form a logical connection that is uniquely identifiable by the two sockets involved, that is by the combination < *local_IP_address, local_port, remote_IP_address, remote_port*>.




TCP provides the following facilities to:

**Stream Data Transfer**

- From the application's viewpoint, TCP transfers a continuous stream of bytes.
- TCP does this by grouping the bytes in TCP segments, which are passed to IP for transmission to the destination.
- TCP itself decides how to segment the data & it may forward the data at its own convenience.

**Reliability**

TCP assigns a sequence number to each byte transmitted, & expects a positive acknowledgment (ACK) from the receiving TCP.

If the ACK is not received within a timeout interval, the data is retransmitted.

The receiving TCP uses the sequence numbers to rearrange the segments when they arrive out of order, & to eliminate duplicate segments.

**Flow Control**

When sending an ACK back to the sender, the receiving TCP also indicates to the sender the number of bytes it can receive beyond the last received TCP segment, without causing overrun & overflow in its internal buffers.

This is sent in the ACK in the form of highest sequence number it can receive without problems.

**Multiplexing**

To allow many processes within a single host to use TCP communication facilities simultaneously, TCP provides a set of addresses or ports within each host.

This forms a socket, concatenated with network & host addresses from internet communication layer.

A pair of sockets uniquely identifies each connection.

**Logical Connections**

The reliability & flow control mechanisms described above require that TCP initializes & maintains certain status information for each data stream.

The combination of this status, including sockets, sequence numbers & window sizes, is called a logical connection.

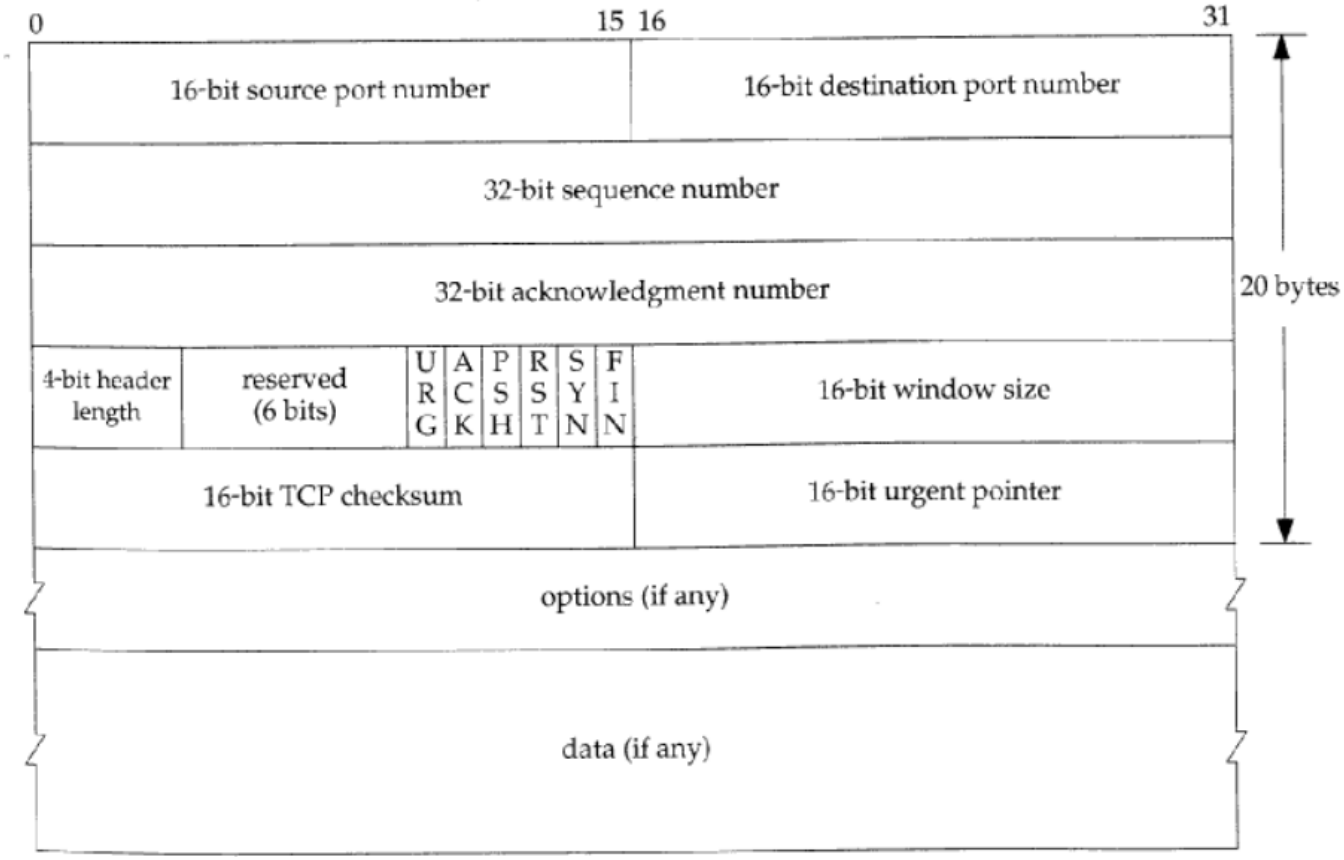
Each connection is uniquely identified by the pair of sockets used by sending & receiving processes.

**Full Duplex**

TCP provides for concurrent data streams in both directions.

# TCP segment header

- TCP data is encapsulated in an IP datagram. The figure shows the format of a TCP header. Its normal size is 20 bytes unless options are present. Each of the fields is discussed below:




- ***Source & destination port:***
  - These fields identify local endpoint of the connection.
  - Each host may decide for itself how to allocate its own ports starting at 1024.
  - The ***source & destination*** socket numbers together identify the connection.
- ***Sequence & ACK number:***
  - This field is used to give a sequence number to each & every byte transferred.
  - This has an advantage over giving the sequence numbers to every packet because data of many small packets can be combined into one at the time of retransmission, if needed.
  - The ***ACK*** signifies the next byte expected from the source & not the last byte received.

- The ***ACK***s are cumulative instead of selective.
- Sequence number space is as large as 32-bit although 17 bits would have been enough if the packets were delivered in order.
- If packets reach in order, then according to the following formula:
  *(sender's window size) + (receiver's window size) < (sequence number space)*
  the sequence number space should be 17-bits.

But packets may take different routes & reach out of order.
So, we need a larger sequence number space; & for optimization, this is 32-bits.

- ***Header length:***
  - This field tells how many 32-bit words are contained in the TCP header.
  - This is needed because the options field is of variable length.
- The 6 one-bit flags are used to relay control information between TCP peers.
  - The possible flags include ***SYN, FIN, RESET, PUSH, URG, & ACK***.
- ***URG:***
  - This bit indicates whether the urgent pointer field in this packet is being used.
- ***ACK:***
  - This bit is set to indicate the ACK number field in this packet is valid.
- ***PSH:***
  - This bit indicates PUSHed data.
  - The receiver is requested to deliver the data to the application upon arrival & not buffer it until a full buffer has been received.
- ***RST:***
  - This flag is used to reset a connection that has become confused due to a host crash or some other reason.

- It is also used to reject an invalid segment or refuse an attempt to open a connection.
- This causes an abrupt end to the connection, if it existed.

- ***SYN:***
  - This bit is used to establish connections.
  - The connection request (1st packet in 3-way handshake) has SYN=1 & ACK=0.
  - The connection reply (2nd packet in 3-way handshake) has SYN=1 & ACK=1.
- ***FIN:***
  - This bit is used to release a connection.
  - It specifies that the sender has no more fresh data to transmit.
  - But it will retransmit any lost or delayed packet.
  - Also, it will continue to receive data from other side.

  Since ***SYN*** & ***FIN*** packets have to be acknowledged, they must have a sequence number even if they don't contain any data.

- ***Window Size:***
  - Flow control in TCP is handled using a variable-size sliding window.
  - The ***Window Size*** field tells how many bytes may be sent starting at the byte acknowledged.
  - Sender can send the bytes with sequence number between (ACK#) to (ACK# + window size - 1).
  - A window size of zero is legal & says that the bytes up to & including ACK# -1 have been received, but the receiver would like no more data for the moment.
  - Permission to send can be granted later by sending a segment with the same ACK number & a nonzero ***Window Size*** field.
- ***Checksum:***
  - This is provided for extreme reliability.
  - It checksums the header, data, & the conceptual pseudo header.
  - The pseudo header contains
    - 32-bit IP address of source & destination machines
    - the protocol number for TCP (6)

        - the byte count for the TCP segment (including the header)

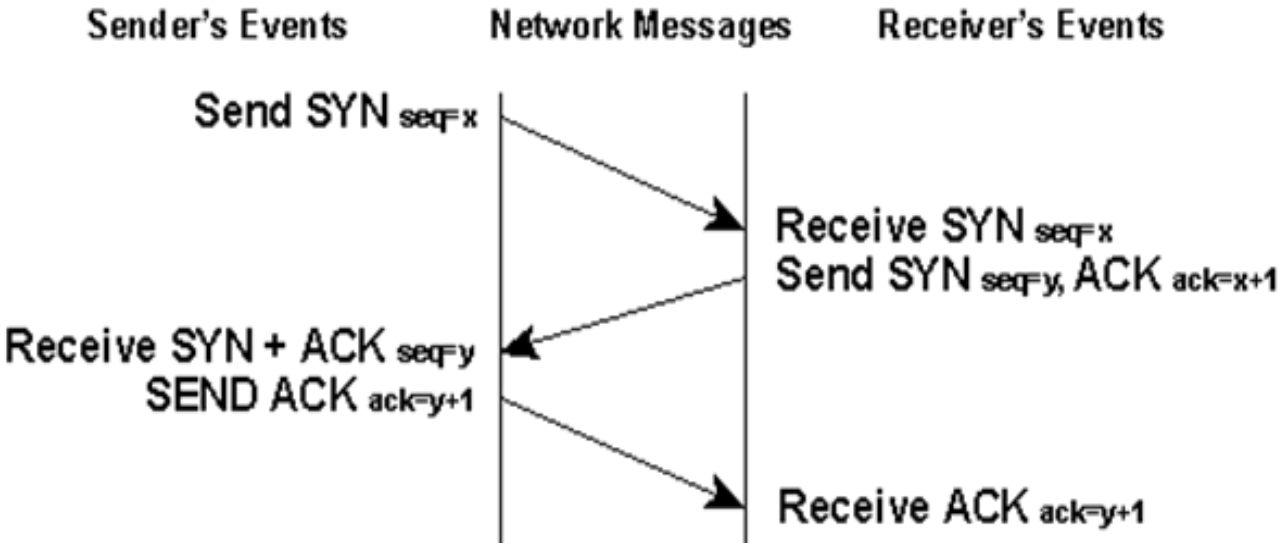
      Including the pseudo header in TCP checksum computation helps detect misdelivered packets, but doing so violates the protocol hierarchy since the IP addresses in it belong to the IP layer, not the TCP layer.

- ***Urgent Pointer:***
    - Indicates a byte offset from the current sequence number at which urgent data are to be found.
    - Urgent data continues till the end of the segment.
    - This is not used in practice.
    - The same effect can be had by using two TCP connections, one for transferring urgent data.
- ***Options:***
    - Provides a way to add extra facilities not covered by the regular header.
    - E.g., Maximum TCP payload that sender is willing to handle.
    - The maximum size of segment is called ***MSS (Maximum Segment Size).***
    - At the time of handshake, both parties inform each other about their capacity.
    - Minimum of the two is honored.
    - This information is sent in the options of the ***SYN*** packets of the three-way handshake.
    - ***Window scale*** option can be used to increase the window size. It can be specified by telling the receiver that the window size should be interpreted by shifting it left by specified number of bits. This header option allows window size up to 230.
- **Data:**
    - This can be of variable size.
    - TCP knows its size by looking at the IP size header.

# TCP Connection establishment

- ***"Three-way handshake"*** is the procedure used to establish a connection.
- This procedure is initiated by one TCP & responded by another TCP.
- The procedure also works if two TCP simultaneously initiate the procedure.
- When simultaneous attempt occurs, each TCP receives a ***"SYN"*** segment which carries no acknowledgment after it has sent a ***"SYN"***.
- Of course, the arrival of an old duplicate "***SYN***" segment can potentially make it appear to the recipient that a simultaneous connection initiation is in progress.
- Proper use of ***"reset"*** segments can disambiguate these cases.
- Three-way handshake reduces the possibility of false connections.
- It is the implementation of a trade-off between memory & messages to provide information for this checking.

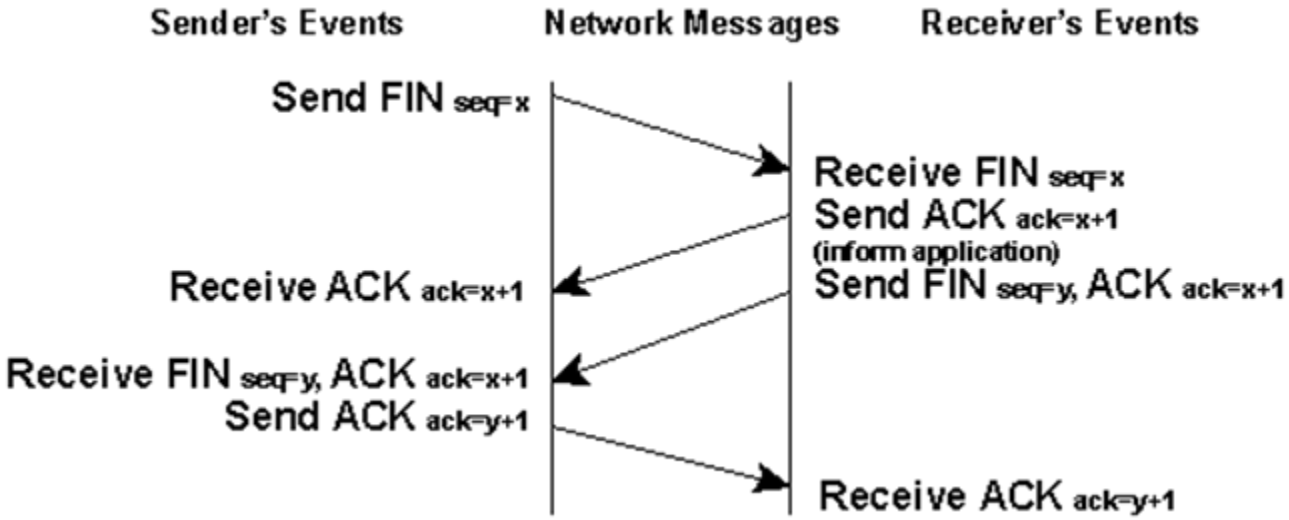
## Connection Establish




- Sender sends a ***SYN*** packet with sequence number, say ***'x'***.
- The receiver, on receiving ***SYN*** packet responds with ***SYN*** packet with sequence number ***'y'*** & ***ACK*** with sequence number ***'x+1'***
- On receiving both ***SYN*** & ***ACK*** packet, the sender responds with ***ACK*** packet with sequence number ***'y+1'***

- When the receiver receives **ACK** packet, the connection is initiated.

## TCP Connection release



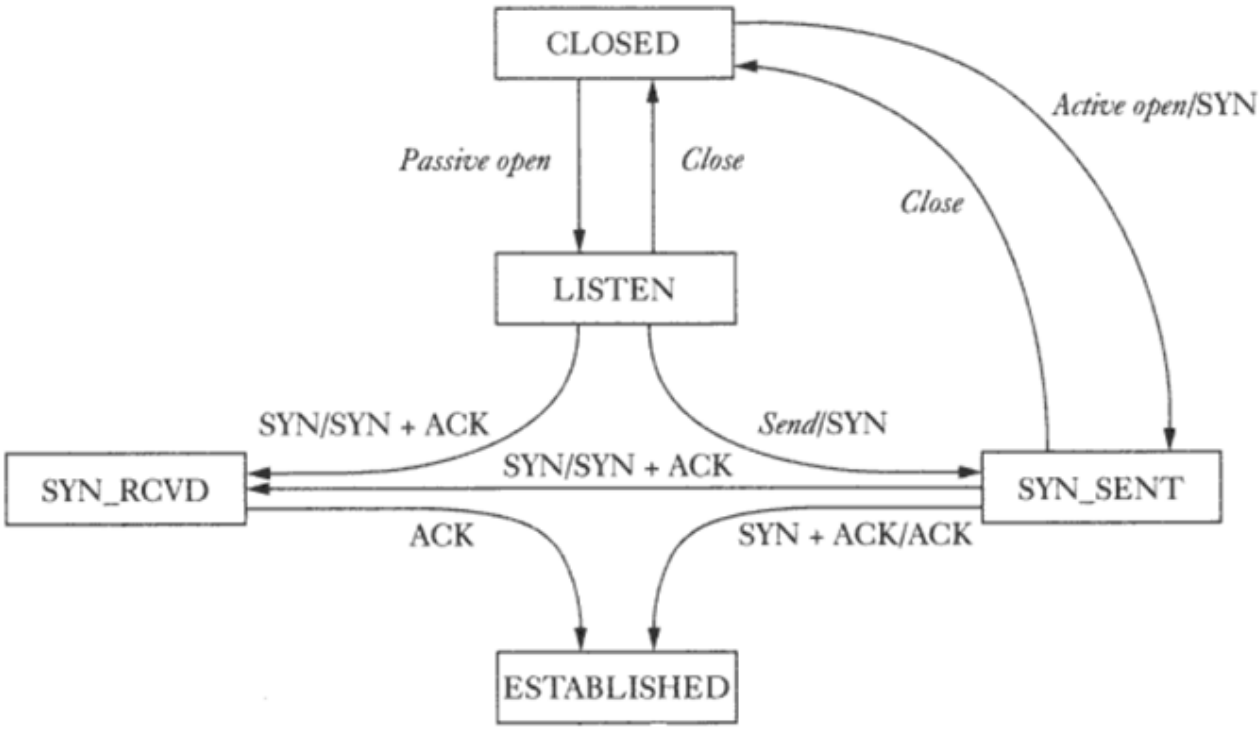

- Initiator sends a ***FIN*** with current sequence & acknowledgement number.
- The responder, on receiving this, informs the application program that it will receive no more data & sends an acknowledgement of the packet.
- The connection is now closed from one side.
- Now the responder will follow similar steps to close the connection from its side.
- Once this is done the connection will be fully closed.

## Connection management modelling

## State Diagram

- The state diagram approach to view TCP connection establishment & closing simplifies the design of TCP implementation.
- The idea is to represent TCP connection state, which progresses from one state to other as various messages are exchanged.

- To simplify the matter, we consider two state diagrams, viz., for TCP connection establishment & TCP connection closing.
- The below figure shows the state diagram for the ***TCP connection establishment***.




TCP Connection establishment

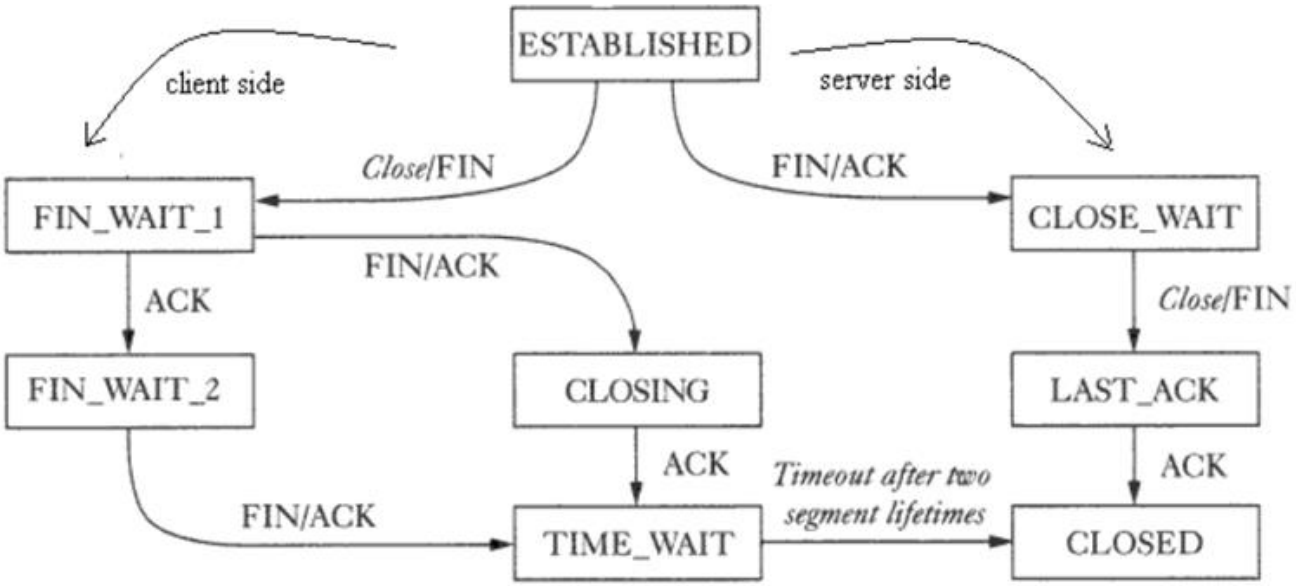
A brief description of each state of the above diagram is given below:

- ***Listen***
  - Represents the state when waiting for connection request from any remote host & port.
  - This specifically applies to a Server.
  - From this state, the server can close the service or actively open a connection by sending ***SYN***.
- ***Syn_Sent***
  - Represents waiting for a matching connection request after sending a connection request.
  - This applies to both server & client side.
  - Even though server is considered as the one with passive open, it can also send a ***SYN*** packet actively.
- ***Syn_Rcvd***

- Represents waiting for a confirmation connection request acknowledgment after having both received & sent connection request.

- ***Established***
  - Represents an open connection.
  - Data transfer can take place from this point onwards.

After the connection has been established, two end-points will exchange useful information & terminate the connection. The below figure shows the state diagram for ***terminating an active connection***:




- ***FIN_WAIT_1:***
  - Represents connection termination request from remote TCP peer, or an acknowledgment of connection termination request previously sent.
  - This state is entered when server issues close call.
- ***FIN_WAIT_2:***
  - Represents waiting for a connection termination request from the remote TCP.
- ***CLOSING:***
  - Represents connection termination request acknowledgment from the remote TCP.
- ***TIME_WAIT:***

    - This represents waiting time enough for the packets to reach their destination.
    - This waiting time is usually 4 min.
- ***CLOSE_WAIT:***
    - Represents a state when the server receives a FIN from the remote TCP, sends ***ACK*** & issues close call sending ***FIN.***
- ***LAST_ACK:***
    - Represents waiting for an ***ACK*** for the previously sent ***FIN_ACK*** to the remote TCP
- ***CLOSE:***
    - Represents a closed TCP connection having received all the ***ACK***s.

# TCP retransmission policy

- After establishing the TCP connection, sender starts transmitting TCP segments to the receiver. A TCP segment sent by the sender may get lost on the way before reaching the receiver. This causes the receiver to send acknowledgement with same ACK number to the sender. As a result, sender retransmits the same segment to the receiver. This is called as **TCP retransmission**.

**When TCP Retransmission Occurs?**

When sender discovers that the segment sent by it is lost, it retransmits the same segment to the receiver.

Sender discovers that the TCP segment is lost when:

1. Either Time Out Timer expires
2. Or it receives three duplicate acknowledgements

**1. Retransmission After Time Out Timer Expiry-**

Each time sender transmits a TCP segment to the receiver, it starts a Time Out Timer. Now, following two cases are possible-

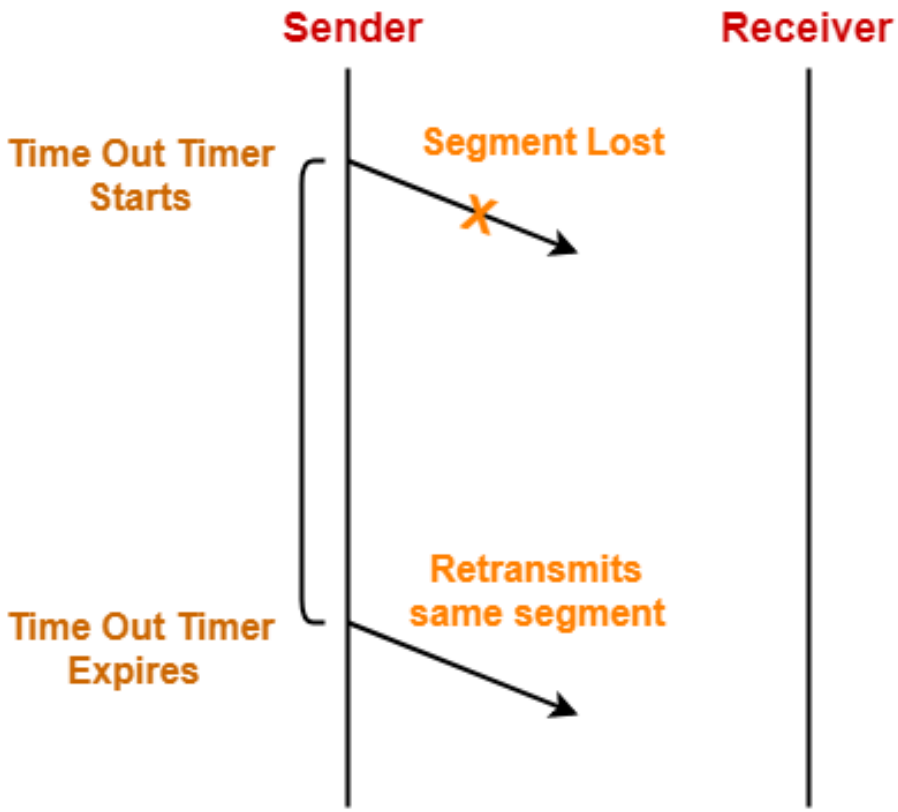
**Case-01:**

- Sender receives an acknowledgement for the ***sent*** segment before the timer goes off. In this case, sender stops the timer.

**Case-02:**

- Sender doesn't receive any acknowledgement for the sent segment & the timer goes off. In this case, sender assumes that the sent segment is lost.
- Sender retransmits the same segment to the receiver & resets the timer.

**Example-**




Retransmission after Time Out Timer Expiry

**2. Retransmission After Receiving 3 Duplicate Acknowledgements-**

- Consider the sender receives three duplicate acknowledgements for a TCP segment sent by it. Then, sender assumes that the corresponding segment is lost.

- So, sender retransmits the same segment without waiting for its time out timer to expire. This is known as **Early retransmission** or **Fast retransmission**.

**Example-**

Consider, a sender sends 5 TCP segments to the receiver. The second TCP segment gets lost before reaching the receiver.

The sequence of steps taking place are-

- On receiving segment-1, receiver sends ***acknowledgement*** asking for segment-2 next.

(Original ACK)

- On receiving segment-3, receiver sends ***acknowledgement*** asking for segment-2 next.

(1st duplicate ACK)

- On receiving segment-4, receiver sends ***acknowledgement*** asking for segment-2 next.

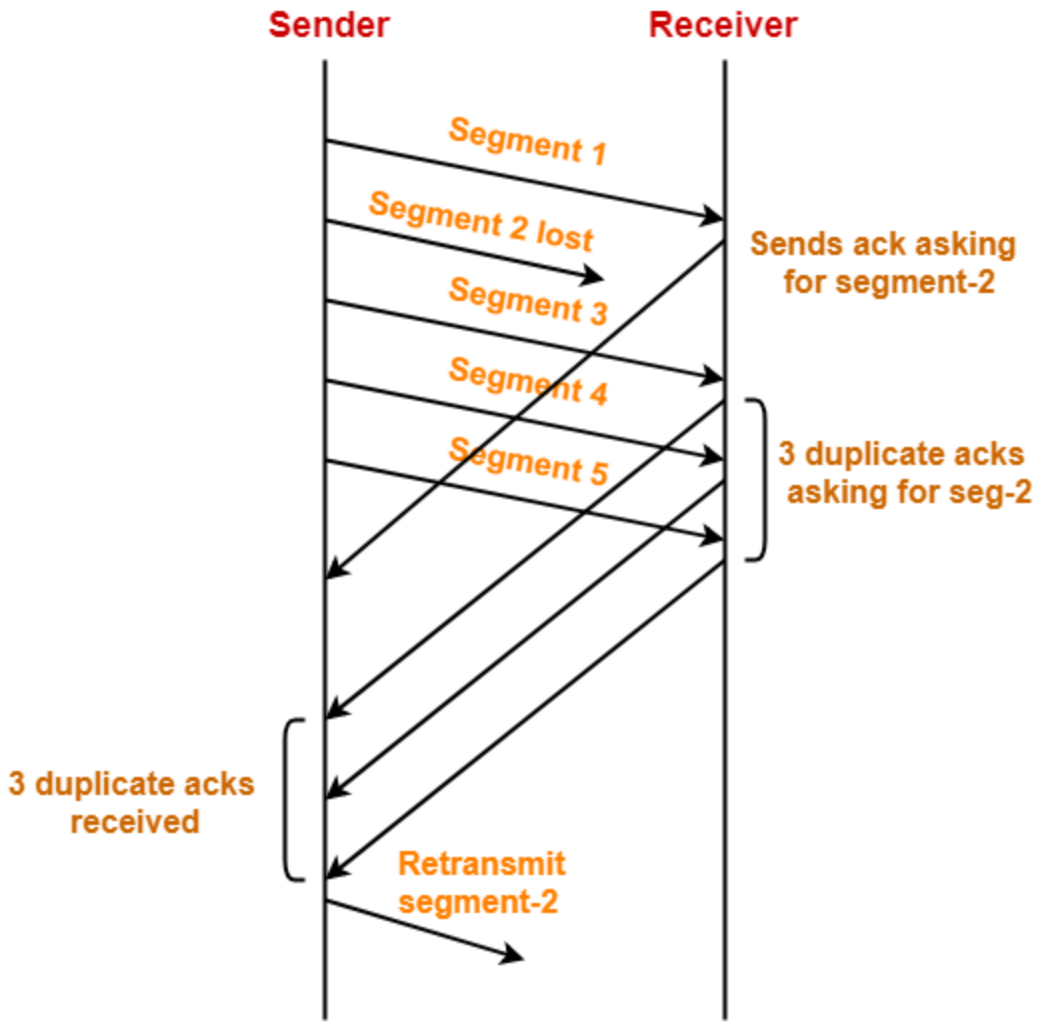
(2nd duplicate ACK)

- On receiving segment-5, receiver sends ***acknowledgement*** asking for segment-2 next.

(3rd duplicate ACK)

Now,

- Sender receives 3 duplicate acknowledgements for segment-2 in total. So, sender assumes that the segment-2 is lost. So, it ***retransmits*** segment-2 without waiting for its timer to go off.




Retransmission after receiving 3 duplicate acks

After receiving the retransmitted segment-2,

Receiver doesn't send the acknowledgement asking for segment-3 or 4 or 5. Receiver sends the acknowledgement asking for segment-6 directly from the sender. This is because previous segments have been already received & acknowledgements for them have been already sent (although wasted in asking for segment-2).

- Consider time out timer expires before receiving the acknowledgement for a TCP segment. This case suggests the stronger possibility of congestion in the network.

- Consider sender receives 3 duplicate acknowledgements for the same TCP segment. This case suggests the weaker possibility of congestion in the network.
- Consider receiver doesn't receive 3 duplicate acknowledgements for the lost TCP segment. In such a case, retransmission occurs only after time out timer goes off.
- Retransmission on receiving 3 duplicate acknowledgements is a way to improve the performance over retransmission on time out

# TCP congestion control

- Congestion in Network refers to a network state where message traffic becomes so heavy that it slows down the network response time.
    - Congestion is an important issue that can arise in ***Packet Switched Network***.
    - Congestion leads to the loss of packets in transit. So, it is necessary to control the congestion in network.
    - It is not possible to completely avoid the congestion.

## Congestion Control

Congestion control refers to techniques & mechanisms that can-

- Either prevent congestion before it happens
- Or remove congestion after it has happened

Now, let us discuss **how congestion is handled at TCP**.

## TCP Congestion Control

TCP reacts to congestion by reducing the sender window size. The size of the sender window is determined by the following two factors-

- ***Receiver window size***
- ***Congestion window size***

## Receiver Window Size-

- Receiver window size is an advertisement of ***"How much data (in bytes) the receiver can receive without acknowledgement?"***

- Sender shouldn't send data greater than receiver window size. Otherwise, it leads to dropping the TCP segments which causes ***TCP Retransmission***. So, sender should always send data less than or equal to receiver window size. Receiver dictates its window size to the sender through ***TCP Header***.

## Congestion Window-

- Sender should not send data greater than congestion window size. Otherwise, it leads to dropping the TCP segments which causes TCP Retransmission. So, sender should always send data less than or equal to congestion window size.

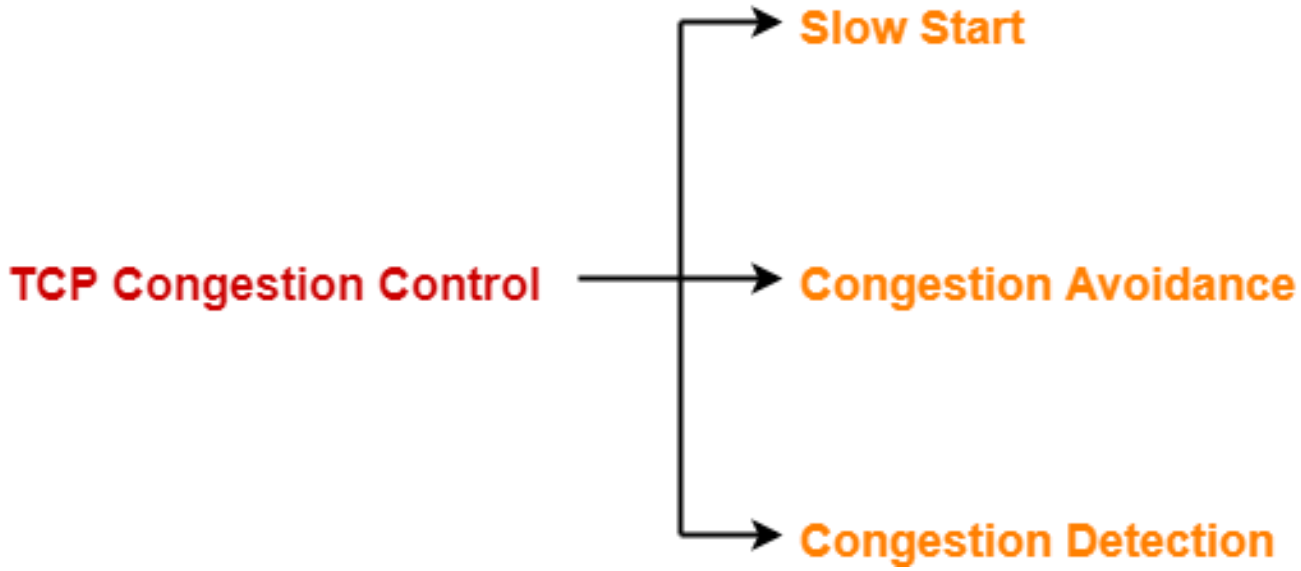
- Different variants of TCP use different approaches to calculate the size of congestion window.

- Congestion window is known only to the sender & is not sent over the links. So, always-

  ***Sender window size = Minimum (Receiver window size, Congestion window size)***

## TCP Congestion Policy-

- TCP's general policy for handling congestion consists of following three phases-



- ***Slow Start***
- ***Congestion Avoidance***
- ***Congestion Detection***

## Slow Start Phase-

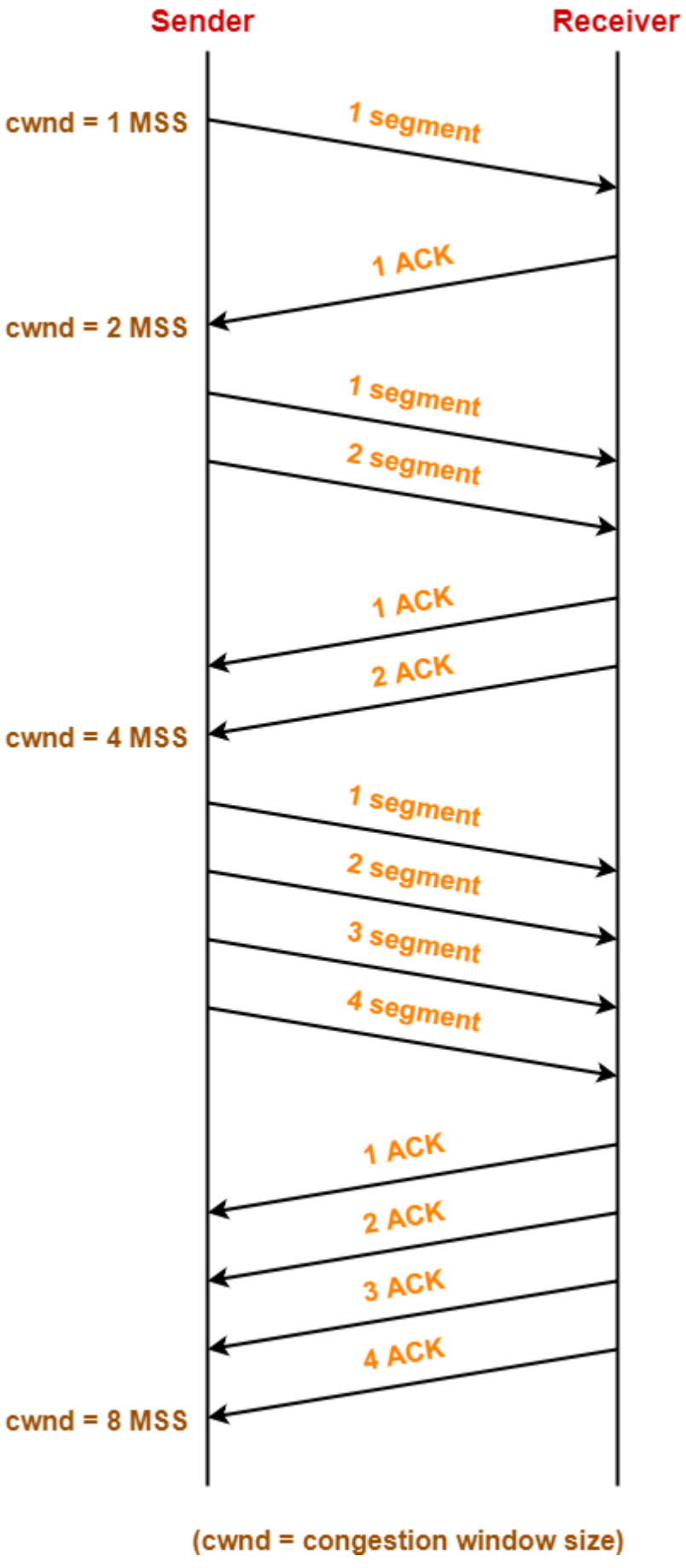
- Initially, sender sets

  congestion window size = Maximum Segment Size (1 MSS).

- After receiving each acknowledgment, sender increases the congestion window size by 1 MSS.
- In this phase, the size of congestion window increases exponentially.
- The followed formula is-

  ***Congestion window size = Congestion window size + Maximum segment size***

- This is shown below-




(cwnd = congestion window size)

- After 1 round trip time, congestion window size = $(2)^1$ = 2 MSS
- After 2 round trip time, congestion window size = $(2)^2$ = 4 MSS
- After 3 round trip time, congestion window size = $(2)^3$ = 8 MSS and so on.
- This phase continues until the congestion window size reaches the slow start threshold.

***Threshold = Maximum number of TCP segments that receiver window can accommodate / 2***

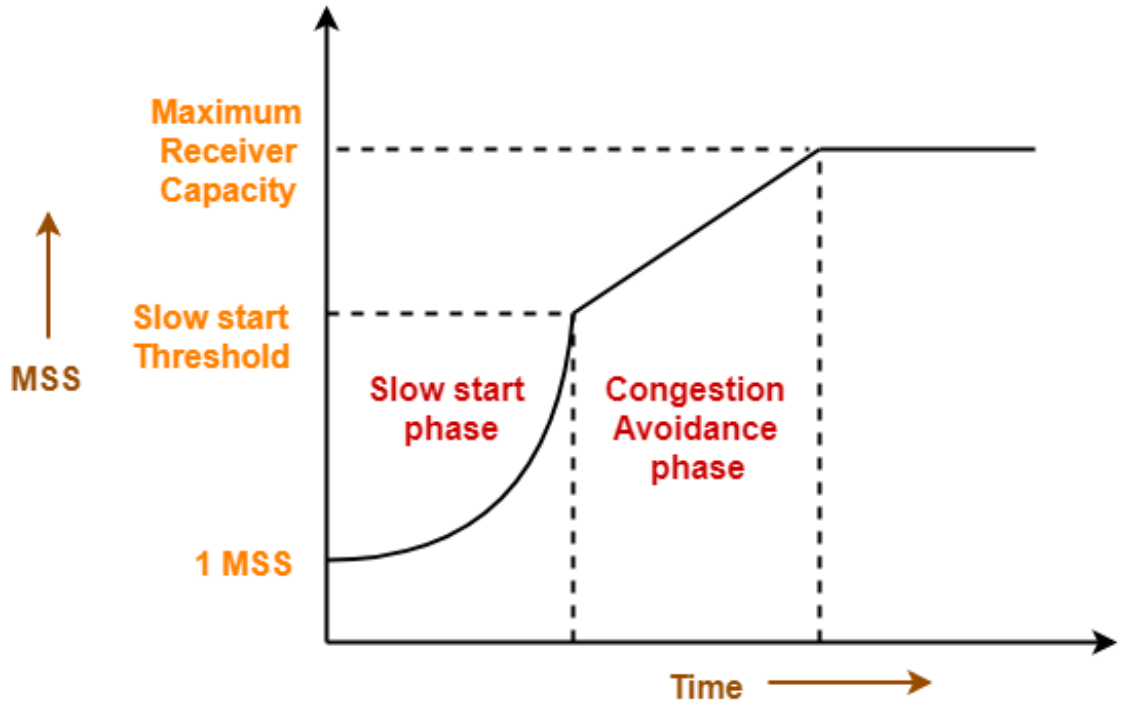
***= (Receiver window size / Maximum Segment Size) / 2***

## Congestion Avoidance Phase-

- After reaching the threshold, sender increases the congestion window size linearly to avoid the congestion.
- On receiving each acknowledgement, sender increments the congestion window size by 1.
- The followed formula is-

***Congestion window size = Congestion window size + 1***

- This phase continues until the congestion window size becomes equal to the receiver window size.



## Congestion Detection Phase-

When sender detects the loss of segments, it reacts in different ways depending on how the loss is detected-

### Case-01: Detection on Time Out-

- Time Out Timer expires before receiving the acknowledgement for a segment. This case suggests the stronger possibility of congestion in the network. There are chances that a segment has been dropped in the network.

### Reaction-

In this case, sender reacts by-

- Setting the slow start threshold to half of the current congestion window size.
- Decreasing the congestion window size to 1 MSS.
- Resuming the slow start phase.

### Case-02: Detection on Receiving 3 Duplicate Acknowledgements-

- Sender receives 3 duplicate acknowledgements for a segment. This case suggests the weaker possibility of congestion in the network. There are chances that a segment has been dropped but few segments sent later may have reached.

### Reaction-

In this case, sender reacts by-

- Setting slow start threshold to half of the current congestion window size.
- Decreasing congestion window size to slow start threshold.
- Resuming congestion avoidance phase.

# Application layer

- A layer where all the applications are found.
- The layers below the application layer are there to provide transport services, but they don't do real work for users.
- In the application layer there is a need for support protocols, to allow applications to function.
- Performs common application service for application processes
- Software programs are written in application layer to handle different terminal types that exist & map virtual terminal software onto real terminal.
- It contains a variety of protocols & is concerned with file transfer as well as electronic mail, remote job entry & various other services of general interest.
- This layer contains a variety of protocols that are commonly needed.
- Another application of this layer is file transfer. Different file system has different file naming conventions, different ways of representing text lines & so on.

# File Transfer Protocol (FTP)

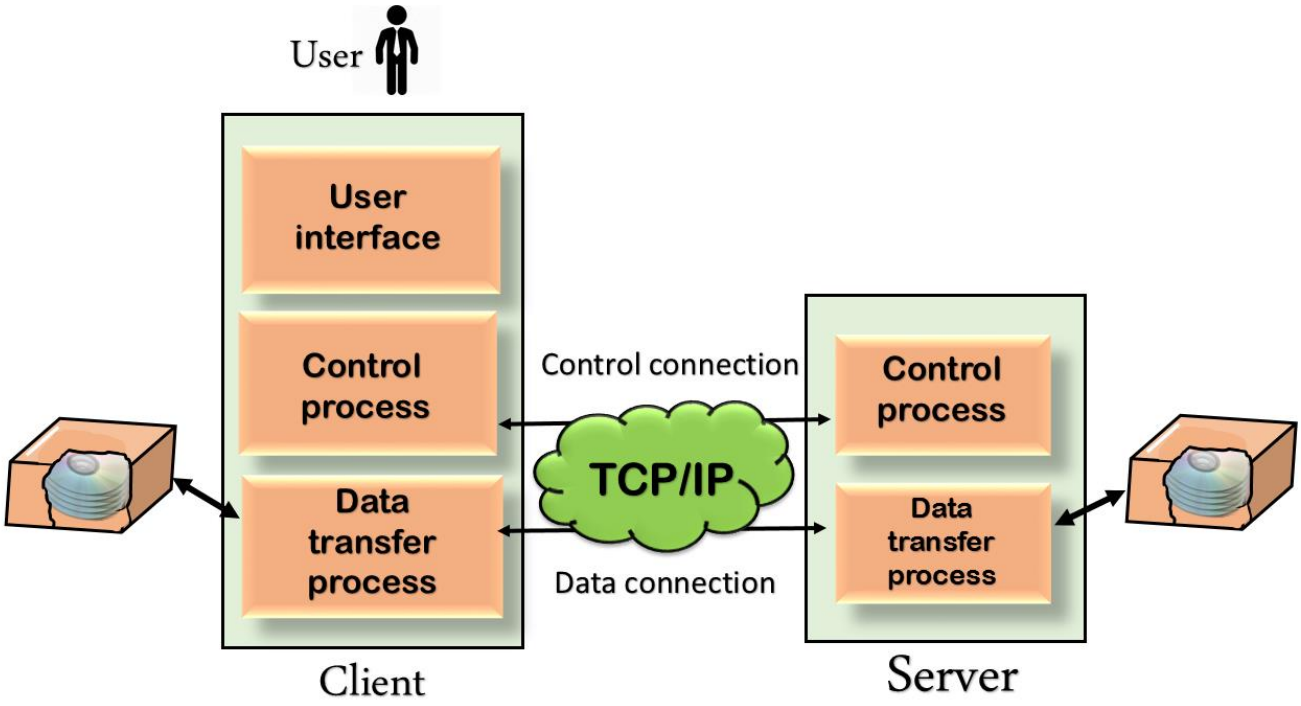
- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used to transmit files from one host to another.
- It is mainly used to transfer web page files from their creator to the computer that acts as a server for other computers on internet.
- It is also used to download the files to computer from other servers.

# Objectives of FTP

- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

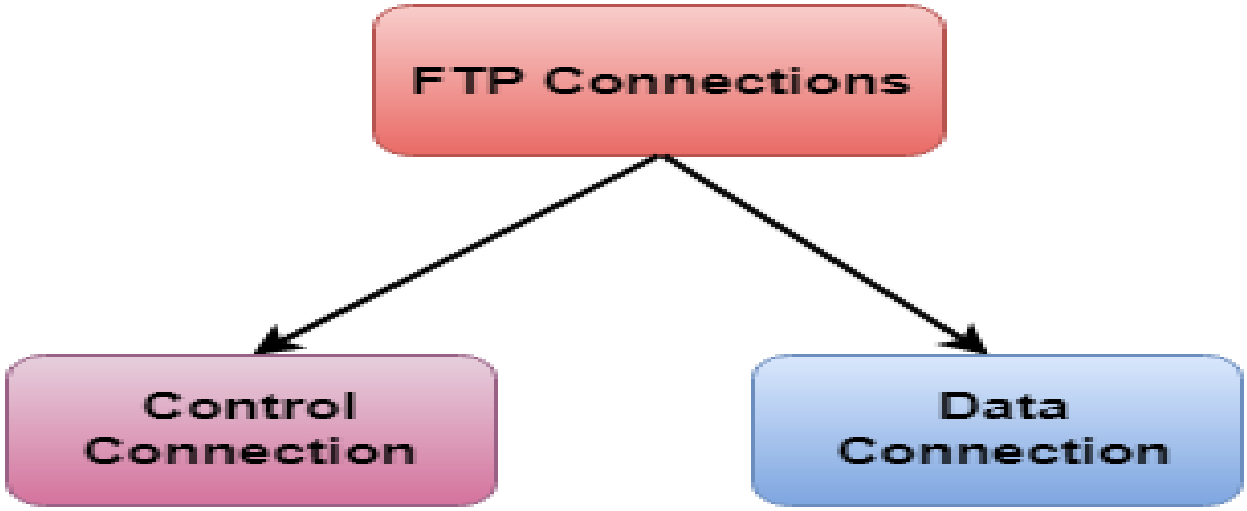
# Why FTP?

- Although transferring files from one system to another is very simple & straightforward, sometimes it can cause problems.
- For example, two systems may have different file conventions.
- Two systems may have different ways to represent text & data.
- Two systems may have different directory structures.
- FTP protocol overcomes these problems by establishing two connections between hosts.
- One connection is used for data transfer, & another connection is used for control connection.




- The above figure shows basic model of FTP.
- The **FTP client** has **three components**:
  - User interface
  - Control process
  - Data transfer process
- The **server** has **two components**:
  - Server control process
  - Server data transfer process

There are two types of connections in FTP:




- **Control Connection:**
  - The control connection uses very simple rules for communication.
  - Through control connection, we can transfer a line of command or line of response at a time.
  - The control connection is made between the control processes.
  - The control connection remains connected during the entire interactive FTP session.
- **Data Connection:**
  - The Data Connection uses very complex rules as data types may vary.
  - The data connection is made between data transfer processes.
  - The data connection opens when a command comes for transferring the files & closes when the file is transferred.

## FTP Clients

- FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.
- It allows a user to connect to a remote host & upload or download files.
- It has a set of commands that we can use to connect to a host, transfer the files between you & your host & close the connection.
- The FTP program is also available as a built-in component in a Web browser.

- This GUI based FTP client makes the file transfer very easy & also doesn’t require to remember FTP commands.

## Advantages of FTP:

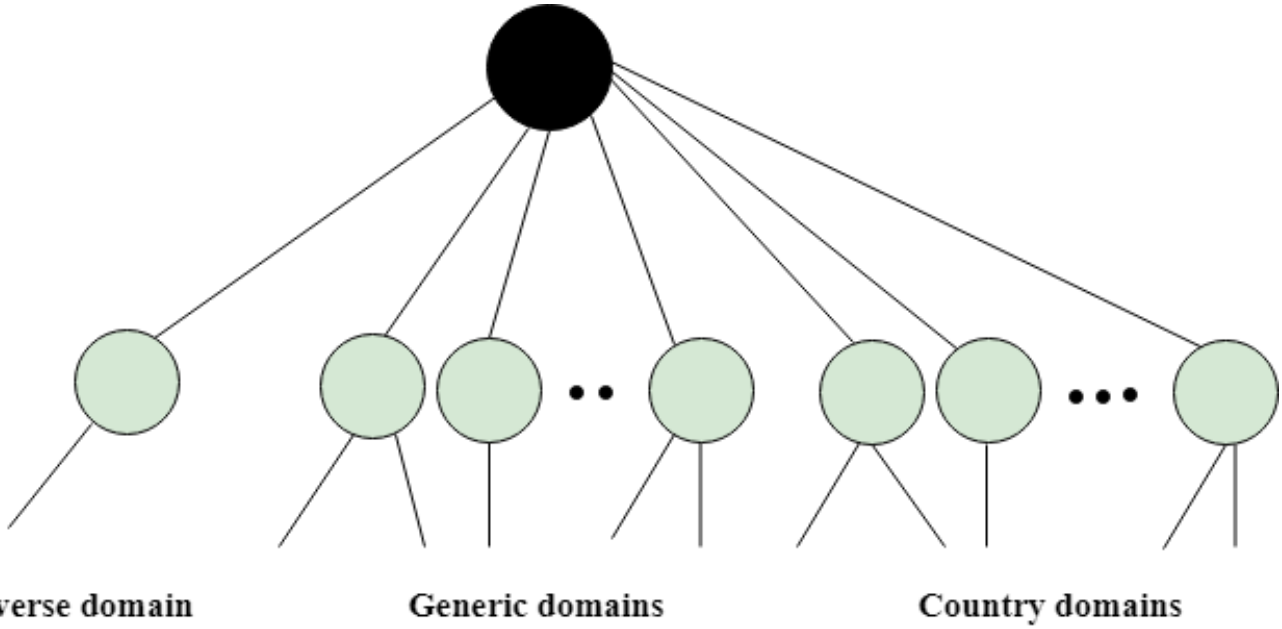
- **Speed:**
  - One of the biggest advantages of FTP is speed.
  - The FTP is one of the fastest ways to transfer files from one computer to another computer.
- **Efficient:**
  - It is more efficient as we don’t need to complete all the operations to get the entire file.
- **Security:**
  - To access the FTP server, we need to login with the username & password.
  - Therefore, we can say that FTP is more secure.
- **Back & forth movement:**
  - FTP allows us to transfer the files back & forth.

## Disadvantages of FTP:

- The standard requirement of industry is that all FTP transmissions should be encrypted.
  - But not all the FTP providers are equal & not all the providers offer encryption. So, **we have to look out for FTP providers that provides encryption.**
- FTP serves two operations, i.e., to send & receive large files on a network.
  - But, the **size limit** of the file is 2GB that can be sent.
  - It also doesn't allow you to run simultaneous transfers to multiple receivers.
- Passwords & file contents are sent in clear text that **allows unwanted eavesdropping**.
  - So, it is quite possible that attackers can carry out the brute force attack by trying to guess FTP password.
- It is **not compatible with every system**.

# Domain Name System (DNS)

- An application layer protocol that defines how the application processes running on different systems, pass the messages to each other.
- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network & its numerical address.
- DNS is required for the functioning of internet.
- Each node in a tree has a domain name, & a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates domain name into IP addresses. This allows users of networks to utilize user-friendly names when looking for other hosts instead of remembering IP addresses.
- For example, suppose FTP site at Edu Soft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.
- DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections:
    - Generic domains
    - Country domains
    - Inverse domain

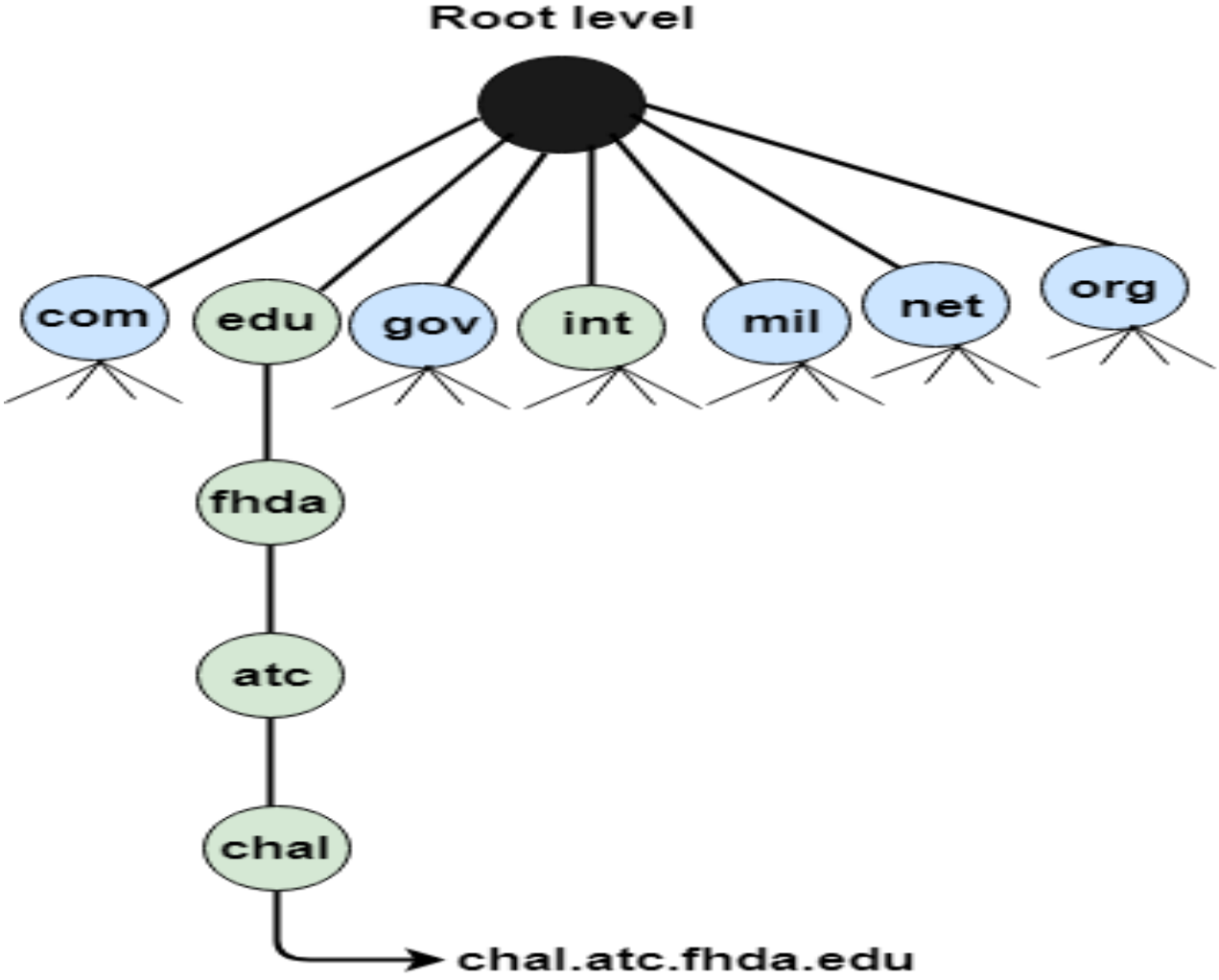



## Generic Domains

- It defines the registered hosts according to their generic behaviour.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, & these labels describe the organization type.

| Label | Description |
|---|---|
| aero | Airlines & aerospace companies |
| biz | Businesses or firms |
| com | Commercial Organizations |
| coop | Cooperative business Organizations |
| edu | Educational institutions |
| gov | Government institutions |
| info | Information service providers |
| int | International Organizations |

| | |
|---|---|
| mil | Military groups |
| museum | Museum & other non-profit organizations |
| name | Personal names |
| net | Network Support centres |
| org | Non-profit Organizations |
| pro | Professional individual Organizations |




## Country Domain

- Format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three-character organizational abbreviations.

## Inverse Domain

- The inverse domain is used for mapping an address to a name.
- When the server has received a request from the client, & the server contains the files of only authorized clients.
- To determine whether the client is on the authorized list or not, it sends a query to the DNS server & ask for mapping an address to the name.

## Working of DNS

- DNS is a client/server network communication protocol.
- DNS clients send requests to the server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a ***forward DNS lookup*** while requests containing an IP address which is converted into a name known as ***reverse DNS lookups***.
- DNS implements a distributed database to store name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as DNS resolver sends a request to the DNS server to obtain the IP address of a hostname.
- If DNS server doesn't contain the IP address associated with a hostname, then it forwards the request to another DNS server.
- If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

# Electronic Mail

- Earliest & most common application of Internet is electronic mail or email.
- Email is a store-&-forward application. This means a message can be sent to someone not currently connected to the Internet. The message remain in the system until the recipient retrieves it.
- Email application allows a user to send messages over a private network or global Internet. Email supports:
    - Sending a single message to one or more recipients.
    - Sending messages that include text, voice, video, or graphics.
    - Organization of message-based criteria such as priority
- Email can be compared & contrasted with regular mail, which is often referred to as ***snail mail***.

## Advantages

Email has several advantages over snail mail:

- **It is faster**
    - In normal traffic hours, an email message can only take a few seconds or minutes to reach the destination if the recipient is in the same country or region; it may take a few hours to reach an overseas recipient.
    - Snail mail usually takes a couple of days, if not a week or two, to reach its destination.
- **Easier to distribute to a group of recipients**
    - Sender creates a distribution list that contains email addresses of all recipients. The list is given a group name.
    - Email is sent using group name as the virtual recipient. The email is delivered to all recipients on the list.
    - In other words, a group name can be used instead of a single recipient. For example, if your clients are categorized according to geographical location, you might have group names such as clients_east, clients_west & so on.
    - With snail mail, separate envelopes must be sent to each individual in the group.
- **It is less expensive**
    - Sending an email is essentially free today if we ignore the cost of being connected to the Internet, which can be justified for other purposes.
    - Sending letters via post office or other couriers is actually more costly than just the price of postage. To find the true cost of sending a letter,

we need to add the cost of office supplies (paper, etc.), the cost of typing the letter, & the cost of delivering the letter to the post office.

- **It can be less time-consuming**
    - The first emails were friendly, informal messages. This tradition has found its way into business today. Email exchanged between organizations tend to be less formal than letters. This means less time & effort on both sides to accomplish the same task.

## Disadvantages

We must not ignore some problems associated with sending of email.

- **An email cannot be certified**
    - Although the sender can check to see if the receiver has received the mail, it can't be used as legal proof. If we need a signature from the recipient, we still need to use the services of snail mail or some other courier service.
- **The privacy of email is still an open question**
    - Although several software packages are on market to make email confidential, it can't be guaranteed unless everyone uses one of these packages.
- **Email messaging is subject to abuse**
    - Unwanted & unsolicited email is a nuisance & can fill mailboxes much like junk mail. In addition, there is the threat of viruses & other potentially damaging code attached to email.

## Architecture & services

- The sending of electronic mail in Internet requires these components:
    - User agents (UAs)
    - Mail transfer agents (MTAs)
    - Protocol that controls mail delivery
- **User agent**:
    - A user agent controls composing, reading, forwarding, replying, & saving of email messages.
    - The user agent is not responsible for sending or receiving email.
- **Mail Transfer Agent (MTA)**:
    - The actual mail transfer requires a mail transfer agent (MTA).
    - To send mail, a system must have a ***client MTA***, & to receive mail, a system must have a ***server MTA***.
    - The ***client MTA*** is installed on the user's computer.

- The ***client*** & the ***server MTA*** are installed on a computer that is used as the mail server.

## Typically, **e-mail systems** support **five basic functions**:

- **Composition**
  - Refers to the process of creating messages & answers
  - Although any text editor can be used for the body of the message, the system itself can provide assistance with addressing & numerous header fields attached to each message.
  - For example, when answering a message, e-mail system can extract the originator's address from the incoming e-mail & automatically insert it into the proper place in the reply.
- **Transfer**
  - Refers to moving messages from the originator to the recipient
  - In large part, this requires establishing a connection to the destination or some intermediate machine, outputting the message, & releasing the connection.
  - The e-mail system should do this automatically, without bothering the user.
- **Reporting**
  - Has to do with telling the originator what happened to the message: ***"Was it delivered? Was it rejected? Was it lost?"***
  - Numerous applications exist in which confirmation of delivery is important & may even have legal significance
- **Displaying**
  - Displaying incoming messages is needed so people can read their e-mail.
  - Sometimes conversion is required or a special viewer must be invoked, for example, if the message is a PostScript file or digitized voice.
  - Simple conversions & formatting are sometimes attempted as well.
- **Disposition**
  - Final step
  - Concerns what the recipient does with the message after receiving it.
  - Possibilities include throwing it away before reading, throwing it away after reading, saving it, & so on.
  - It should also be possible to retrieve & reread saved messages, forward them, or process them in other ways.

- Most systems allow users to create ***mailbox*es** to store incoming e-mail. Commands are needed to create & destroy mailboxes, inspect the contents of mailboxes, insert & delete messages from mailboxes, & so on.
- ***Mailing list***, is a list of e-mail addresses. When a message is sent to the ***mailing list***, identical copies are delivered to everyone on the list.
- Other advanced features are:
    - Carbon copies
    - Blind carbon copies
    - High-priority e-mail
    - Secret (i.e., encrypted) e-mail
    - Alternative recipients if the primary one is not currently available
    - Ability for secretaries to read & answer their bosses' e-mail.
- A key idea in e-mail systems is the distinction between the **envelope** & its contents.
    - The **envelope** encapsulates the message.
    - It contains all the information needed for transporting the message, such as the destination address, priority, & security level, all of which are distinct from the message itself.
    - The ***message transport agents*** use the envelope for routing, just as the post office does.
- The message inside the envelope consists of two parts:
    - the **header**
    - the **body**
- The ***header contains control information for the user agents***.
- The ***body is entirely for the human recipient***.

# Message Formats

### *RFC 822*

- Messages consist of:
    - a primitive envelope (described in RFC 821)
    - some number of header fields
    - a blank line
    - message body
- Each header field (logically) consists of
    - a single line of ASCII text containing the field name
    - a colon; &
    - for most fields, a value.

RFC 822 was designed decades ago & doesn't clearly distinguish the envelope fields from the header fields.

In normal usage, the user agent builds a message & passes it to the message transfer agent, which then uses some of the header fields to construct the actual envelope. The principal header fields related to message transport are

- The ***To:*** **field**
  - Gives the DNS address of the primary recipient.
  - Having multiple recipients is also allowed.
- **The *Cc:* field**
  - Gives the addresses of any secondary recipients.
  - In terms of delivery, there is no distinction between primary & secondary recipients.
  - It is entirely a psychological difference that may be important to the people involved but is not important to the mail system.
- **The term *Cc:* (Carbon copy)**
  - Email addresses of secondary recipient
- **The *Bcc:* (Blind carbon copy) field**
  - Same like the *Cc:* field, except that this line is deleted from all the copies sent to the primary & secondary recipients.
  - This feature allows people to send copies to third parties without primary & secondary recipients knowing this.
- The next two fields, ***From:*** & ***Sender:***
  - tell who wrote & sent the message, respectively.
- **A line containing Received:**
  - Added by each ***message transfer agent*** along the way.
  - The line contains
    - the agent's identity
    - the date & time the message was received
    - other information that can be used to find bugs in the routing system.
- **The Return-Path: field**
  - Added by the final ***message transfer agent***
  - Intended to tell how to get back to the sender

## **Some fields used in the RFC 822 message header**:

| Header | Meaning |
|---|---|
| Date: | The date and time the message was sent |
| Reply-To: | E-mail address to which replies should be sent |
| Message-Id: | Unique number for referencing this message later |
| In-Reply-To: | Message-Id of the message to which this is a reply |
| References: | Other relevant Message-Ids |
| Keywords: | User-chosen keywords |
| Subject: | Short summary of the message for the one-line display |

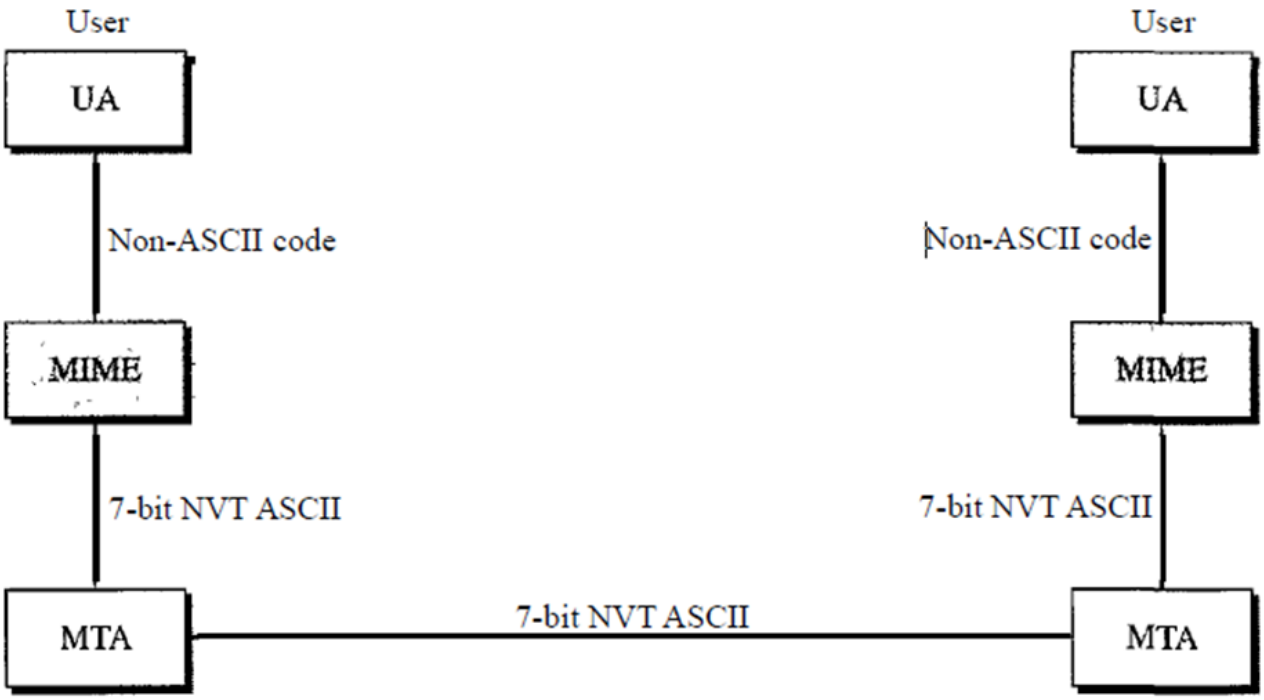
## Multipurpose Internet Mail Extension (MIME)

In 1990s, worldwide use of Internet & demand to send richer content through the mail system meant that this approach was no longer adequate. The problems include sending & receiving

1. Messages in languages with accents (e.g., French & German).
2. Messages in non-Latin alphabets (e.g., Hebrew & Russian).
3. Messages in languages without alphabets (e.g., Chinese & Japanese).
4. Messages not containing text at all (e.g., audio or images).

A solution was proposed in RFC 1341 & updated in RFCs 2045–2049. This solution, called ***MIME (Multipurpose Internet Mail Extensions)*** is now widely used.

- The basic idea of MIME is to continue to use RFC 822 format, but to add structure to the message body & define encoding rules for non-ASCII messages.
    - By not deviating from RFC 822, MIME messages can be sent using the existing mail programs & protocols.
    - All that has to be changed are the sending & receiving programs, which users can do for themselves.




- MIME defines five new message headers, as shown in the figure:

| Header | Meaning |
|---|---|
| MIME-Version: | Identifies the MIME version |
| Content-Description: | Human-readable string telling what is in the message |
| Content-Id: | Unique identifier |
| Content-Transfer-Encoding: | How the body is wrapped for transmission |
| Content-Type: | Type and format of the content |

- The first of these simply tells the user agent receiving the message that it is dealing with a MIME message, & which ***version*** of MIME it uses.
    - ***MIME-Version*** tells the user agent receiving the message that it is dealing with a MIME message, & which version of MIME it uses.
    - Any message not containing a ***MIME-Version:*** header is assumed to be an English plaintext message & is processed as such.
- The ***Content-Description:*** header is an ASCII string telling what is in the message.

    - This header is needed so the recipient will know whether it is worth decoding & reading the message.
- The ***Content-Id:*** header identifies the content.
    - It uses the same format as the standard ***Message-Id:*** header.
- The ***Content-Transfer-Encoding:*** tells how the body is wrapped for transmission through a network that may object to most characters other than letters, numbers, & punctuation marks. Five schemes (plus an escape to new schemes) are provided.
    - The simplest scheme is just ***ASCII text***. ASCII characters use 7 bits & can be carried directly by the e-mail protocol provided that no line exceeds 1000 characters.
    - The next simplest scheme is the same thing, but ***using 8-bit characters***, that is, all values from 0 up to & including 255.
    - Another encoding is ***binary encoding***.
        - These are arbitrary binary files that not only use all 8 bits but also don't even respect the 1000-character line limit.
        - Executable programs fall into this category. No guarantee is given that messages in binary will arrive correctly.
    - The correct way to encode binary messages is to use ***base64 encoding***, sometimes called ***ASCII armor***. In this scheme, groups of 24 bits are broken up into four 6-bit units, with each unit being sent as a legal ASCII character.
        - The coding is "A" for 0, "B" for 1, & so on, followed by the 26 lower-case letters, the ten digits, & finally + & / for 62 & 63, respectively.
        - The == & = sequences indicate that the last group contained only 8 or 16 bits, respectively.
        - Carriage returns & line feeds are ignored, so they can be inserted at will to keep the lines short enough.
        - Arbitrary binary text can be sent safely using this scheme.
    - For messages that are almost entirely ASCII but with a few non-ASCII characters, *base64 encoding* is somewhat

inefficient. Instead, an encoding known as ***quoted-printable encoding*** is used.

- This is just 7-bit ASCII, with all the characters above 127 encoded as an equal sign followed by the character's value as two hexadecimal digits.
- Binary data should be sent encoded in base64 or quoted-printable form.

  - When there are valid reasons not to use one of these schemes, it is possible to specify a ***user-defined encoding*** in the ***Content-Transfer-Encoding:*** header.

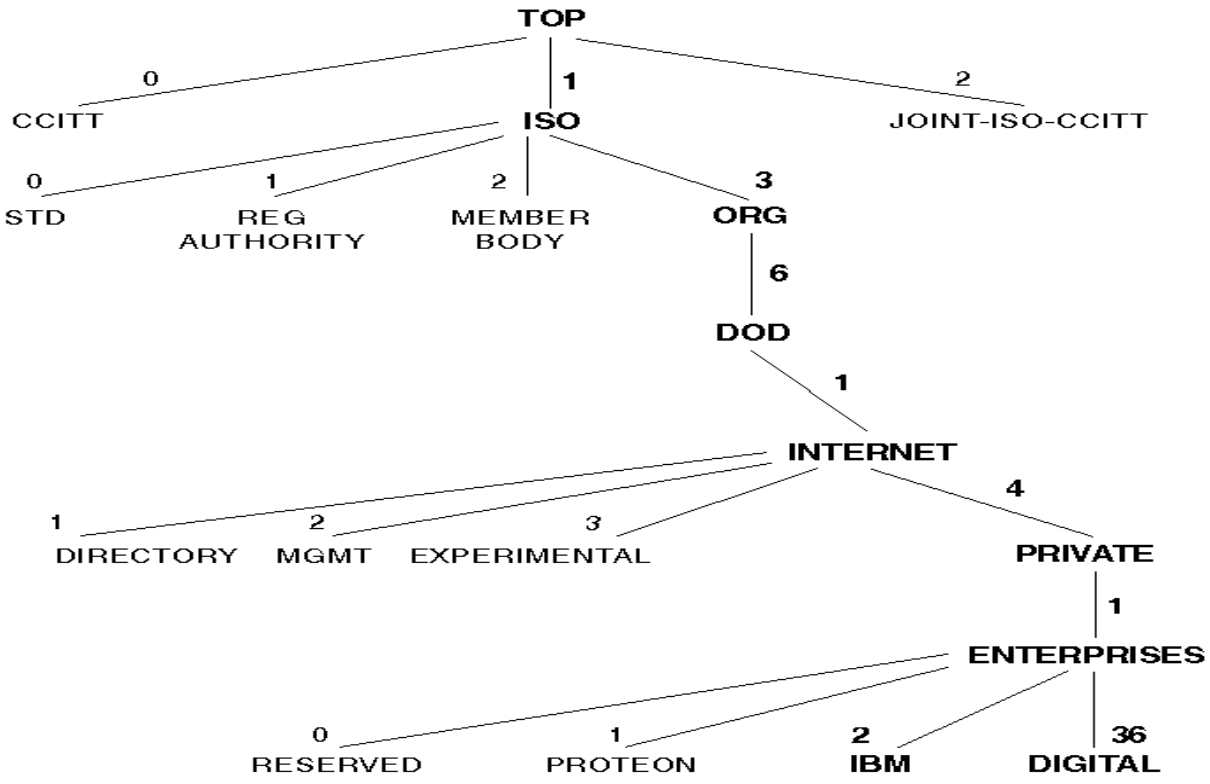
- The next header ***Content-Type*** specifies the nature of the message body.
  - Seven types are defined in RFC 2045, each of which has one or more subtypes. The type & subtype are separated by a slash, as in Content-Type: video/mpeg
  - The subtype must be given explicitly in the header; no defaults are provided. The initial list of types & subtypes specified in RFC 2045 is given in following table:

| Type | Subtype | Description |
|---|---|---|
| Text | Plain | Unformatted text |
| | Enriched | Text including simple formatting commands |
| Image | Gif | Still picture in GIF format |
| | Jpeg | Still picture in JPEG format |
| Audio | Basic | Audible sound |
| Video | Mpeg | Movie in MPEG format |
| Application | Octet-stream | An uninterpreted byte sequence |
| | Postscript | A printable document in PostScript |
| Message | Rfc822 | A MIME RFC 822 message |
| | Partial | Message has been split for transmission |
| | External-body | Message itself must be fetched over the net |
| Multipart | Mixed | Independent parts in the specified order |
| | Alternative | Same message in different formats |
| | Parallel | Parts must be viewed simultaneously |
| | Digest | Each part is a complete RFC 822 message |

# Simple Network Management Protocol (SNMP)

- A large network can often get into various kinds of trouble due to routers (dropping too many packets), hosts (going down) etc. One has to keep track of all these occurrence & adapt to such situations. A protocol has been defined.
- Under this scheme all entities in the network belong to 4 class:
  - ***Managed Nodes***
  - ***Management Stations***
  - ***Management Information*** (called ***Object***)
  - ***A Management Protocol***
- The ***managed nodes*** can be hosts, routers, bridges, printers or any other device capable of communicating status information to others.
  - To be managed directly by SNMP, a node must be capable of running an SNMP management process, called ***SNMP agent***.

- Network management is done by management stations by exchanging information with the nodes. These are basically general-purpose computers running special management software.
- The ***management stations*** poll the stations periodically.
- Since SNMP uses unreliable service of UDP, the polling is essential to keep in touch with the nodes.
- Often the nodes send a trap message indicating that it is going to go down. The management stations then periodically check (with an increased frequency). This type of polling is called ***trap directed polling***.
- Often a group of nodes are represented by a single node which communicates with the management stations. This type of node is called ***proxy agent***. The ***proxy agent*** can also serve as a ***security arrangement***.
- All the variables in this scheme are called ***Objects***.
- Each variable can be referenced by a specific addressing scheme adopted by this system. The entire collection of all ***objects*** is called ***Management Information Base (MIB).***
- The addressing is hierarchical as seen in the picture.




- Internet is addressed as ***1.3.61***.

- All the ***objects*** under this domain have this string at the beginning.
- The information is exchanged in a standard & vendor-neutral way.
- All the data are represented in ***Abstract Syntax Notation 1 (ASN.1).***
- It is similar to XDR (External Data Representation) as in RPC but it has widely different representation scheme.
- A part of it actually adopted in SNMP & modified to form the Structure of Information Base.

- The ***Protocol*** specifies various kinds of messages that can be exchanged between the ***managed nodes*** & the ***management station***.

| **Message** | **Description** |
|---|---|
| ***1. Get Request*** | Request the value for a variable |
| ***2. Get Response*** | Returns the value of the variable asked for |
| ***3. Get_Next_Request*** | Request a variable next to the previous one |
| ***4. Set Request*** | Set the value of an Object. |
| ***5. Trap*** | Agent to manager Trap report |
| ***6. Get_bulk_request*** | Request a set of variables of same type |
| ***7. Inform_Request*** | Exchange of MIB among Management stations |

- The last two options have been actually added in the **SNMPv2**.
- The 4th option needs some kind of authentication from ***management station***.

## Addressing Example:

- Following is an example of the kind of address one can refer to, when fetching a value in the table: -

(20) IP-Addr-Table = Sequence of IPAddr-Entry (1)

IPAddrEntry = SEQUENCE {

IPADDENTRYADDR : IPADDR (1)

Index : integer (2)

Netmask : IPAddr (3) }

- So, when accessing the netmask of some IP-entity the variable name would be:
*1.3.6.1.2.4.201.3. key-value*
- Here since IP address the unique key to index any member of the array, the address can be like: -
*1.3.6.1.2.4.20.1.3.128.10.2.3*

# World Wide Web

The **Web**, as the ***World Wide Web*** is popularly known, is an architectural framework for accessing linked content spread out over millions of machines all over the Internet.

In 10 years, it went from being a way to coordinate the design of high-energy physics experiments in Switzerland to the application that millions of people think of as being ***"The Internet".***

It is easy for beginners to use & provides access with a rich graphical interface to an enormous wealth of information on almost every conceivable subject, from aardvarks to Zulus.

The Web began in 1989 at CERN, the European Centre for Nuclear Research. The initial idea was to help large teams, often with members in half a dozen or more countries & time zones, collaborate using a constantly changing collection of reports, blueprints, drawings, photos, & other documents produced by experiments in particle physics.

The proposal for a web of linked documents came from CERN physicist Tim Berners-Lee. The first (text-based) prototype was operational 18 months later. A public demonstration given at the Hypertext '91 conference caught the attention of other researchers, which led Marc Andreessen at the University of Illinois to develop the first graphical browser. It was called ***Mosaic*** & released in February 1993.

***Mosaic*** was so popular that a year later Andreessen left to form a company, Netscape Communications Corp., whose goal was to develop Web software. For the next three years, Netscape Navigator & Microsoft's Internet Explorer engaged in a "browser war", each one

trying to capture a larger share of the new market by frantically adding more features (& thus more bugs) than the other one.

Through the 1990s & 2000s, ***Web sites*** & ***Web pages***, as ***Web content*** is called, grew exponentially until there were millions of sites & billions of pages. A small number of these sites became tremendously popular. Those sites & the companies behind them largely define the Web as people experience it today.

Examples include:

- A bookstore (Amazon, started in 1994, market capitalization $50 billion)
- A flea market (eBay, 1995, $30B)
- Search (Google, 1998, $150B)
- Social networking (Facebook, 2004, private company valued at more than $15B)

The period through 2000, when many Web companies became worth hundreds of millions of dollars overnight, only to go bust practically the next day when they turned out to be hype, even has a name. It is called the ***dot com era***.

New ideas are still striking it rich on the Web. Many of them come from students. For example,

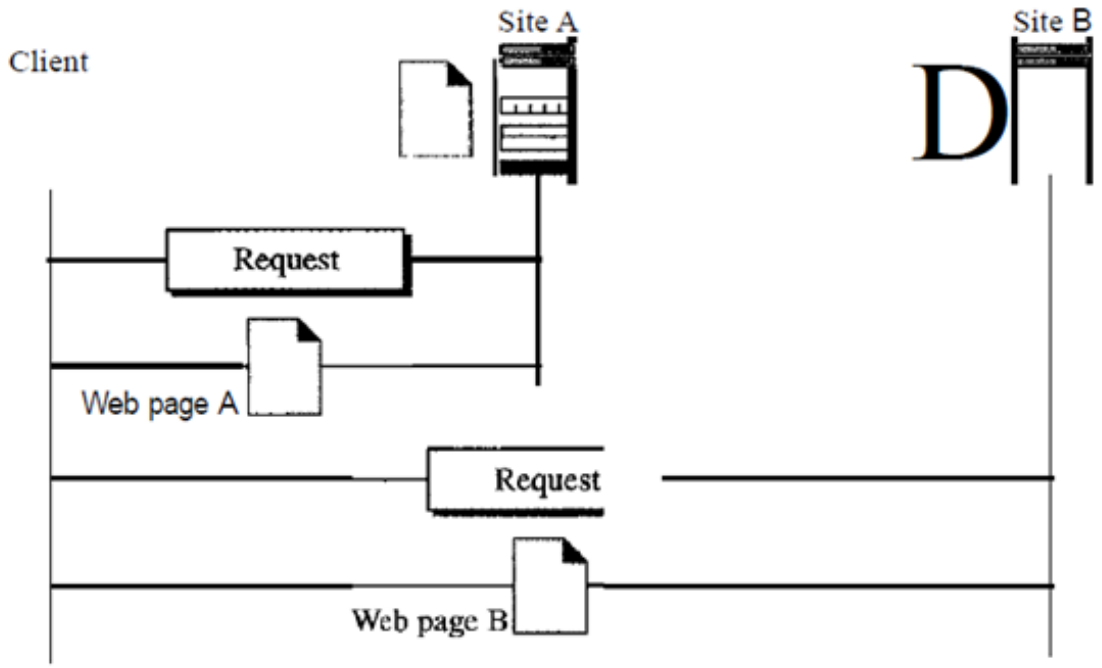
- Mark Zuckerberg was a Harvard student when he started Facebook
- Sergey Brin & Larry Page were students at Stanford when they started Google.

In 1994, CERN (European Council of Nuclear Research) & M.I.T. (Massachusetts Institute of Technology) signed an agreement setting up the ***W3C (World Wide Web Consortium),*** an organization devoted to further developing the Web, standardizing protocols, & encouraging interoperability between sites. Berners-Lee became the director. Since then, several hundred universities & companies have joined the consortium.

- **The World Wide Web (WWW) is a repository of information linked together from points all over the world.**
- **The WWW has a unique combination of flexibility, portability, & user-friendly features that distinguish it from other services provided by the Internet.**
- **The WWW project was initiated by CERN (European Laboratory for Particle Physics) to create a system to handle distributed resources necessary for scientific research.**

## Architectural overview

- The ***WWW*** today is a distributed client-server service, in which a client using a browser can access a service using a server.
- But the service provided is distributed over many locations called ***sites***, as shown in Figure

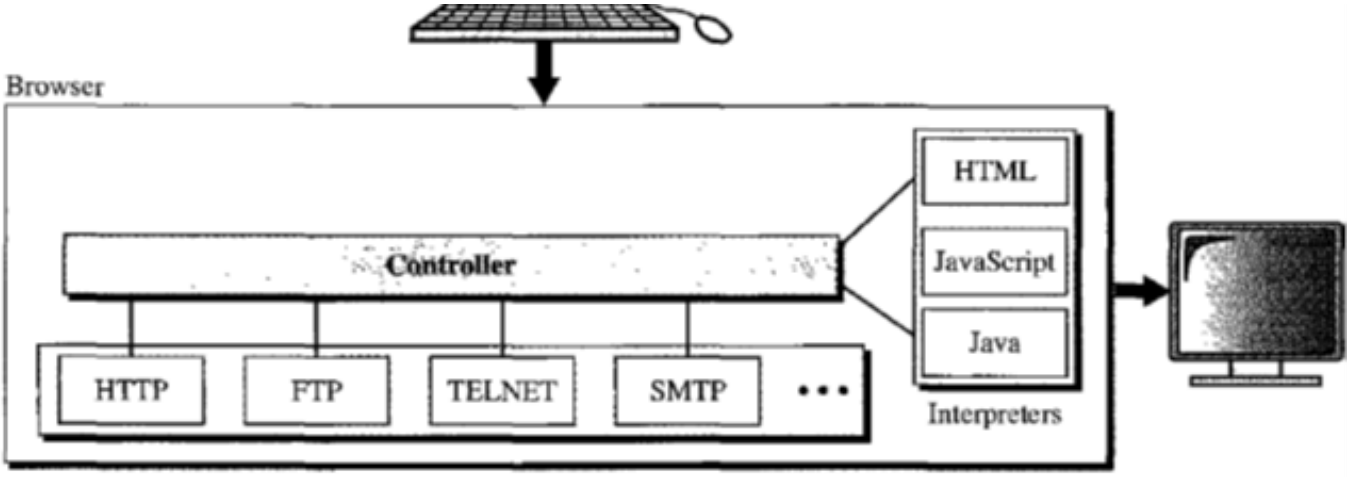



- Each site holds one or more documents, referred to as ***Web pages***.
- Each Web page can contain a link to other pages in the same site or at other sites.
- The pages can be retrieved & viewed by using browsers.

- A client needs to see some information that it knows belongs to site A.
    - It sends a request through its ***browser***, a program that is designed to fetch Web documents.
    - The ***request***, among other information, includes the ***address of the site*** & the ***Web page***, called the ***URL***.
    - The server at site A finds the document & sends it to the client.
    - When the user views the document, she finds some references to other documents, including a Web page at site B.
    - The reference has the URL for the new site. The user is also interested in seeing this document.
    - The client sends another request to the new site, & the new page is retrieved.

## Client (Browser)

- A variety of vendors offer commercial browsers that interpret & display a Web document, & all use nearly the same architecture.
- Each browser usually consists of three parts:
    - A controller
    - Client protocol
    - Interpreters
- The ***controller*** receives input from keyboard or mouse & uses client programs to access the document.
    - After document has been accessed, the controller uses one of the interpreters to display the document on the screen.
- The ***client protocol*** can be one of the protocols described previously such as FTP.
- The ***interpreter*** can be HTML, Java, or JavaScript, depending on the type of document.
- Illustration of a browser is shown below:




## Server

- The Web page is stored at the ***server***.
- Each time a ***client request*** arrives, the corresponding document is sent to the client.
- To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk.
- A server can also become more efficient through ***multithreading*** or ***multiprocessing***. In this case, a server can answer more than one request at a time.

## Uniform Resource Locator

- A client that wants to access a Web page needs address. To facilitate the access of documents distributed throughout the world, ***HTTP*** uses locators. ***The uniform resource locator (URL)*** is a standard for specifying any kind of information on the Internet.
- The URL defines four things:
    - Protocol
    - Host computer
    - Port
    - Path
- The ***protocol*** is the client/server program used to retrieve document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP.

- The ***host*** is the computer on which the information is located, although the name of the computer can be an alias.
    - Web pages are usually stored in computers, & computers are given alias names that usually begin with the characters "www".
    - This is not mandatory, but, as the host can be any name given to the computer that hosts the Web page.
- The URL can optionally contain the ***port number*** of the server. If the port is included, it is inserted between the host & the path, & it is separated from the host by a colon.
- ***Path*** is the ***pathname*** of the file where the information is located. The path can itself contain slashes that, in the UNIX operating system, separate the directories from subdirectories & files.

## Cookies

- The ***World Wide Web*** was originally designed as a stateless entity.
- A client sends a request; a server responds. Their relationship is over.
- The original design of ***WWW***, retrieving publicly available documents, exactly fits this purpose. Today the Web has other functions; some are listed here:
    - Some websites need to allow access to registered clients only.
    - Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, & pay at the end with a credit card.
    - Some websites are used as portals: the user selects the Web pages he wants to see.
    - Some websites are just advertising.
- For these purposes, the cookie mechanism was devised.

## Creation & Storage of Cookies

- The creation & storage of cookies depend on the implementation; however, the principle is the same.
    1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, & so on), a timestamp, & other information depending on the implementation.
    2. The server includes the cookie in the response that it sends to the client.
    3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.

## Using Cookies

- When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request.
- When the server receives the request, it knows that this is an old client, not a new one.
- The contents of the cookie are never read by the browser or disclosed to the user. It is a cookie made by the server & eaten by the server.

Let us see how a cookie is used for the four previously mentioned purposes:

- The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time. For any repeated access, only those clients that send the appropriate cookie are allowed.
- An electronic store (e-commerce) can use a cookie for its client shoppers.

    - When a client selects an item & inserts it into a cart, a cookie that contains information about the item, such as its number & unit price, is sent to the browser.
    - If the client selects a second item, the cookie is updated with the new selection information, & so on.
    - When the client finishes shopping & wants to check out, the last cookie is retrieved & the total charge is calculated.
- A Web portal uses the cookie in a similar way.
    - When a user selects her favourite pages, a cookie is made & sent.
    - If the site is accessed again, the cookie is sent to the server to show what the client is looking for.
- A cookie is also used by advertising agencies.
    - An advertising agency can place banner ads on some main website that is often visited by users.
    - The advertising agency supplies only a URL that gives the banner address instead of the banner itself.
    - When a user visits the main website & clicks on the icon of an advertised corporation, a request is sent to the advertising agency. The advertising agency sends the banner, a GIF file, for example, but it also includes a cookie.
    - Any future use of the banners adds to the database that profiles the Web behaviour of the user.
    - The advertising agency has compiled the interests of the user & can sell this information to other parties.

    This use of cookies has made them very controversial.

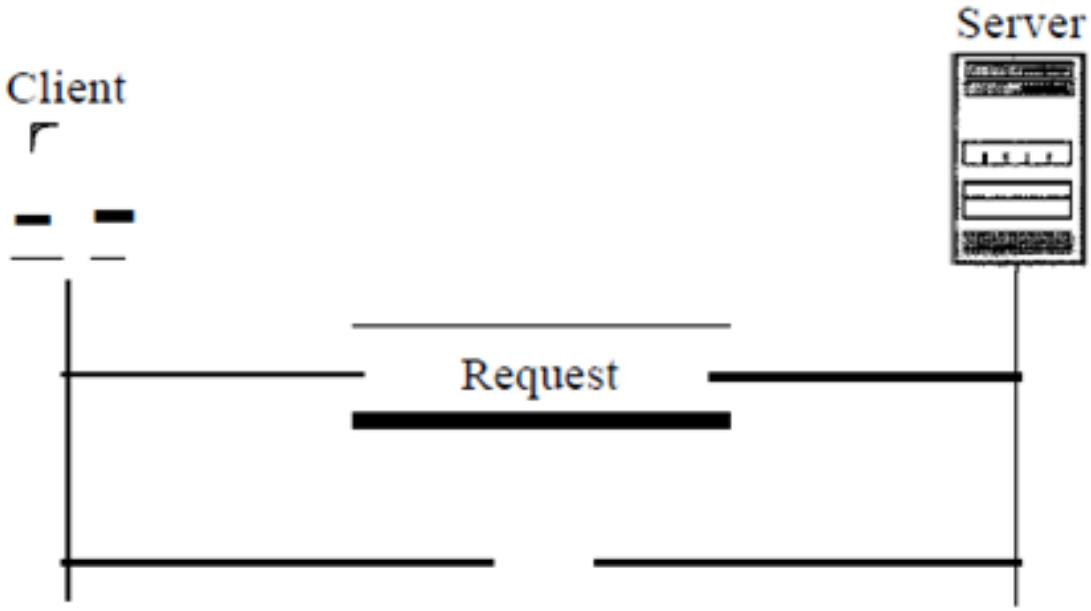
## WEB DOCUMENTS

- The documents in the WWW can be grouped into three broad categories:
    - Static
    - Dynamic

- Active
- The category is based on the time at which the contents of the document are determined.

## Static Documents

- Static documents are fixed-content documents that are created and stored in a server.
- The client can get only a copy of the document.
- In other words, the contents of the file are determined when the file is created, not when it is used.
- Of course, the contents in the server can be changed, but the user can't change them.
- When a client accesses the document, a copy of the document is sent.
- The user can then use a browsing program to display the document.




Static HTML document

## HTML

- ***Hypertext Markup Language (HTML)*** is a language to create Web pages.

- The term ***markup language*** comes from the book publishing industry. Before a book is typeset & printed, a copy editor reads the manuscript & puts marks on it. These marks tell the compositor how to format the text.
- For example, if the copy editor wants part of a line to be printed in boldface, he or she draws a wavy line under that part.
- In the same way, data for a Web page are formatted for interpretation by a browser.

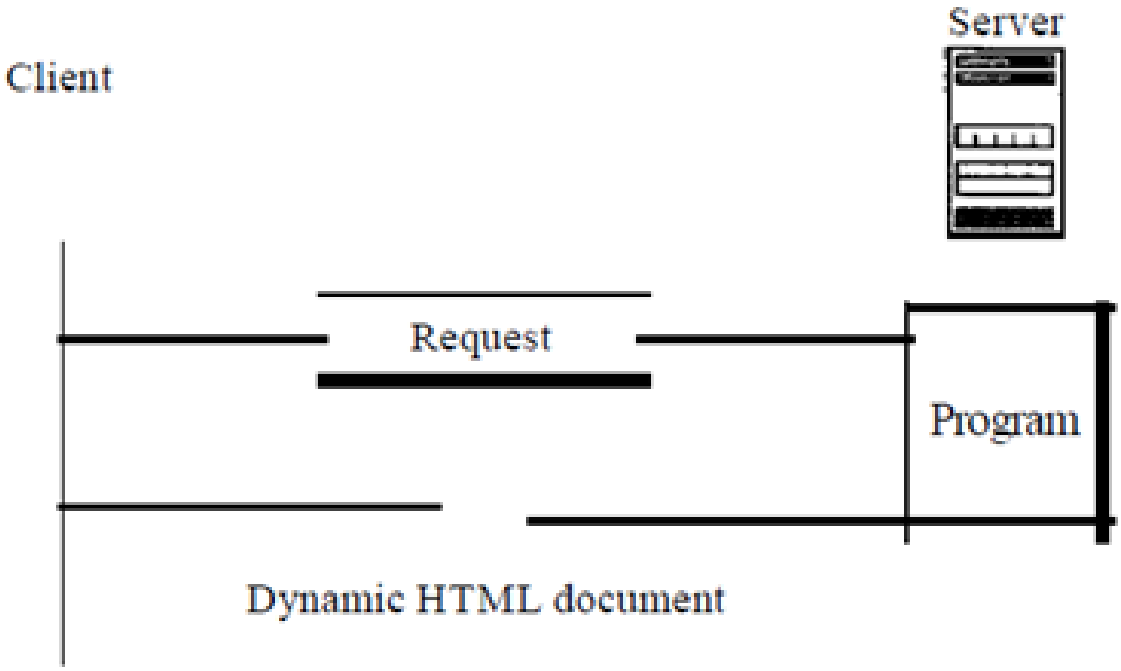
## Dynamic Documents

- A **dynamic document** is created by a Web server whenever a browser requests the document.
- When a request arrives, the Web server runs an application program or a script that creates the dynamic document.
- The server returns the output of the program or script as a response to the browser that requested the document.
- Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another.
- A very simple example of a dynamic document is the retrieval of the time & date from a server.
- Time & date are kinds of information that are dynamic in that they change from moment to moment.
- The client can ask the server to run a program such as the *date* program in UNIX & send the result of the program to the client.

## Common Gateway Interface (CGI)

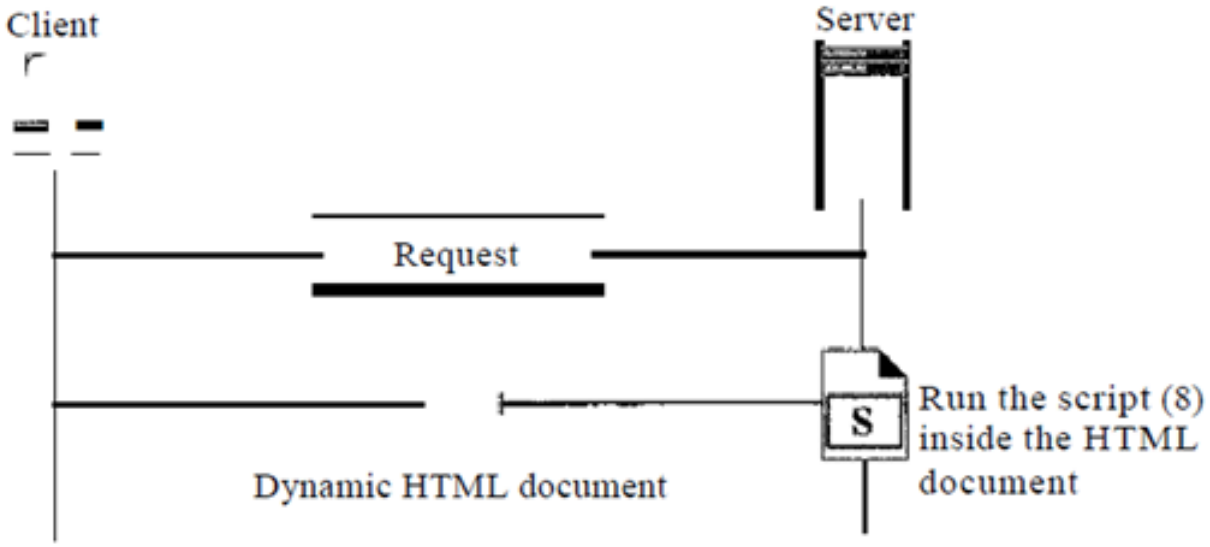
- The ***Common Gateway Interface (CGI)*** is a technology that creates & handles dynamic documents.
- ***CGI*** is a set of standards that defines how a dynamic document is written, how data are input to the program, & how the output result is used.

- Dynamic document using CGI is illustrated below:




## Scripting Technologies for Dynamic Documents

- The problem with CGI technology is the inefficiency that results if part of the dynamic document that is to be created is fixed & not changing from request to request.
- A few technologies have been involved in creating dynamic documents using scripts.
- Among the most common are
    - ***Hypertext Preprocessor (pHP)***, which uses the Perl language
    - ***Java Server Pages (JSP),*** which uses the Java language for scripting
    - ***Active Server Pages (ASP)***, a Microsoft product which uses Visual Basic language for scripting
    - ***ColdFusion,*** which embed SQL database queries in the HTML document.
- Dynamic documents are sometimes referred to as server-site dynamic documents
- Dynamic document using server-site script is illustrated below:


Client
Server
Request
S
Run the script (8) inside the HTML document
Dynamic HTML document

## Active Documents

- For many applications, we need a program or a script to be run at the client site. These are called ***active documents***.
- For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user. The program definitely needs to be run at the client site where the animation or interaction takes place.
- When a browser requests an active document, the server sends a copy of the document or a script. The document is then run at the client (browser) site. E.g., *Java Applets:* -One way to create an active document is to use Java applets.
- An active document using Java Applet is illustrated below: