INFOSYS 322

Rahul Issar

September 2017

1 Week 5A

Multiple-access resolution

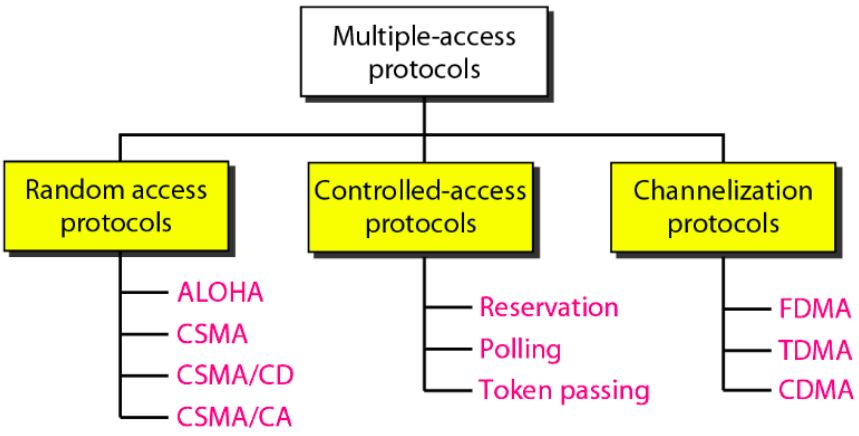


Figure 1: ShareLaTeX logo

Key issues and challenges Situation: How can a radio transmission network be used to send packetized information(data network)

Radio transmission of digital information is based on sending short bursts of data(packets)

Transmission technology makes up the first layer(physical layer) and is not concerned with reliability of information delivery.

Since radio frequencies are a scarce resource transmission may be constrained to a single channel rendering communication half-duplex

A solution was invented

ALOHA was the first digital data protocol for shared access networks that took care of at least two issues:

- How to overcome the fact that only a half-duplex channel is available.

- How to assure reliable delivery of transmitted frames (packets)

ALOHA is a data link layer protocol. Vulnerable time is the length of time in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking Tfr seconds to send.

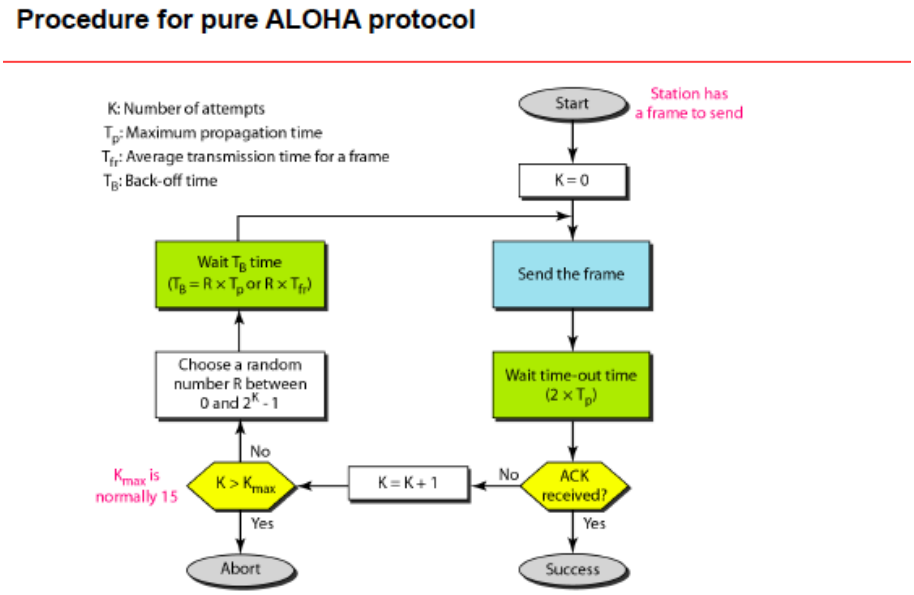


Figure 2: ShareLaTeX logo

Example The stations on a wireless ALOHA network are a maximum of 600km apart. If we assume that signals propagate at speed of light , we find

$$Tp = (600 * 10^3)/(3 * 10^8) = 2ms \tag{1}$$

Now we can find the value of TB for different values of K. a. For K = 1, the range is 0,1. The station needs to generate a random number with a value of 0 or 1. This means that TB is either 0 ms (0 * 2) or 2ms (1 * 2), based on the outcome of the random variable.

b. For K = 2, the range is 0,1,2,3. This means that TB can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.

1.1 Carrier Sense Multiple Access (CSMA)

This method is an improvement from ALOHA which is a procedure that forces the station to sense the medium before transmitting. This method later evolved into two parallel methods: Carrier sense multiple access with collision detection (CSMA/CD), which tells the station what to do when a collision is detected, and carrier sense multiple access with collision avoidance (CSMA/CA), which tries to avoid the collision.

To minimize the chance of collision and, therefore increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station sense the medium before trying to use

it. CSMA requires that each station first listen to the medium (or check the state of the medium) before sending.

Channelization

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency or through code, between different stations. In this section, we discuss three channelization protocols. These three channels are : Frequency-division multiple access (FDMA), Time-division multiple access (TDMA) Code-division multiple access (CDMA).

FDMA

In frequency- division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time. Each station also uses a bandpass filter to confine the transmitter frequencies. To prevent station interference's, the allocated bands are separated from one another by small guard bands.

TDMA

In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in its assigned time slot. The main problem with TDMA lies in achieving synchronization between the different stations. Each stations need to know the beginning of its slot and the location of its slot. This may be difficult because of propagation delays introduced in the system if the stations are spread over a large area.

CDMA

Code-division multiple access (CDMA) was conceived several decades ago. Recent advances in electronic technology have finally made its implementation possible. CDMA differs from FDMA in that only one channel occupies the entire bandwidth of the link. It differs from TDMA in that all stations can send data simultaneously; there is no timesharing. In CDMA, one channel carries all transmissions simultaneously.

Data representation

We follow these rules for encoding: If a station needs to send a 0 bit it encodes it as -1; if it needs to send a 1 bit, it encodes it as +1. When a station is idle, it sends no signal, which is interpreted as a 0.

Data bit 0 ————> − 1 Data bit 1 ————> +1 Silence ————> 0

2 Week 5B Add more

Media Access Control (MAC)

- Data encapsulation
- Frame delimiting
- Addressing

- Error detection
- Control of frame placement on and off the media
- Media recovery

3 Week 6

IPv4 addressing

There are three notations in IPv4 addressing

- Binary (1000000 00001011 00000011 00011111)
- Dotted decimal (128 . 11 . 3 . 31)
- Hexadecimal (80 0B 03 1F)

Classful addressing When the internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed - length prefixes were designed instead of one (n = 8, n = 16 and n = 24): The whole address space was divided into five classes(Class A, B, C, D, and E).

The reason that classful addressing has become obsolete is address depletion. Since the addresses were not distributed properly, the internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals needed to be connected to the internet.

Although classful addressing had several problems and became obsolete, it had one advantage: Given an address, we can easily find the class of the address and the prefix length for each class.

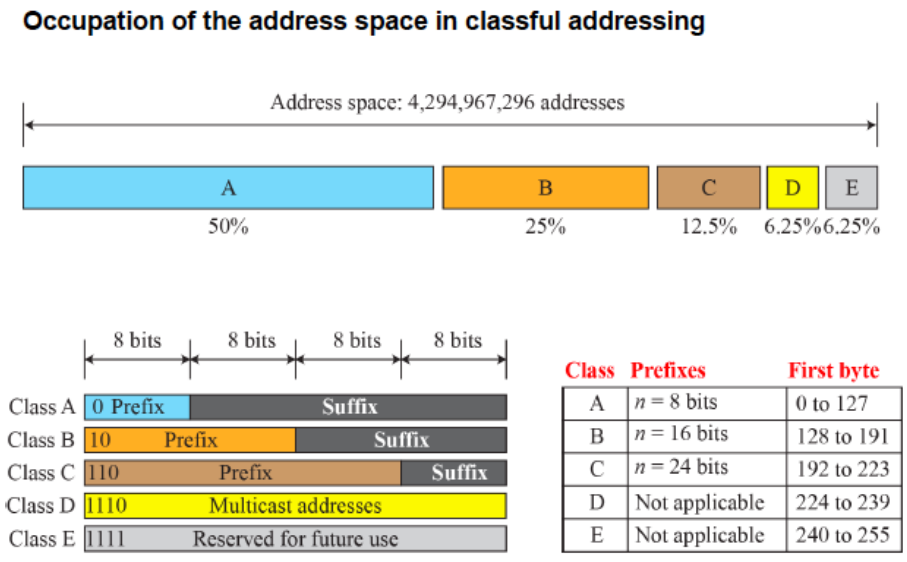


Figure 3: ShareLaTeX logo

3.1 Classless addressing

With the growth of the internet, it was clear that a larger address space was needed as a long - term solution. The larger address space, however requires that the length of IP addresses also be increased, which means the format of the IP packets need to be changed. The short-term solution still uses IPv4 addresses, but it is called IPv4 addressing. The class privilege was removed from the distribution to compensate for the address depletion.

Unlike classful addressing, the prefix length in classless addressing is variable. Example shown below of a classless address problem:

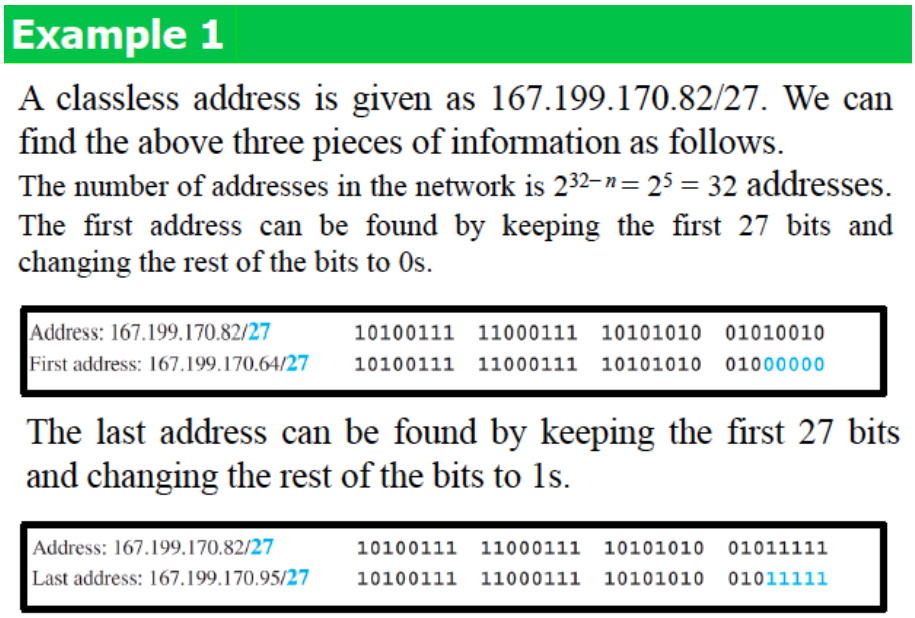


Figure 4: ShareLaTeX logo

Extracting information from an address

- The number of addresses in the block is found as N to the power of 32-n
- To find the first address, we keep the n leftmost bits and set the (32-n) rightmost bits all to 0s
- To find the last address, we keep the n leftmost bits and set the (32-n) rightmost bits all to 1s

3.2 Connection-less packet switching

When the network layer provides a connectionless service, each packet traveling in the internet is an independent entity; there is no relationship between packets belonging to the same message. Each packet is routed based on the information contained in its header: source and destination addresses. The destination address defines where it should go; the source address defines where it comes from. The router in his case routes the packet based only on the destination address. The source address may be used to send an error message to the source of the packet is discarded.

In the datagram approach, the forwarding decision is based on the destination address of the packet.

3.3 Virtual circuit approach

In a connection-oriented service, there is a relationship between all packets belonging to a message. Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After the connection setup, the datagrams can all follow the same path. In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.

Each packet is forwarded based on the **label** in the packet. To follow the idea of connection-oriented design to be used in the Internet, we assume that the packet has a label when it reaches the router.

Forwarding process

In the virtual-circuit approach, the forwarding decision is based on the label of the packet.

Sending a request packet and acknowledgement packets

A request packet is sent from the source to the destination. This auxiliary packet carries the source and destination addresses. A special packet, called the acknowledgement packet, completes the entries in the switching tables.

4 Week 7

4.1 Network Layer:Unicast Routing

In unicast routing, a packet is routed, hop by hop, from its source to its destination by the help of forwarding tables. The source host needs no forwarding table because it delivers its packet to the default router in its local network. The destination host needs no forwarding table either because it receives the packet from its default router in its local network. This means that only the routers that glue together the networks in the internet need forwarding tables.

4.2 Least-Cost Routing

When an internet is modeled as a weighted graph, one of the ways to interpret the best route from the source router to the destination router is to find the least cost between the two. In other words, the source router chooses a route to the destination router in such a way that the total cost for the route is the least cost among all possible routes. **Least-Cost Trees**

If there are N routers in an internet, there are (N-1) least-cost paths from each router to any other router. This means we need N x (N -1) least-cost paths for the whole internet. If we have only 10 routers in an internet, we need 90 least-cost paths. A better way to see all of these paths is to combine them in a least-cost tree.

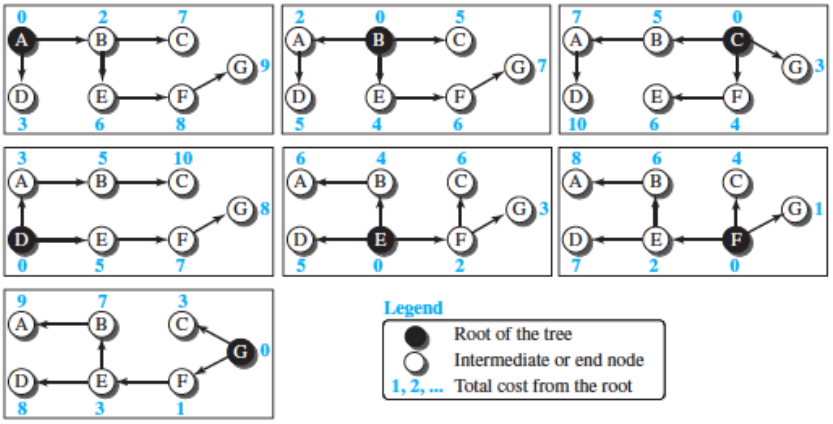


Figure 5: ShareLaTeX logo

A least-cost tree (as shown above) is a tree with the source router as the root that spans the whole graph (visits all other nodes) and in which the path between the root and any other node is the shortest.

4.3 Distance Vector Routing

The distance-vector(DV) routing uses the goal we discussed in the introduction, to find the best route. In distance - vector routing, the first thing each node creates is its own least-cost tree with the rudimentary information it has about its immediate neighbours. The incomplete trees are exchanged between immediate neighbours to make the trees more and more complete and to represent the whole internet. We can say that in distance-vector routing, a router continuously tells all of its neighbours what it knows about the whole internet(although the knowledge can be incomplete).

Bellman-Ford Equation

The heart of distance-vector routing is the famous Bellman-ford equation. This equation is sued to find the least cost (shortest distance) between a source node,x, and a destination node, y, through some intermediary nodes (a, b,c . . .) when the costs between the source and intermediary nodes and the least costs between the intermediary nodes and the destination are given. The following shows the general case in which Dij is the shortest distance and Cij is the cost between nodes i and j.

4.4 Internet Structure

The internet has changed from a tree-like structure, with a single backbone, to a multi-backbone structure run by different private corporations today. Although it is difficult to give a general view of the internet today, we can say that the internet has a structure similar to what is shown in next slide.

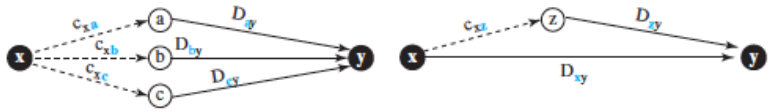
$$D_{xy} = \min\{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}), \ldots\}$$

In distance-vector routing, normally we want to update an existing least cost with a least cost through an intermediary node, such as z, if the latter is shorter. In this case, the equation becomes simpler, as shown below:

$$D_{xy} = \min\{D_{xy}, (c_{xz} + D_{zy})\}$$

Figure 20.3 shows the idea graphically for both cases.

Figure 20.3 *Graphical idea behind Bellman-Ford equation*



a. General case with three intermediate nodes b. Updating a path with a new route

Figure 6: ShareLaTeX logo

4.5 Routing Information Protocol

The Routing Information Protocol(RIP) is one of the most widely used intradomain routing protocols based on the distance-vector routing algorithm. RIP was started as part of the Xerox Network System (XNS), but it was the Berkeley Software Distribution (BSD) version of UNIX that helped make the use of RIP widespread.

Forwarding tables

Although the distance-vector algorithm is concerned with exchanging distance vectors between neighbouring nodes, the routers in an autonomous system need to keep forwarding tables to forward packets to their destination networks. A forwarding table in RIP is a three-column table in which the first column is the address of the destination network, the second column is the address of the next router to which the packet should be forwarded, and the third column is the cost (the number of hops) to reach the destination network.

5 Week 8

5.1 Simple Protocol

Our first protocol is a simple protocol with neither flow nor error control. We assume that the receiver can immediately handle any frame it receives. In other words, the receiver can never be overwhelmed

with incoming frames. The data-link layers of the sender and receiver provide transmission services for their network layers.

5.2 FSMs

The sender site should not send a frame until its network layer has a message to send. The receiver site cannot deliver a message to its network layer until a frame arrives. We can show these requirements using two FSMs. The FSMs only have one state which is the ready state. The sending machine remains in the ready state until a request comes from the process in the network layer. When this event occurs, the sending machine encapsulates the message in a frame and sends it to the receiving machine. The receiving machine remains in the ready state until a frame arrives from the sending machine. The machine then decapsulates the message once received and delivers it to the process at the network layer.

Stop-and-Wait Protocol

Our second protocol is called the Stop-and-Wait protocol, which uses both flow and error control. In this protocol, the sender sends one frame at a time and waits for an acknowledgement before sending the next one. To detect corrupted frames, we need to add a CRC to each data frame. When a frame arrives at the receiver site, it is checked. If its CRC is incorrect, the frame is corrupted and silently discarded. The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.

Note

Every time the sender sends a frame, it starts a timer. If an acknowledgement arrives before the timer expires, the timer is stopped and the sender sends the next frame(if it has one to send). If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted.

The sender is initially in the ready state, but it can move between the ready and blocking state

Ready State.

When the sender is in this state, it is only waiting for a packet from the network layer. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only tiers and sends the frame. The sender then moves to the blocking state.

Blocking State.

When the sender is in this state, three events can occur:

- If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.
- If a corrupted ACK arrives, it is discarded
- If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

Receiver - The receiver is always in the ready state. Two events may occur:

- If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent.
- If a corrupted frame arrives, the frame is discarded.

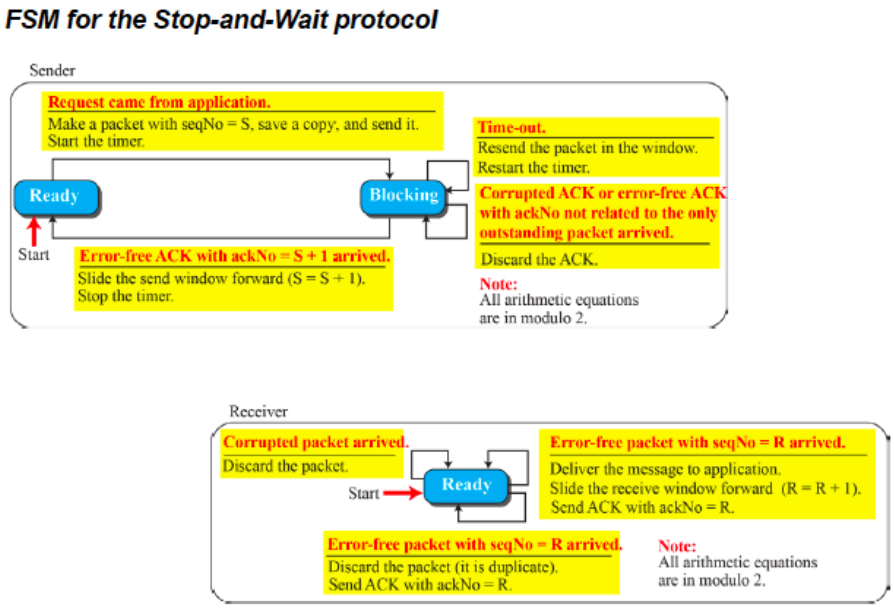


Figure 7: ShareLaTeX logo

5.3 Transport Layer: Process-to-Process delivery

The transport layer is responsible for process-to-process delivery- the delivery of a packet, part of a message, from one process to another. Two processes communicate in a client/server relationship. Communication is provided using a logical connection.

The **Logical connection at the transport layer** is also end-to-end. The transport layer at the source host gets the message from the application layer, encapsulates it in a transport-layer packet called a segment. It then sends it through the logical. connection, to the transport layer at the destination host. Transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host.

Network layer versus Transport layer

The first duty of a transport-layer protocol is to provide process-to-process communication. A process is an application-layer entity that sues the services of the transport layer. The network layer is responsible for communication and process-to-process communication. A network-layer protocol can deliver the message only to the destination computer. However, this is an incomplete delivery. The message still needs to be handed to the correct process.This is where a transport-layer protocol takes over. A transport-layer protocol is responsible for deliver of the message to the appropriate process.

ICANN has divided the port numbers into three ranges: well-known, registered, and dynamic.

- Well-known: The ports ranging from 0 to 1023 are assigned and controlled by ICANN.

- Registered ports: The ports ranging from 1024 to 49,151 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication.
- Dynamic ports: The ports ranging from 49.152 to 65.535 are neither controller nor registered. They can be used as temporary or private port numbers.

Socket Addresses

A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection. The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely.

Encapsulation and Decapsulation

To send a message from one process to another, the transport-layer protocol encapsulate and decapsulates messages. Encapsulation happens at the sender site. When a process has a message to send, it passes the message to the transport layer along with a pair of socket address and some other pieces of information, which depend on the transport-layer protocol. Decapsulation happens at the receiver site. When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer. The sender sock address is passed to the process in case it needs to respond to the message received.

Multiplexing and demultiplexing

Whenever an entity accepts items from one than one source, this is referred to as multiplexing (many to one); whenever an entity delivers items to more than one source, this is refereed to as demultiplexing(many to one). The transport layer at the source performs multiplexing; the transport layer at the destination performs demultiplexing.

Pushing or Pulling

Delivery of items from a producer to a consumer can occur in one of two ways: pushing or pulling. If the sender delivers items whenever they are produced- without a prior request from the consumer- the deliver is referred to as pushing. If the producer delivers the items after the consumer has requested them, the delivery is referred to as pulling.

Flow control at the transport layer

In communication at the transport layer, we are dealing with four entities: sending process, sender transport layer, receiver transport layer, and receiver process. The sending process at the application layer is only a producer. It produces message chunks and pushes them to the transport layer. The sending transport layer has a double role: it is both a consumer and a producer. It consumes the messages pushed by the producer. It encapsulates the messages in packets and pushes them to the receiving transport layer. The receiving transport layer also has a double role: it is the consumer for the packets received from the sender and the producer that decapsulates the messages and delivers them to the application layer.

Error control at the transport layer

Reliability can be achieved to add error control services to the transport layer. Error control at the transport layer is responsible for:

- Detecting and discarding corrupted packets

- Keeping track of lost and discarded packets and re-sending them
- Recognizing duplicate packets and discarding them.
- Buffering out-of-order packets until the missing packets arrive.

5.4 User Datagram Protocol UDP

The user Datagram protocol(UDP) is a connectionless, unreliable transport protocol. It does not add anything to the services of IP except for providing process-to-process communication instead of host-to-host communication. UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message using UDP takes much less interaction between the sender and receiver than using TCP.

Connectionless services

UDP provides a conectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination.

5.5 Intro to TCP

Sliding Window

Since the sequence numbers use modulo 2 to the power of m, a circle can represent the sequence numbers from 0 to 2 to the power of m -1. The buffer is represented as a set of slices, called the sliding window, that occupies part of the circle at any time. At the sender site, when a packet is sent, the corresponding slice is marked. When all the slides are marked, it means that the buffer is full and no further messages can be accepted from the application layer. When an acknowledgement arrives, the corresponding slice is unmarked, the window slices over the range of the corresponding sequence numbers to allow more free slices at the end of the window.

Window.png

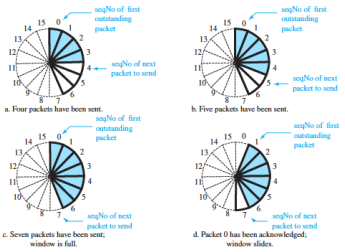


Figure 8: ShareLaTeX logo

Connection

A transport-layer protocol, like a network-layer protocol, can provide two types of services: conectionless and connection-oriented.The nature of these services at the transport layer, however, is different from the ones at the network layer. At the network layer, a conectionless service may mean

different paths for different datagrams belonging to the same message.Connectionless service at the transport layer means independency between packets; connection-oriented means dependency. The data exchange can only happen after the connection establishment. After data exchange, the connection needs to be torn down.We can also implement flow control, error control, and congestion control in a connection oriented protocol.

6 Week9

6.1 Transport Control Protocol TCP

TCP segment format

- source port address 16 bits
- Destination port address 16 bits
- sequence number 32 bits
- Acknowledgment number 32 bits

6.2 Data transfer in TCP

- Send window
- Receiving application
- (last byte, next byte, last byte received)

States for TCP

- Closed - No connection exists
- Listen - Passive open received; waiting for SYN
- SYN-SENT - SYN sent; waiting for ACK
- SYN-RCVD - SYN + ACK sent; waiting for ACK
- Established - connection established; data transfer in progress
- FIN-WAIT-1 - First FIN sent; waiting for ACK
- FIN-WAIT-2 - ACK to first FIN received; waiting for second FIN
- close-wait - First FIN received, ACK sent; waiting for application to close
- time-wait - second FIN received, ACK sent; waiting for 2MSL time-out
- Last-ACK - second FIN sent, waiting for ACK
- closing - both sides decided to close simultaneously

6.3 Windows in TCP

TCP uses two windows: send and receive window. For each direction of data transfer, which means four windows for a bidirectional communication. To make the discussion simple, we make an unrealistic assumption that communication is only unidirectional.

A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data. TCP sliding windows are byte-oriented.

6.4 FSM for data transfer in TCP

Data transfer in TCP is close to the selective-repeat protocol with a slight similarity to GBN. Since TCP accepts out-of-order segments, TCP can be thought of as behaving more like the SR protocol, but since the acknowledgements are cumulative, it looks like GBN.

Sender-slide FSM :

- We assume that the communication is unidirectional and the segments are acknowledged using ACK segments. We also ignore selective acknowledgements and congestion control for the moment.

Receiver-slide FSM:

- Now let us show a simplified FSM for the receiver-side TCP protocol similar to the one we discuss for the SR protocol, but with some changes specific to TCP. We assume that the communication is unidirectional and the segments are acknowledged using ACK segments.

Lost segment

- We show what happens when a segment is lost or corrupted. A lost or corrupted segment is treated the same way by the receiver.
- A lost segment is discarded somewhere in the network; a corrupted segment is discarded by the receiver itself.
- Both are considered lost.

6.5 Notes on sliding window

Some points about TCP sliding windows:

- The size of the window is the lesser of rwnd and cwnd
- The source does not have the send a full window's worth of data
- The window can be opened or closed by the receiver, but should not be shrunk.
- The destination can send an acknowledgement at any time as long as it does not result in a shrinking window.
- The receiver can temporarily shut down the window; the sender, however, can always send a segment of 1 byte after the window is shut down.

- ACK segments do not consume sequence numbers and are not acknowledged.
- In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.
- No retransmission timer is set for an ACK segment

6.6 Flow Control / Error control

Flow control balances the rate a a producer creates data with the rate a consumer can use the data. TCP separates flow control from error control. TCP is a reliable transport-layer-protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.

7 Week 10

7.1 Application-Layer Paradigms

It should be clear that to use the internet we need two application programs to interact with each other: one running on a computer somewhere in the world, the other running on another computer somewhere else in the world.

The two programs need to send messages to each other through the internet infrastructure. The two paradigms have been developed: The client - server paradigm and the peer-to-peer paradigm.

How can a client process communicate with a server process?

A computer program is normally written in a computer language with a predefined set of instructions that tells the computer what to do. If we need a process to be able to communicate with another process, we need a new set of instructions to tell the lowest four layers of the TCP/IP suite to open the connection, send and receive data from the other end, and close the connection. A set of instructions of this kind is normally referred to as an application programming interface(API).

7.2 Socket Addresses

A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection. The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely.

7.3 HyperText Transfer Protocol

The HTTP is used to define how the client-server programs can be written to retrieve web pages from the Web. An HTTP client sends a request; an HTTP server returns a response. The server

uses the port number 80; the client uses a temporary port number. HTTP uses the services of TCP, which, as discussed before, is a connection-oriented and reliable protocol.

Nonpersistent connections

In a nonpersistent connection, one TCP connection is made for each request/response. The following lists the steps in this strategy:

- The client opens a TCP connection and sends a request.
- The server sends the response and closes the connection
- The client reads the data until it encounters an end-of-file marker; it then closes the connection.

The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened.

Persistent connection

In a persistent connection, the server leaves the connection open for more requests after sending a response. The server can close the connection at the request of a client or if a time-out has been reached. The sender usually sends the length of the data with each response. However, there are some occasions when the sender does not know the length of the data. This is the case when a document is created dynamically or actively.

7.4 Request and Response messages

The HTTP protocol defines the format of the request and response messages. The first section in the request message is called the request line whereas the first message in the response message is called the status line. The other three sections have the same name but different contents.

Request messages

There are three fields in this line separated by one space and terminated by two characters. The fields are called method, URL and version. The method field defines the request types.

Methods used for request messages

- GET - Requests a document from the server
- HEAD - Requests information about a document but no the document itself
- PUT - sends a document from the client to the server
- POST - Sends some information from the client to the server
- TRACE - Echoes the incoming request
- DELETE - Removes the web page
- CONNECT - Reserved
- OPTIONS - Inquires about available options

After the request line, we can have zero or more request header lines. Each header line sends additional information from the client to the server. Each header line has a header name, a colon, a space, and a header value.

Response Message

A response message consists of a status line, header lines, a blank line, and sometimes a body. The first line in a response message is called the status line. There are three fields in this line separated by spaces and terminated by a carriage return and line feed. The first field defines the version of HTTP protocol. The status code field defines the status of the request. It consists of three digits. Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site. The status phrase explains the status code in text form.

Response header names:

- Date - shows the current date
- Upgrade - Specifies the preferred communication protocol
- Server - Gives information about the server
- Set-Cookie - The server asks the client to save a cookie
- Content - Encoding - Specifies the encoding scheme
- Content - language - Specifies the language
- Content - Length - Shows the length of the document
- Content - Type - Specifies the media type
- Location - To ask the client to send the request to another site
- Accept-ranges - The server will accept the requested byte-ranges
- Last-modified - Gives the date and time of the last change

The body contains the document to be sent from the server to the client. The body is present unless the response is an error message.

7.5 Electronic Mail

Electronic mail allows users to exchange messages. The nature of this application, however, is different from other applications discussed so far. In an application, such as HTTP or FTP, the server program is running all the time, waiting for a request from a client. When the request arrives, the server provides the service. There is a request and there is a response. In the case of electronic mail, the situation is different. First, e-mail is considered a one-way transaction.

When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs and a pair of MAAs. This is the most common situation today.

Services of user agent User agent provides service to the user to make the process of sending and receiving a message easier. A user agent is a software package that composes, reads, replies to , and forwards messages. It also handles local mailboxes on the user computers.

- Composing messages
- Reading messages

- Replying to messages
- Forwarding messages
- Handling mailboxes

7.6 File Transfer Protocol FTP

FTP is the standard protocol provided by TCP/IP for copying a file from one host to another. Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. FTP uses the services of TCP. It needs TCP connections. The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.

Network Virtual Terminal

When accessing a remote computer in the world, we need to know what kind of operating system the computer uses and what kind of characters it uses. For instance DOC and UNIX use different characters to implement a command such as "end-of-file".

8 Week 11

8.1 Traffic

Data traffic

The main focus on congestion control and quality of service is data traffic. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic.

Three traffic profiles

- Constant bit rate - The data rate of an ATM service class that is designed for customers requiring real-time audio or video services.
- Variable bit rate - The data of an ATM service class for users needing a varying bit rate.
- Bursty - Data with varying instantaneous transmission rates.

Congestion

Congestion in a network may occur if the load on the network - the number of packets sent to the network - is greater than the capacity of the network - the number of packets a network can handle. Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

Queues in a router

Network layer:

Router is a computer with several interfaces which have buffers in and out. The packets arrive on the input and are being queued as they wait for computer to process them. The TPU processes a packet. CPU will make a decision based on the database (routing table).

In terms of through put The more packets you send, the more traffic you have (increasing throughput). At some point you will get congestion. The capacity you were promised, your link may be able to support a higher value.

Quality Of Service

Quality of service (QoS) is an internetworking issue, which we can define as something a flow seeks to attain.

8.2 Flow characteristics

The four characteristics are : Reliability, Delay, Jitter, Bandwidth.

Questions we want to ask ourselves is FTP sensitive to reliability, delay, jitter or bandwidth ?

Reliability

Reliability is a characteristic that a flow needs in order to deliver the packets safe and sound to the destination. Lack of reliability means losing a packet or acknowledgment, which entails retransmission.

Delay

Source-to-destination delay is another flow characteristic. Again, applications can tolerate delay in different degrees. In this case, telephony, audio conferencing, video conferencing, and remote logging need minimum delay, while delay in file transfer or e-mail is less important. Cannot have delay equal to 0, it would be idea but might not be possible.

Jitter

Jitter is the variation in delay for packets belonging to the same flow.

Bandwidth

The range of frequencies contained in a composite signal is its bandwidth. Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.

Queuing

weighted fair queuing is a method of automatically smoothing out the flow of data in packet-switched communication networks by sorting packets to minimize the average latency.

Priority queuing is a congestion management technique. PQ schedules traffic suc that the higher-priority queues "always" gets serviced first. This can cause the traffic of other lower-priority queues to starve out. PQ uses 4 different queues- high, medium, normal and low.