

Telecommunications Systems & Networks – INTRODUCTION PART 2

Małgorzata Langer

Main types of protocols

- ‚PROTOCOL‘ – what it is?
- Connection-oriented (as **TCP/IP**)
- Connectionless (as **UDP/IP**)

Ping

- A software name used in TCP/IP (as internet) to diagnose net connections. It allows for testing the connection between testing and tested parties (also its quality, and latence)

- The range of valid host addresses contains *between* the subnet number (host field of all zeros) and the broadcast address (host field of all ones)
- So the host address range for subnet 172.16.8.0/22 is

10101100.00010000.00001000.00000001=172.16.8.1

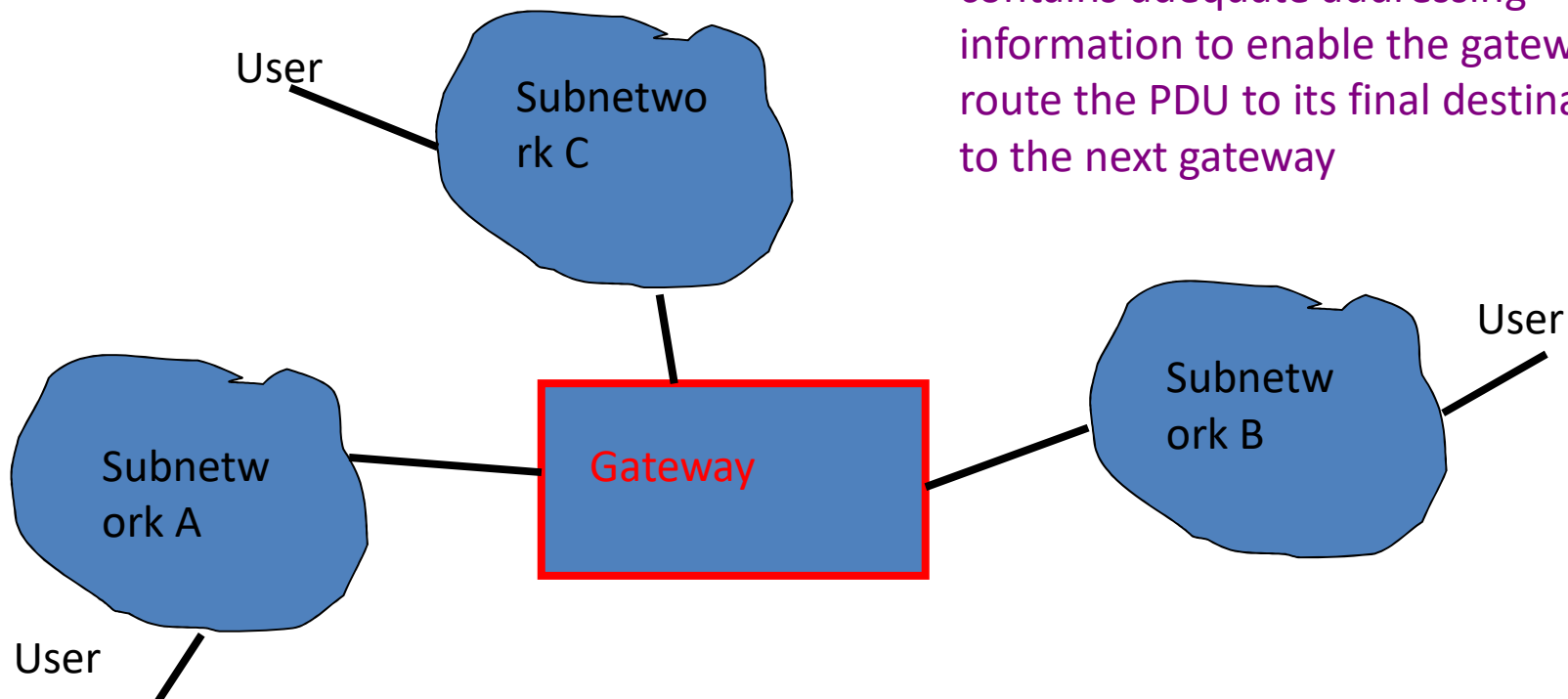
through

10101100.00010000.00001011.11111110=172.16.11.254

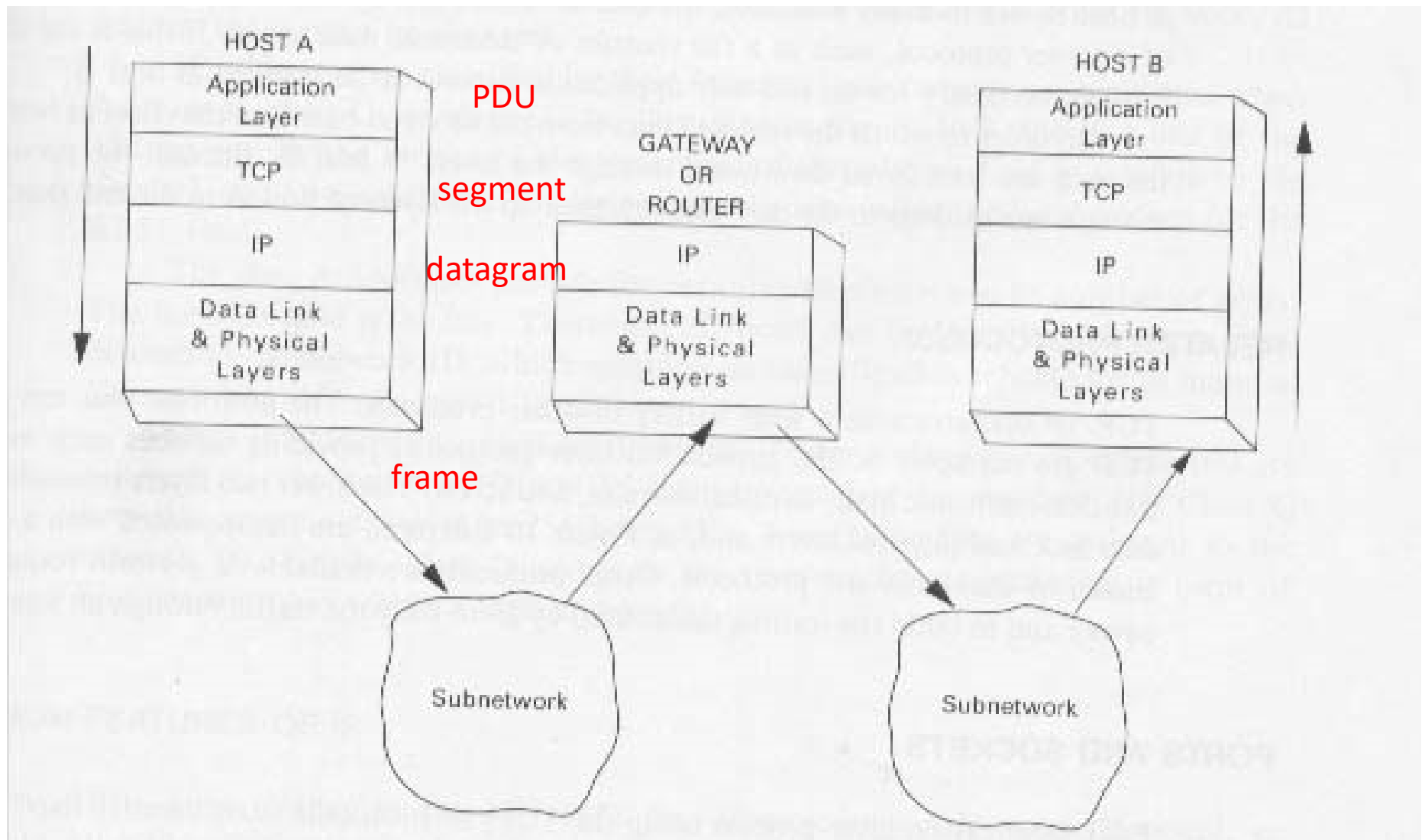
TCP/IP

- The Transmission Control Protocol/Internet Protocol
- Gateway, router

The principal purpose of the gateway is to receive a protocol data unit (PDU) that contains adequate addressing information to enable the gateway to route the PDU to its final destination, or to the next gateway



Example of TCP/IP Operations



- TCP/IP is unaware of what goes on inside the network. The network manager is free to manipulate and manage the PDU in any manner necessary. In most instances the Internet PDU (data and headers) remains unchanged as it is transmitted through the subnet.

At the gateway...

- PDU is processed through the lower layers and passed to the IP (network) layer. Here routing decisions are made based on the address provided by the host computer
- The datagram is passed to the communications link that is connected to the appropriate subnetwork
- The datagram is re-encapsulated into the data link layer PDU (as a frame) and passed to the next subnetwork.

Ports & Sockets

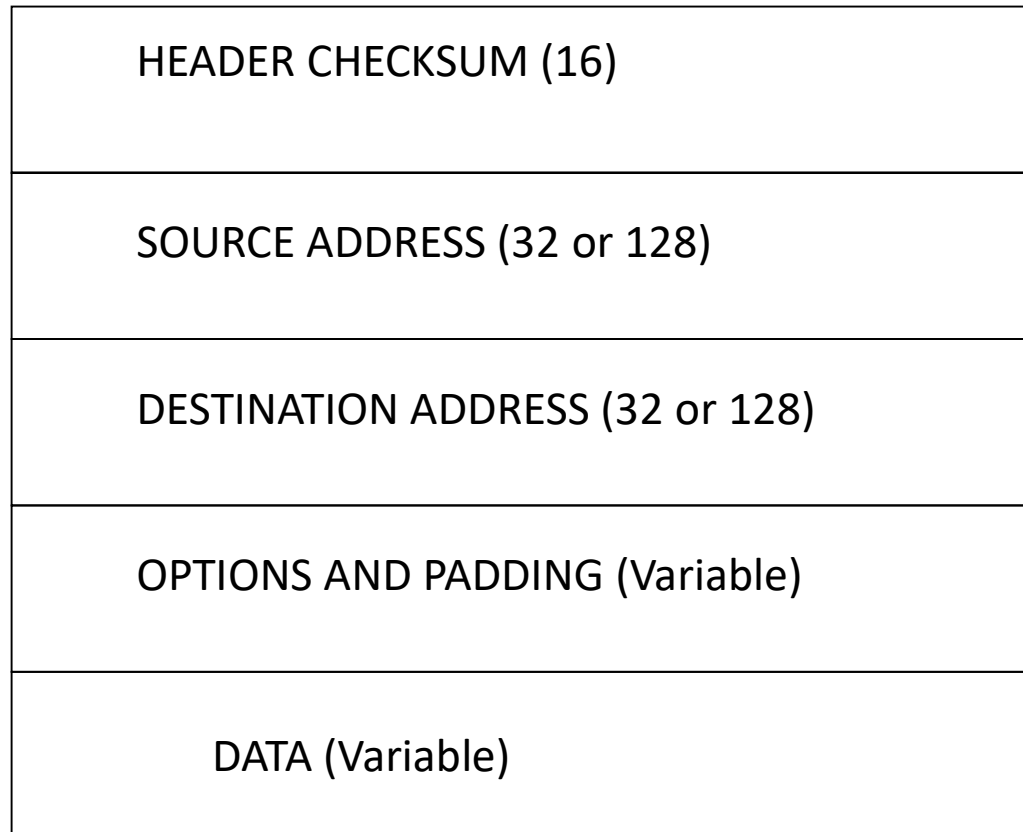
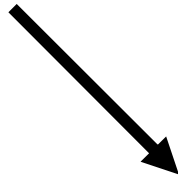
- Each application layer process using the TCP/IP protocols must identify itself with a *port* number
- A *socket* identifies an endpoint communications process (a concatenation of a port number and the network address)
- Some port numbers are preassigned (from 0 to 255) and called *well-known-ports*

The IP Datagram

VERSION (4)	HEADER LENGTH (4)
TYPE OF SERVICE (8)	
TOTAL LENGTH (16)	
FLAGS (3)	FRAGMENT OFFSET (13)
TIME TO LIVE (8)	
PROTOCOL (8)	



The IP Datagram - continued



IP Datagram

- **Version field** – identifies the version of IP in use (some network nodes may not have the same release of protocol) – the current version of IP is 6 (128 bits) or still 4 (32 bits)
- **Header length** field contains 4 bits which are set to a value to indicate the length of the datagram header, measured in 32-bit words. Typically a header contains 20 bytes, and the value in the length field usually is 5

- **Type of service** (TOS) can be used to identify several functions (the first 2 bits)
 - Bit 3 – delay bit (D bit); when set to 1 this TOS requests a short delay through an internet
 - Bit 4 – throughput bit (T bit); when set to 1 requests for high throughput through an internet
 - Bit 5 – reliability bit (R bit) which allows a user to request high reliability for the datagram
 - Bits 6 and 7 – not used

- Total length field specifies the total length of the IP datagram; measured in bytes and includes the length of the header and the data. The maximum possible length of a datagram is 65,535 bytes (2^{16})

All gateways must accommodate datagrams of 576 bytes in total length

IP subtracts the header length from the total length field to compute the size of the data field.

Three fields in the header to control datagram fragmentation and reassembly

- **Identifier** – it serves with the source address at the receiving host to identify the fragment
- **Flags** – contains bits to determine if the datagram may be fragmented, and if yes, one of the bits can be set to determine if this fragment is the last fragment of the datagram
- **Fragmentation offset** – contains a value which specifies the relative position of the fragment to the original datagram; the value is initialized as 0 and is subsequently set to the proper number if the gateway fragments the data; measured in units of eight bytes

TTL – time-to live

- The parameter is used to measure the time a datagram has been in the internet. Each gateway is required to check this field and decrements this field in each datagram it processes – prevents endless loops
also may be set for diagnostic purposes

- **Protocol field** – identifies the next level protocol above the IP that is to receive the datagram at the final host destination; numbers are used for the most widely used: 6=TCP, 20=OSI transport layer...)
- **Header checksum** is used to detect a distortion that may have occurred in the header. Checks are not performed on the user data stream.
- **Source and destination addresses** – remain the same value throughout the life of the datagram; they are IP addresses; 128 is a very long string; there are some methods to reduce it

- **Option field** is used to identify several additional services; is not used in every datagram; often for network management and diagnostics
- **Padding field** – may be used to make certain that the datagram header aligns on an exact 32-bit boundary
- **Data field** contains the user data; the combination of the data field and the header cannot exceed 65,535 bytes.

Major IP Services

- Source routing
- Routing operations
- Loose and strict routing
- Route-Recording Option
- Timestamp Option
- ICMP module

Source routing

- Allows an upper-layer protocol (ULP) to determine *how the IP gateways route the datagrams*. The ULP has the option of passing a list of internet addresses to the IP module – the intermediate IP nodes that are to be transited during the routing of the datagrams to the final destination – the last address on the list.

Source routing - continued

- When IP receives a datagram, it uses the address in the source routing field to determine the next intermediate hop. IP uses a pointer field to learn about the next IP address. Then IP replaces the value in the source routing list with its own address (and increments the pointer by one address (4 bytes) in order for the next hop to retrieve the next IP address in the route. And that is why the datagram follows the source route dictated by the ULP and also records the route along the way

Routing Operations

- The IP gateway makes routing decisions based on the routing list. If the destination host resides in another network, the IP gateway must decide how to route to the other network
- Each gateway maintains a routing table (static or dynamic) that contains an entry for each reachable network (the address of the network and the one of a neighbour gateway).

The neighbour gateway

- Is the shortest route to the destination network

The distance metric

The number of hops between the gateway and the final destination

If no match is found....

- The gateway builds an error message to send back to the IP source by *the Internet Control Message Protocol (ICMP)*, that contains a 'destination unreachable' code.

Loose & Strict Routing

- **Loose source routing** – IP modules use intermediate hops to reach the addresses obtained in the source list, as long as the datagram traverses the nodes listed
- **Strict source routing** – the datagram travels through the networks, whose addresses are indicated in the source list, only. If the strict source route cannot be followed, the originating host IP is notified with an error message.

Route-Recording Option

- It operates as source routing with the recording feature

The Timestamp Option

Each IP module gives its stamp – the time is based on milliseconds, using universal time (Greenwich)

ICMP – the internet control message protocol

- **Why?** – the IP is a connectionless-mode protocol, and as such it has no error-reporting or error-correcting mechanisms
- **What for?** – the ICMP reports errors in the processing of a datagram and provides for some administrative and status messages

ICMP Message Format

IP HEADER
TYPE (8)
CODE (8)
CHECKSUM (16)
PARAMETERS, OR NOT USED (32)
INFORMATION (VARIABLE)

the protocol field in the IP header is set to 1 to signify the use of ICMP

to define the type of message

To describe the type of error or status information

To compute a 16-bit 1s complement on the ICMP message

ICMP messages are carried in the data portion of the IP datagram;

The ICMP error-reporting message also carries the internet header and the first 64 bits of the user data field, useful for troubleshooting and problem analysis

ICMP Error- & Status-Reporting Procedures

- Time exceeded on datagram lifetime
- Parameter unintelligible
- Destination unreachable
- Source quench
- Echo and echo replay
- Redirect
- Timestamp and timestamp reply
- Information request or information reply
- Address mask request and reply

TCP

- TCP resides in **the transport layer** (above IP and below the upper layers)
- TCP performs the tasks of reliability, flow control, sequencing, opening and closing
- It is used as TCP/IP but can support other protocols (ISO, FTP, etc...)
- TCP is a connection-oriented protocol

TCP is a connection – oriented protocol

- Is responsible for the end-to-end transfer of data across one network or multiple networks
- Is responsible for the reliable transfer of each character (byte, octet) passed to it from an upper layer – creates a virtual circuit
- Returns a positive acknowledgement (ACK) to the sending TCP module

Stream-oriented protocol

- Sends **individual characters**, and *not blocks, frames, etc.*
- When the bytes arrive at the TCP layer – are grouped in the *segments*
- The length of the segment is determined by TCP (an administrator determines how TCP makes this decision)

Other functions of TCP

- TCP checks for duplicate data
- Supports a 'push' function
- Can use the segments' numbers for ACK, but also to resequence the segments at the final destination
- Eliminates duplicate segments
- Multiplexes multiple user sessions within a single host computer onto the ULPs (sharing ports and sockets)
- Provides full-duplex transmission between two TCP entities
- Gives the capability to specify levels of security and precedence
- Provides a 'graceful close' to a virtual circuit

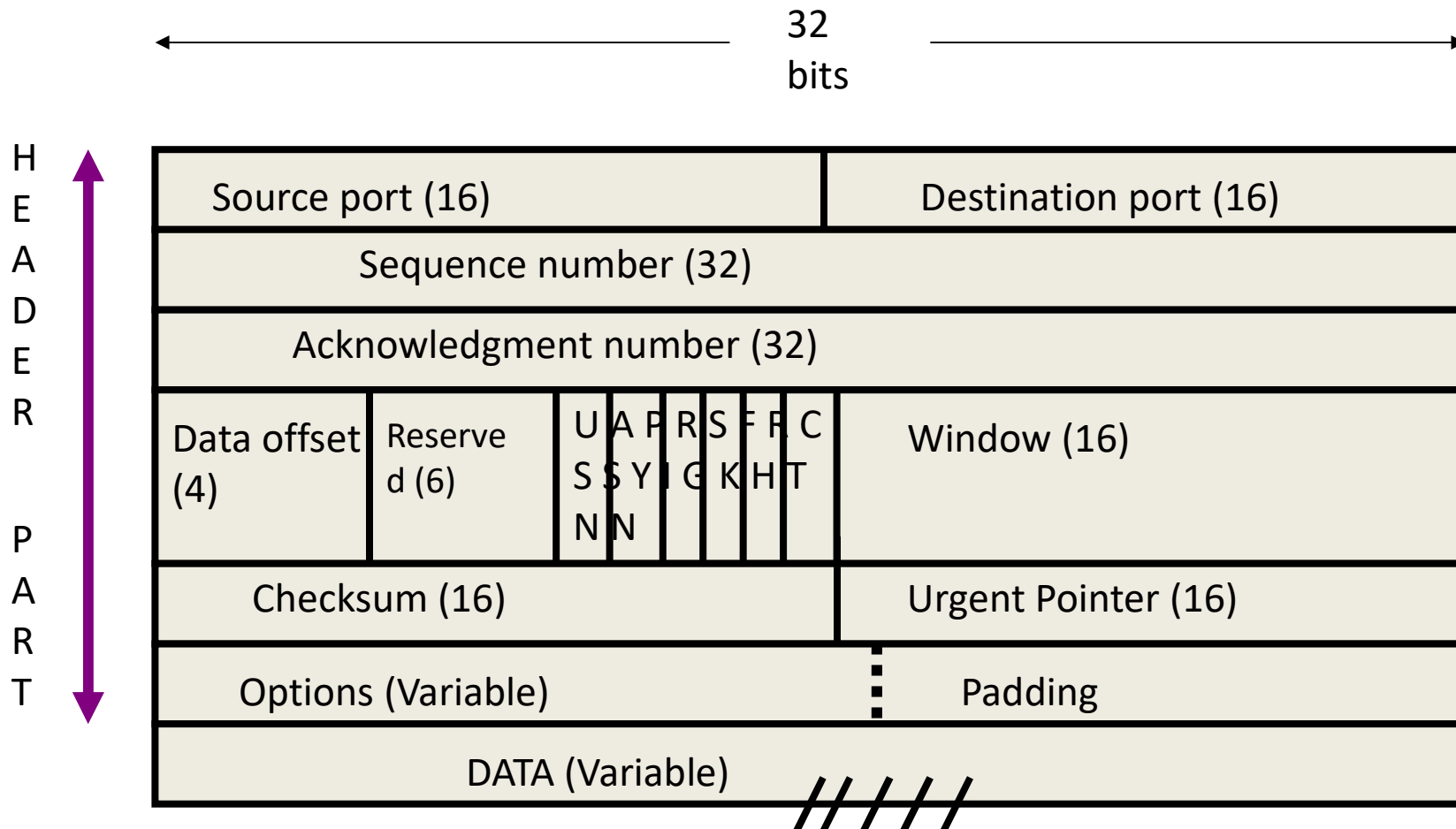
Passive and Active Opens

- The passive open – a server tells the TCP that it is to wait for the arrival of connection request. There will be no delay to accommodate the communications then when it arrives
- The active open – the ULP (server) specifically designates another socket through which a connection is established (the active open is issued to a passive open port in order to establish a virtual circuit)

The Transmission Control Block (TCB)

- It stores the information about each virtual connection.
It also contains several variables associated with the send and receive sequence numbers.

The TCP Segment (PDU)



- **Source port & Destination port** – identify the upper layer application programs that are using the TCP connection
- **Sequence number** – the sequence number of the first byte in the user data field (specifies the position of the transmitting module's byte stream)
- ***During connection management operation*** – it specifies the **initial send sequence** (ISS) number that is to be used for the subsequent numbering of the user data

- **Acknowledgment number** is set to a value which acknowledges data previously received (all the bytes up to and including this number, minus 1)
- **Data offset** field specifies the number of the 32-bit aligned words that comprise the TCP header; is used to determine where the data field begins
- **Reserved field** – consists of 6 bits that MUST be set to 0 (is reserved for future use)

Flags

- **URG** indicates that the urgent pointer field is significant
- **ACK** signifies whether the acknowledgment field is significant
- **PSH** signifies that the module is to exercise the push button
- **RST** indicates that the connection is to be reset

Flags - continued

- **SYN** indicates that the sequence numbers are to be synchronized
- **FIN** indicates that the sender has no more data to send (as EOT)

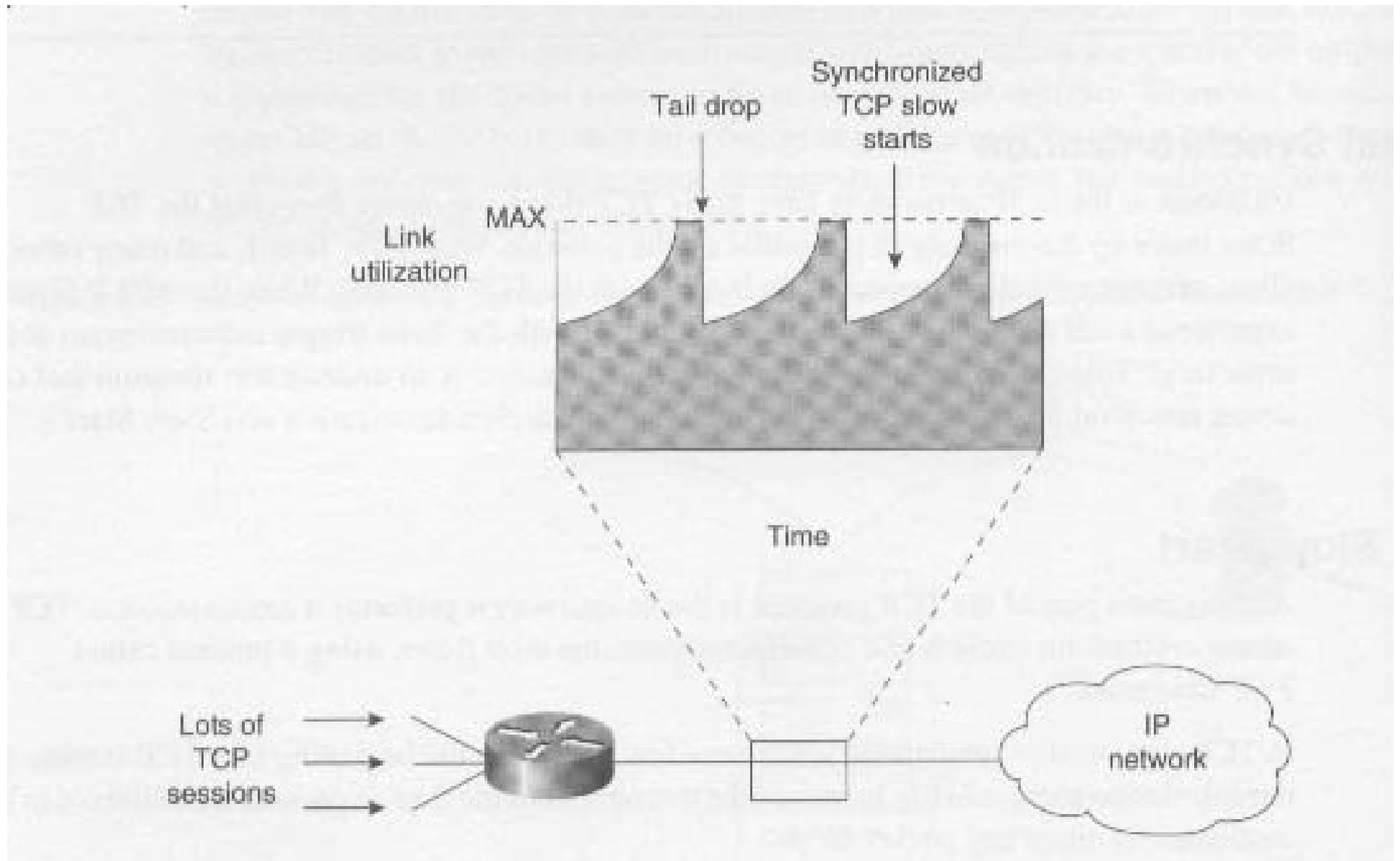
- **Window** – is set to a value indicating how many bytes the receiver is willing to accept
- **Checksum** – performs a 16-bit 1s complement of the 1s sum of all the 16-bit words in the segment (it includes the header and the text)
- **Urgent pointer** – used only if URG flag is set. The purpose is to signify the data byte in which urgent (*out-of-band*) data follow.

- **Options** – the field provides for future enhancements to TCP; three options are defined:
 - 0: end-of-option list
 - 1: no operation
 - 2: maximum segment size
- **Padding** field insures that the TCP header is filled to an even multiple of 32 bits.

TCP Slow Start

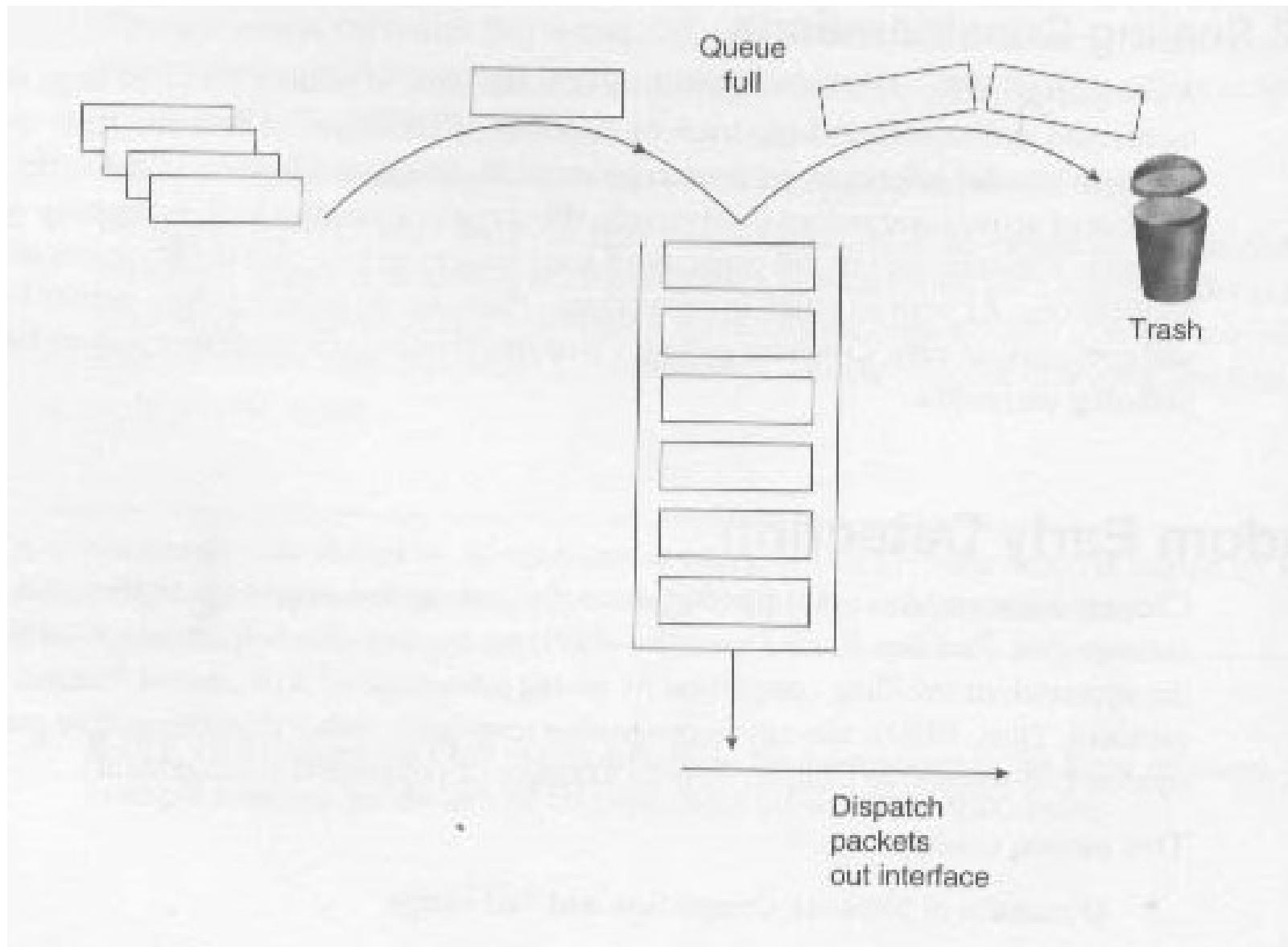
- In TCP slow start, an endstation sends very few packets at the beginning of a TCP session or retransmission and gradually increases the throughput of the flow as packets are delivered to the destination without any packet drops

If many TCP flows go through TCP slow start at the same time, the traffic on the network as a whole drops abruptly (a sawtooth pattern)



The cycle of traffic rising to saturation and crashing down

The main cause – when an interface's queue system is full, the router has no choice but to cause a TAIL DROP



- Tail drop – trailing packets are dropped
- TCP will retransmit the packets that were dropped, (but for example, UDP – no)
- The mostly used queue configuring: FIFO (first-in, first-out); but other queuing systems also experience a tail-drop condition

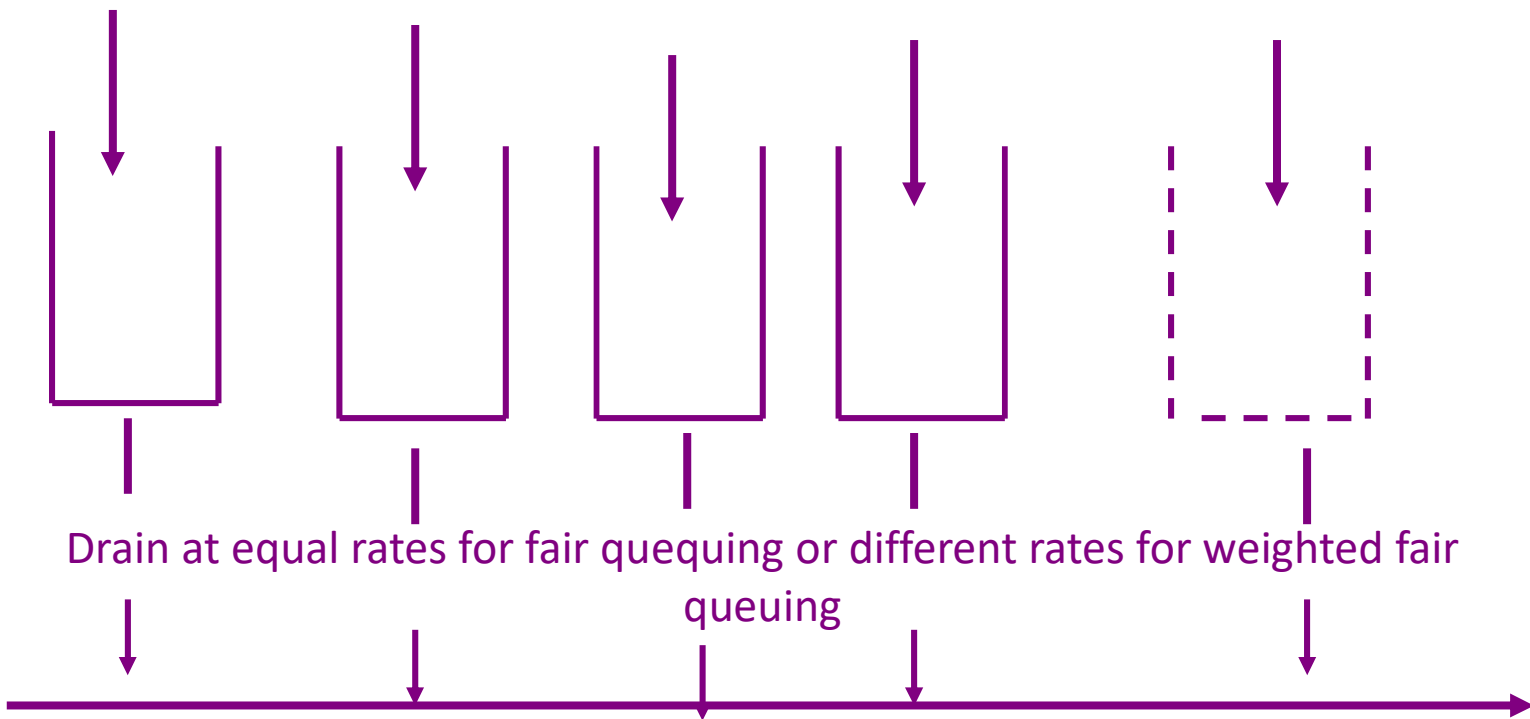
Weighted Fair Queuing (WFQ)

- It dynamically detects traffic flows between applications and automatically manages separate queues for those flows
- In WFQ terms these flows are called *conversations*. A conversation could be a Telnet session, an FTP transfer, a video stream over IP, a transmission of a web page

- WFQ manages multiple conversation queues, one for each unique conversation and sorts packets into their appropriate queues based on the conversations
- WFQ is invoked only when an outbound link is congested
- **fair-queue** sets WFQ, **no fair-queue** disables and sets the queuing strategy into FIFO
- Some interfaces have WFQ as the default for their configuration, so will not show it with **show running-config** command

Fair queuing creates a separate Queue for each conversation

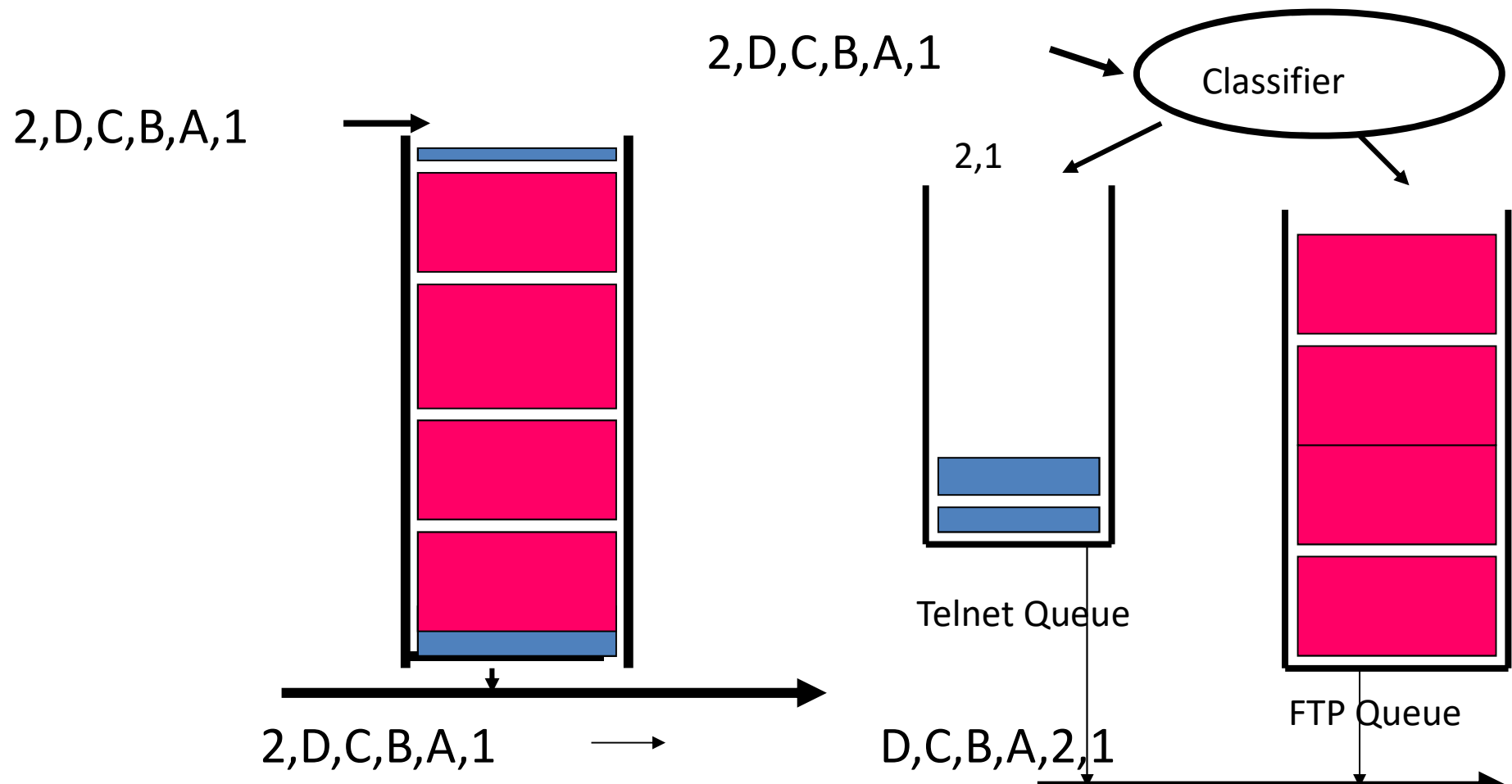
Mark's Video Ann's Telnet Jim's FTP Mark's new conversation Router detects a WWW



Fair Queuing Versus FIFO

A FIFO Queuing Example

Fair Queuing Example



- If Packets 1 and 2 are small packets compared to the large Packets A through D then draining an equal number of bytes from each queue means multiple small packets will be drained for each large packet that is drained
- Even if those high-volume conversations send packets at a very aggressive rate, they do not affect the queues belonging to low-volume conversations. All conversation queues are separate from each other, and the router drains each queue fairly.

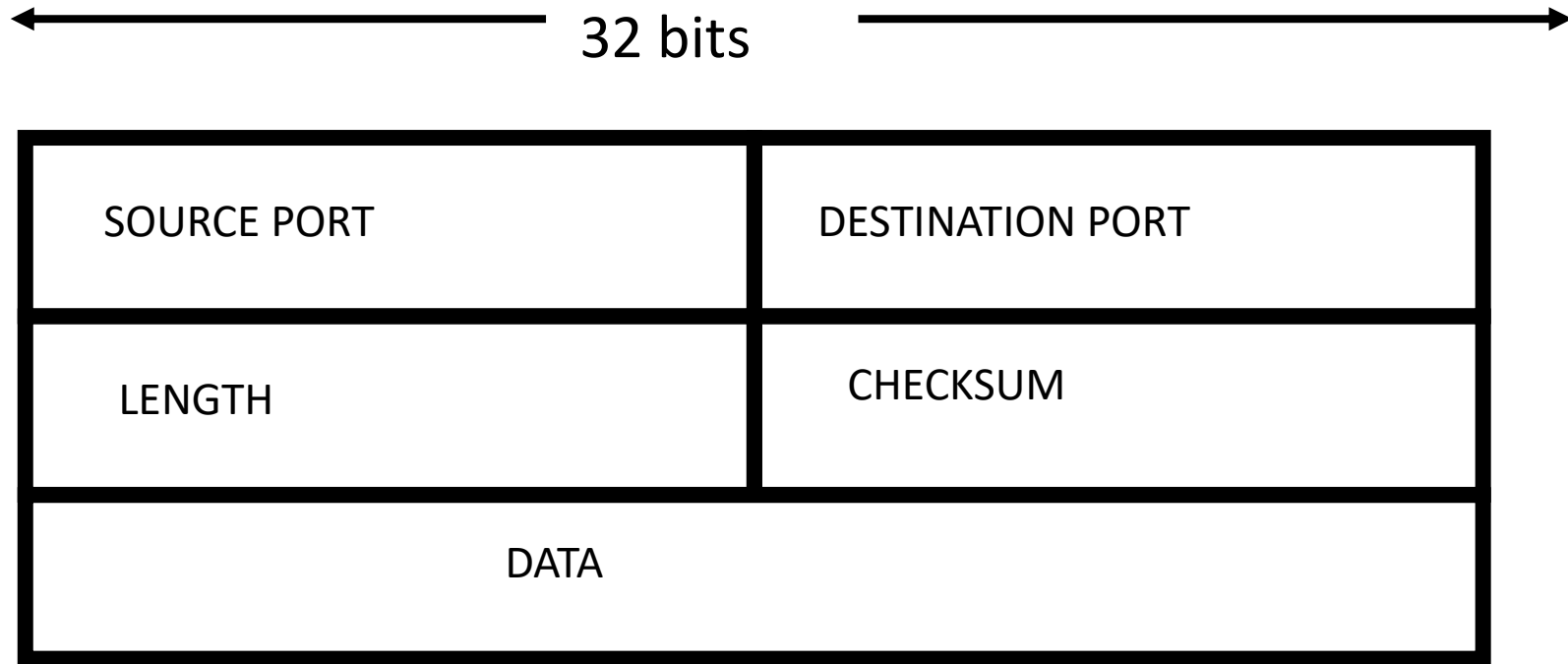
fair-queue 64 128

- Sets the maximum to 128 queues, and 64 as the limit of the number of packets held in any one queue
- Weighting and IP precedence gives weighted fair queuing
- WFQ inspects the IP precedence value of a packet to calculate a number called a *weight* for that packet, and uses this weight to determine how fast that packet will drain out of a conversation queue.

If the full range of services of the TCP is not needed, one may use USER DATAGRAM PROTOCOL (UDP)

- UDP is classified as a connectionless protocol
– provides no reliability or flow control mechanisms, it has also no error recovery procedures
- UDP serves as a simple application interface to the IP – works as a multiplexer / demultiplexer for receiving and sending of IP traffic

Format for the UDP



Source Port: identifies the port of the sending application process. The field is optional. If it's not used, a value of 0 is inserted in this field

- **Destination Port:** identifies the receiving process on the destination host machine
- **Length** – identifies the length of the user datagram, including the header and the data. This value implies that the minimum length is 8 bytes
- **Checksum** – 16-bit 1s complement of the 1s complement sum of the pseudo-IP header, the UDP header, and the data

Random Early Detection (RED)

- RED randomly drops packets based on the number of packets queued on an interface; as a queue reaches its maximum capacity, RED drops packets more aggressively to avoid a tail drop
- RED is a solution for the sawtooth pattern caused by TCP Slow Starts

The problems of the sawtooth pattern:

- Link utilization is not 100% the entire time, and overall throughput is less than the optimal rate
- Transmission rates start and stop; throughput is inconsistent
- Recognizing bandwidth oversubscription and planning for network upgrades is difficult

RED is analogous to pouring the water into the funnel more slowly because one sees the funnel starting to fill in

WRED – Weighted RED

- The combination of RED and IP precedence level: high-priority applications are less likely to experience a packet drop (and TCP slow start) than low-priority ones

Committed Access Rate (CAR)

- Is used to control bandwidth coming into or going out of an interface. (also: rate limiting, policing...)
- CAR may be used to limit the bandwidth coming from a particular source or application
- Traffic exceeding the specified threshold limit can be dropped or reclassified (using IP precedence)

The Application Layer Protocols

- TELNET for terminal services
- TFTP (Trivial File Transfer Protocol) for simple file transfer services
- FTP (File Transfer Protocol) for more elaborate file transfer services
- SMTP (Simple Mail Transfer Protocol) for message transfer services
- Outlook, Netscape

To prepare

- 1- Overageing for lower priorities (problems and solutions)
- 2- How to reduce the number of ACKs and NACKs?