Computer Networks(2015 Pattern) Unit V - Transport Layer

By Prof. A.R.Jain

PVG's COE,Nashik

Note: Material for this presentations are taken from Internet and books and only being used for student reference

8/21/2017

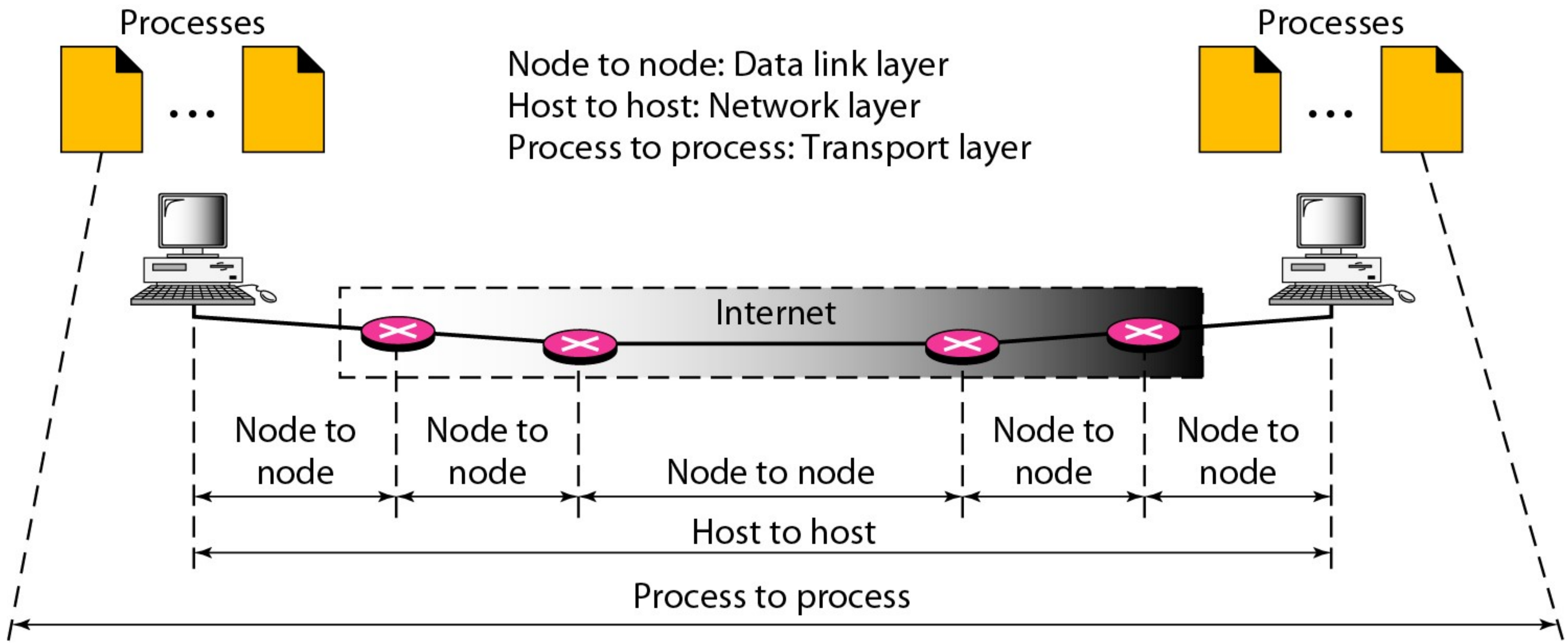
Outline

- **Services,**
- Addressing
- Berkley Sockets,
- Multiplexing,
- TCP
- Connection establishment,
- Connection release,
- Flow control and buffering,
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

Transport Layer Services

- Process to Process delivery
- Connection less as well as connection oriented data delivery
- Error control
- multiplexing/demultiplexing
- reliable data transfer
- flow control
- congestion control

Process to Process Data Delivery



Outline

- **Services,**
- **Addressing**
- Berkley Sockets,
- Multiplexing,
- TCP
- Connection establishment,
- Connection release,
- Flow control and buffering,
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

Addressing

Data Link Layer

- Mac Address (48 bit)
- Physical address
- NIC Card

Network Layer

- IP Address (32 or 128 bit)
- Logical Address
- Machine

Transport Layer

- Port Address (16 bit)
- Logical Address
- Application

Transport services and protocols

provide *logical communication* between app processes running on different hosts

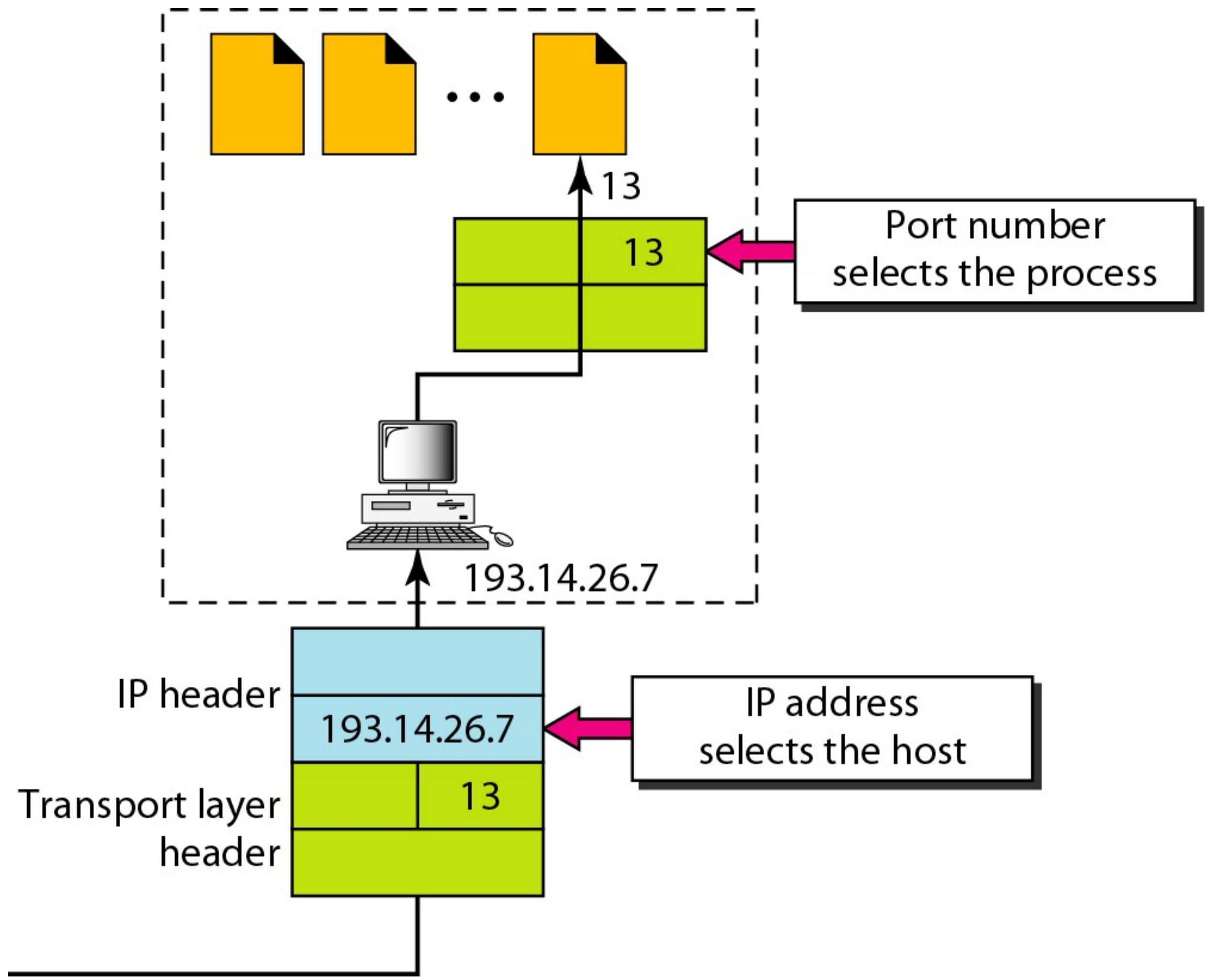
transport protocols run in end systems

- send side: breaks app messages into segments, passes to network layer
- rcv side: reassembles segments into messages, passes to app layer

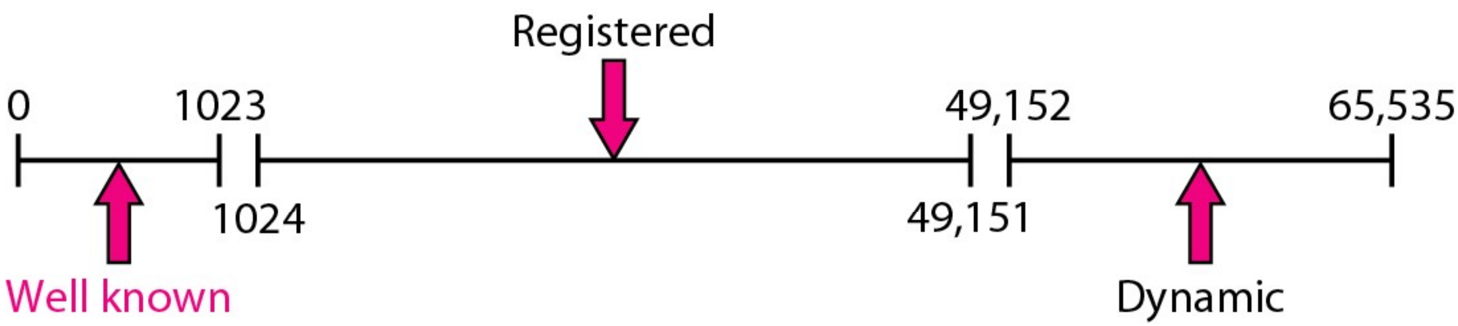
more than one transport protocol available to apps

- Internet: TCP and UDP and SCTP

IP addresses versus port numbers



***PORT ranges by IANA** (Internet Assigned Number Authority)*



Port Ranges by IANA

Well Known

- From 0-1023
- Assigned & controlled by IANA
- These are port no.s for servers ex. FTP(20,21),SMTP (25)

Registered

- From 1024-49151
- Not assigned & controlled by IANA
- Can only be registered with IANA
- Ex. MySQL(3306), MongoDB (27017)

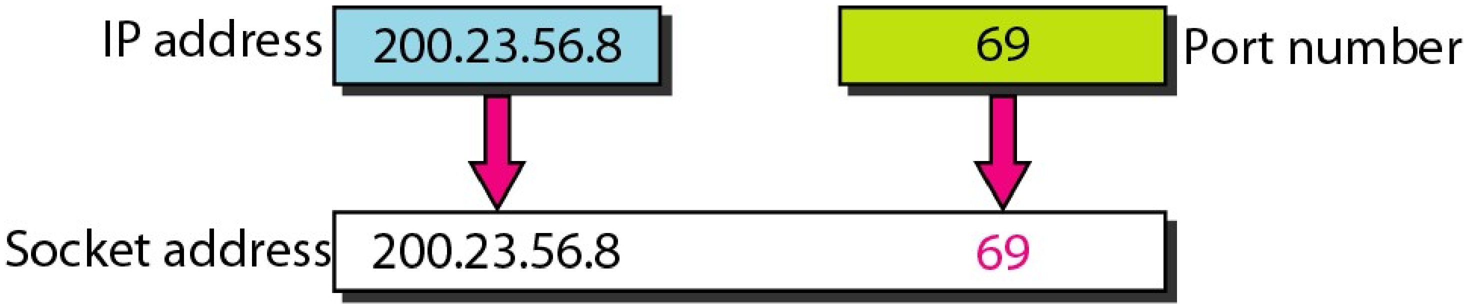
Dynamic

- From 49152-65535
- Nighter controlled nor registered by IANA
- They can be used by any client Program(Process)

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- Multiplexing,
- TCP
- Connection establishment,
- Connection release,
- Flow control and buffering,
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

Socket address



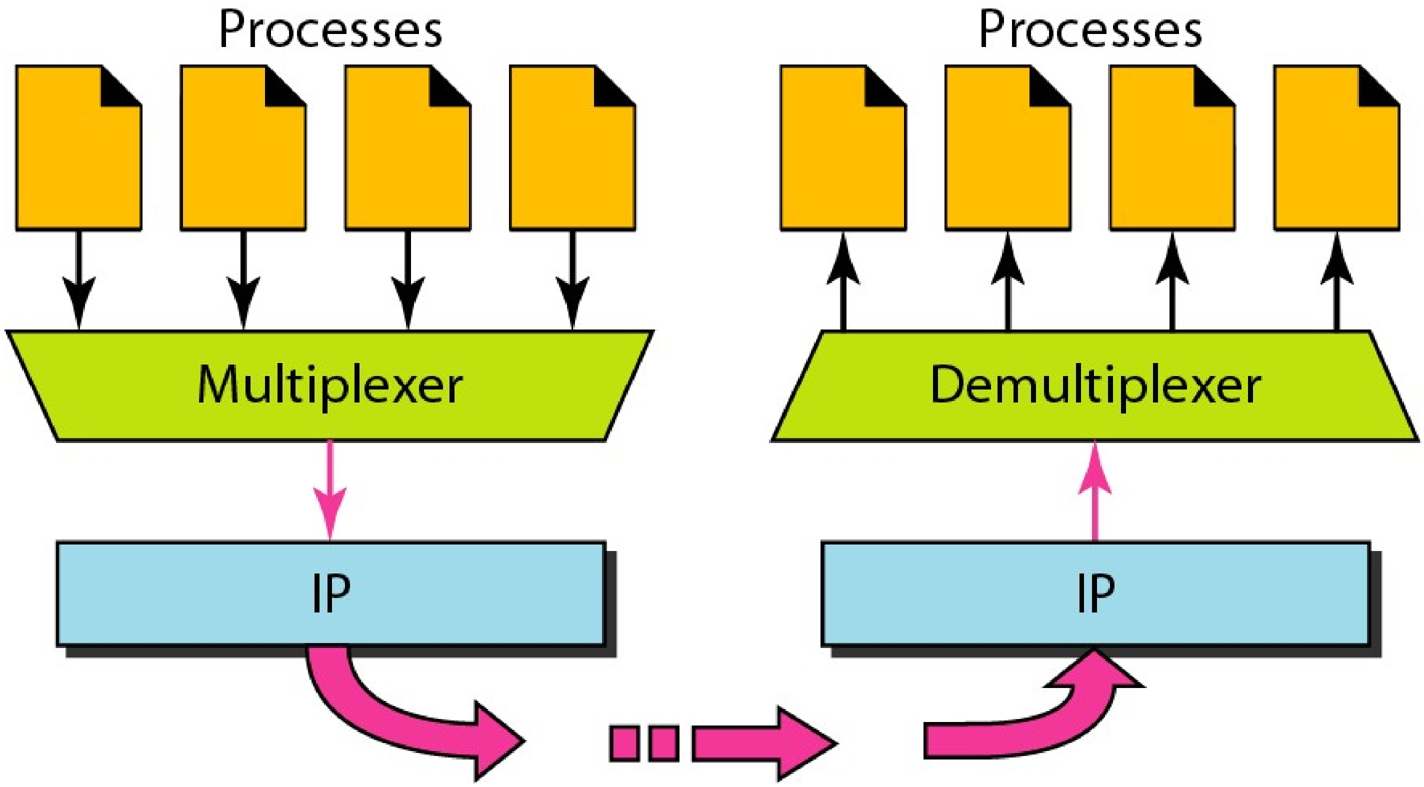
Transport Service Primitives
Berkeley Sockets

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- TCP
- Connection establishment,
- Connection release,
- Flow control and buffering,
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

Multiplexing and demultiplexing



Multiplexing/demultiplexing

Multiplexing at send host:

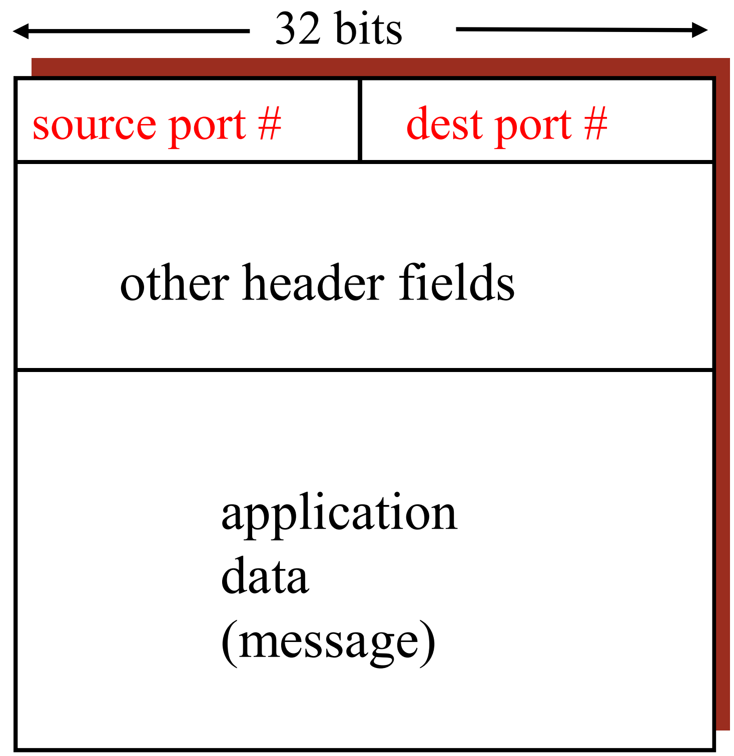
- Gathering data from multiple sockets, enveloping data with header (later used for demultiplexing

Demultiplexing at rcv host:

- Delivering received segments to correct socket

How demultiplexing works

- host receives IP datagrams
  - each datagram has source IP address, destination IP address
  - each datagram carries 1 transport-layer segment
  - each segment has source, destination port number
- host uses IP addresses & port numbers to direct segment to appropriate socket



TCP/UDP segment format

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- Connection establishment,
- Connection release,
- Flow control and buffering,
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

Internet transport-layer protocols

reliable, in-order delivery (TCP)

- congestion control
- flow control
- connection setup

unreliable, unordered delivery: UDP

- Faster data delivery

Stream Control Transmission Protocol (SCTP):

- Faster and reliable data delivery

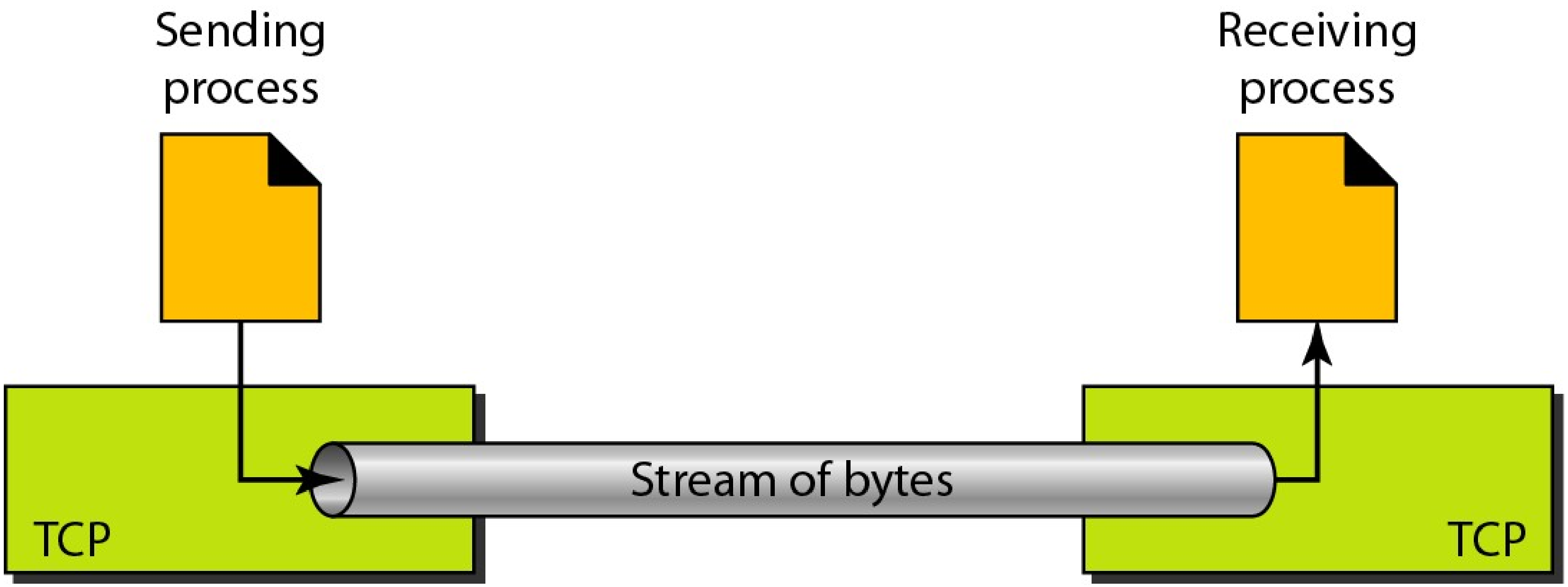
TCP (Transmission control protocol)

- *TCP is a connection-oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level.*

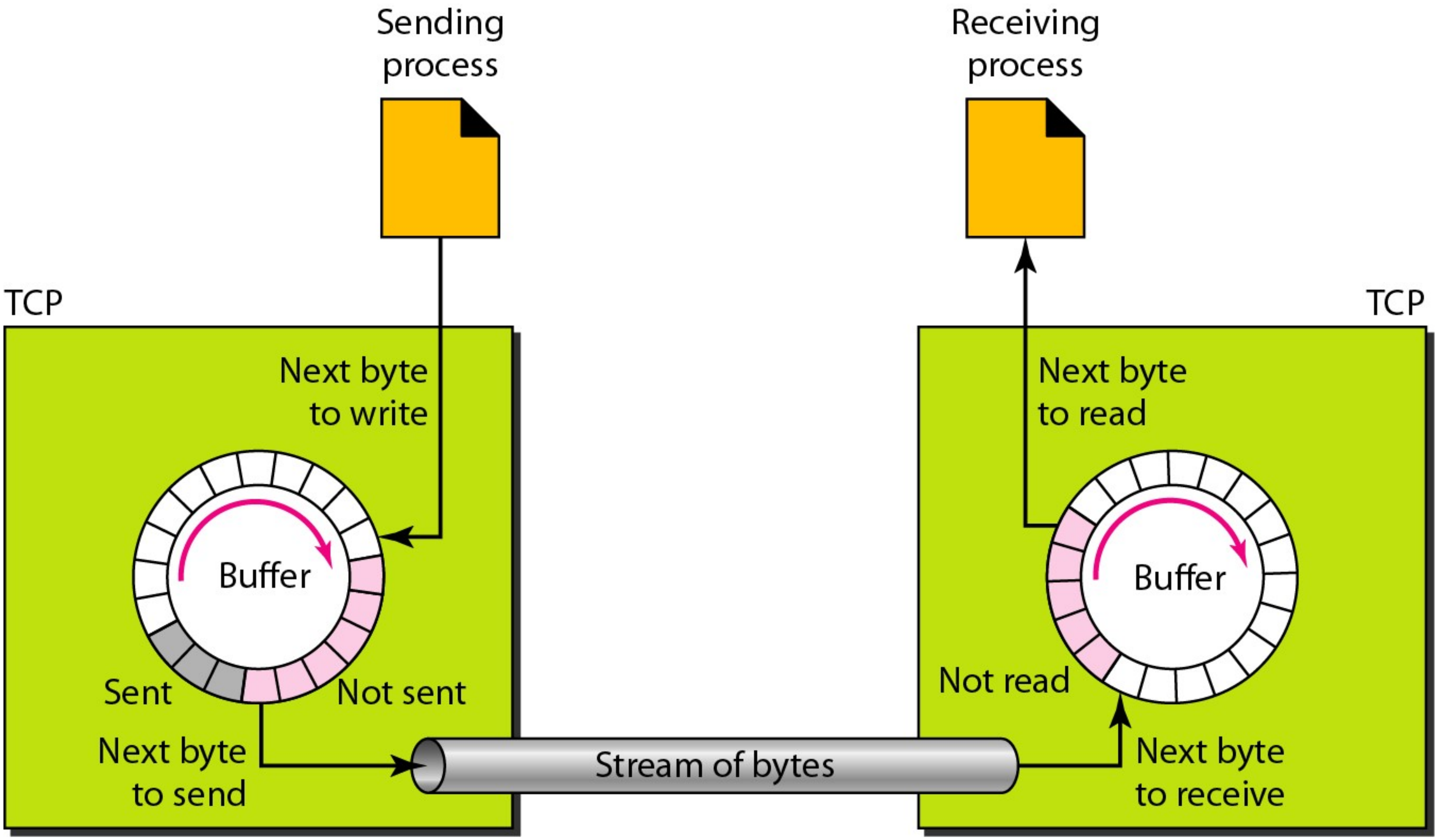
The TCP Service Model-Some assigned ports.

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

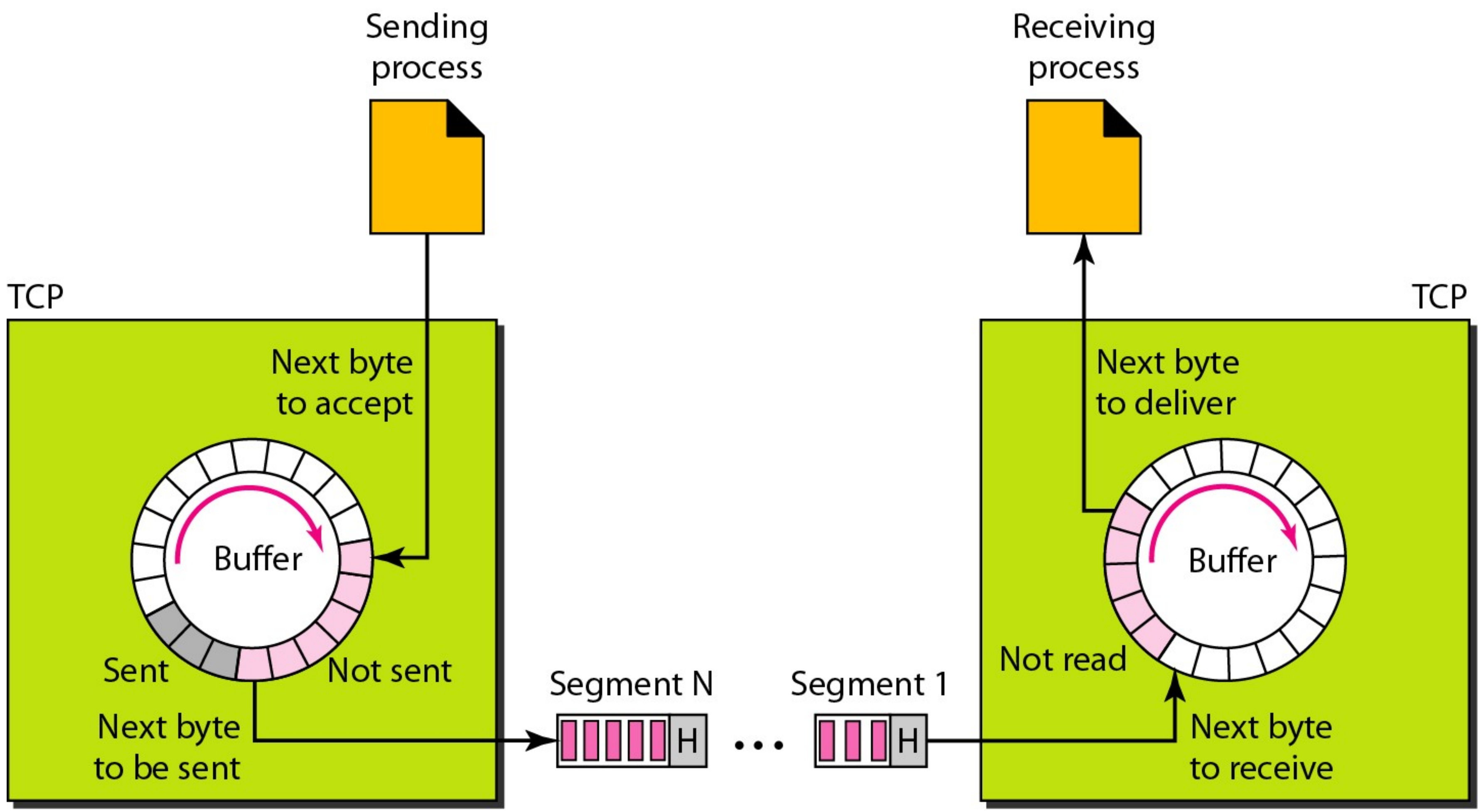
TCP-Stream delivery



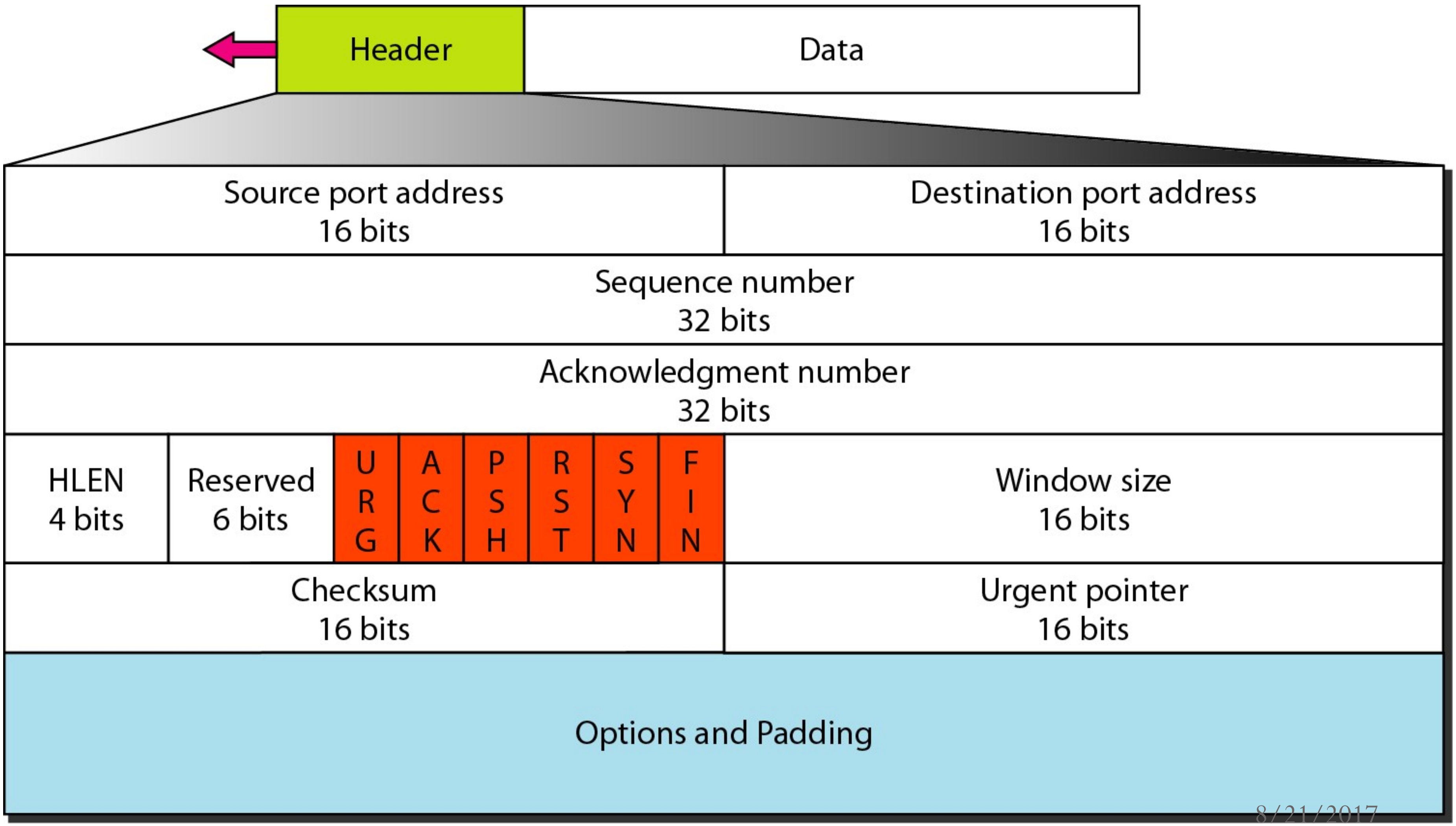
TCP- Sending and receiving buffers



TCP segments

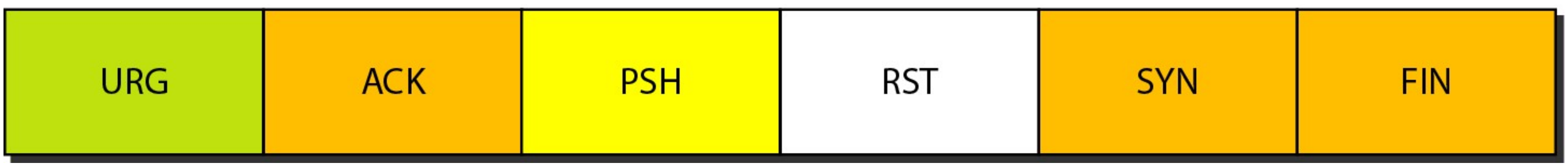


TCP segment format (TCP Header)



TCP -Control field

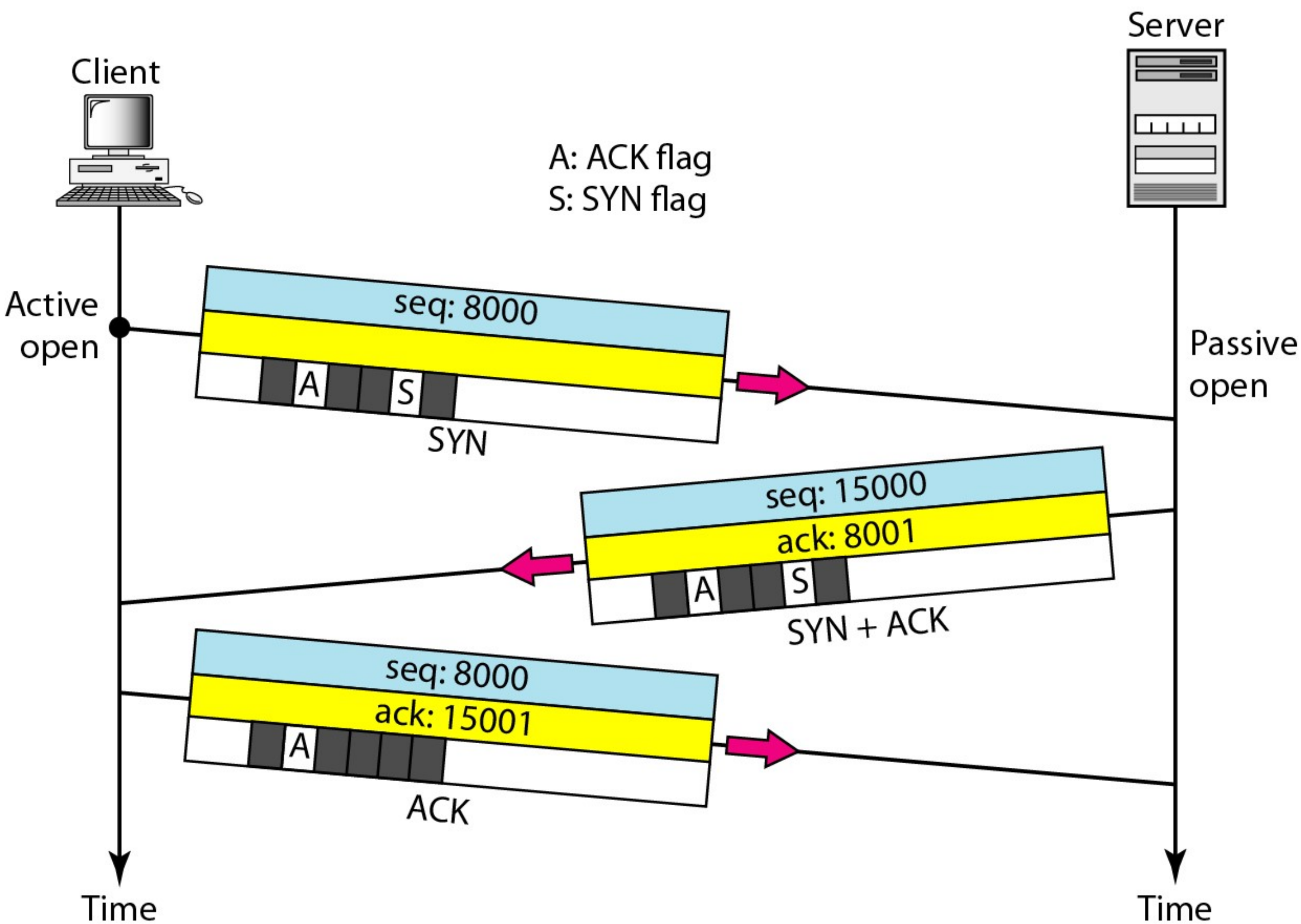
URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



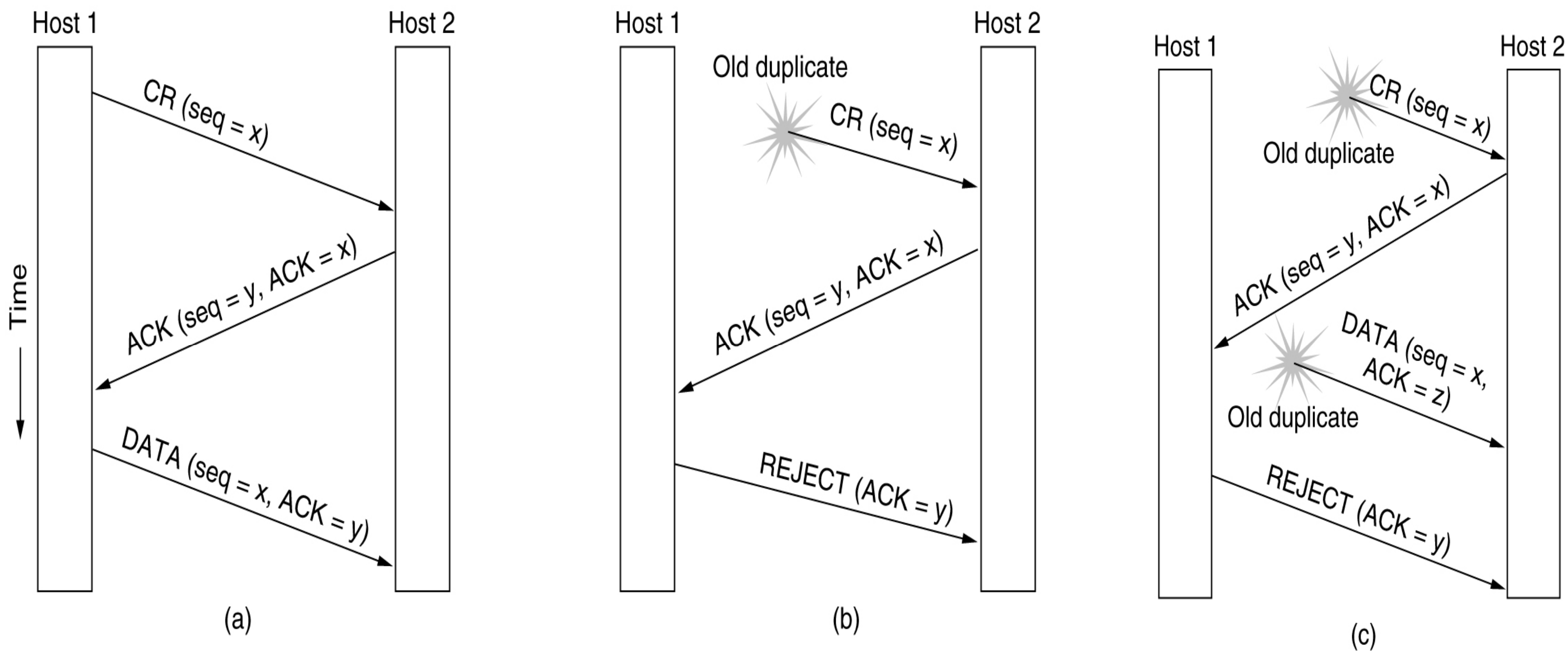
Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- Connection release,
- Flow control and buffering,
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

TCP Connection establishment using three-way handshaking



Connection Establishment -3 Scenarios



Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.

(a) Normal operation,

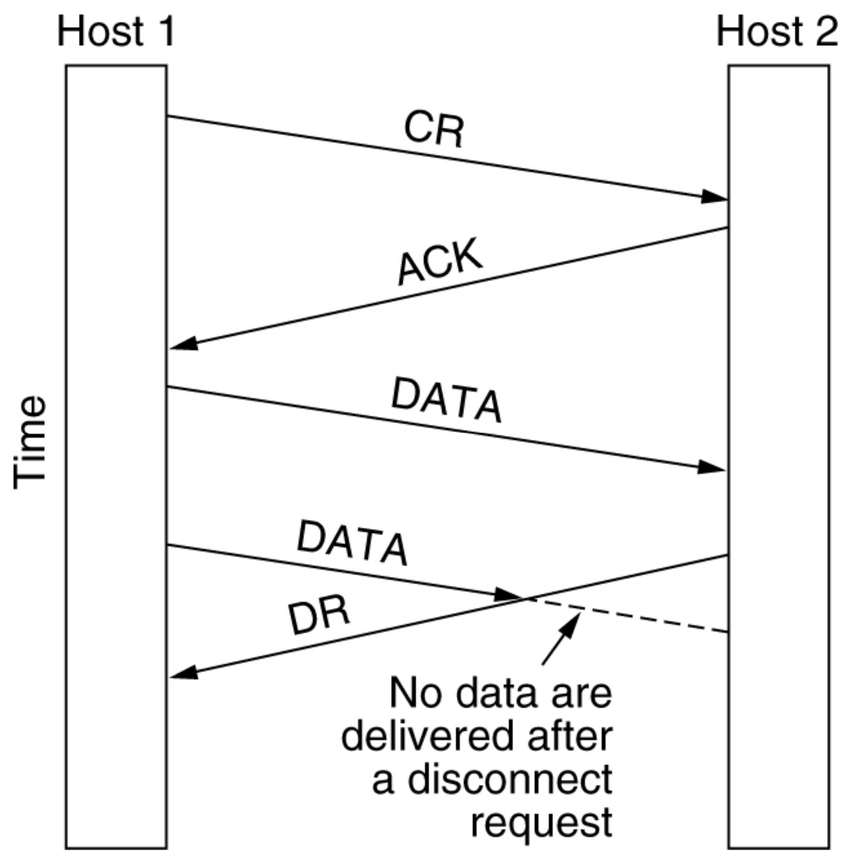
(b) Old CONNECTION REQUEST appearing out of nowhere.

(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Outline

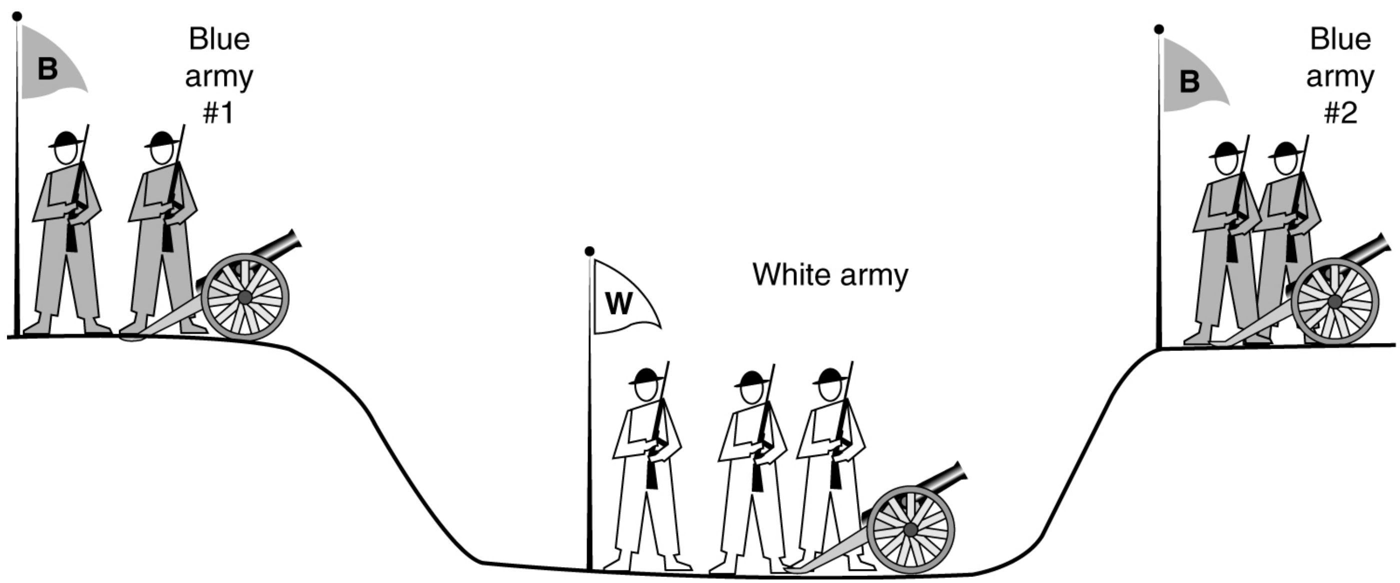
- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- Flow control and buffering,
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

TCP- Connection Release

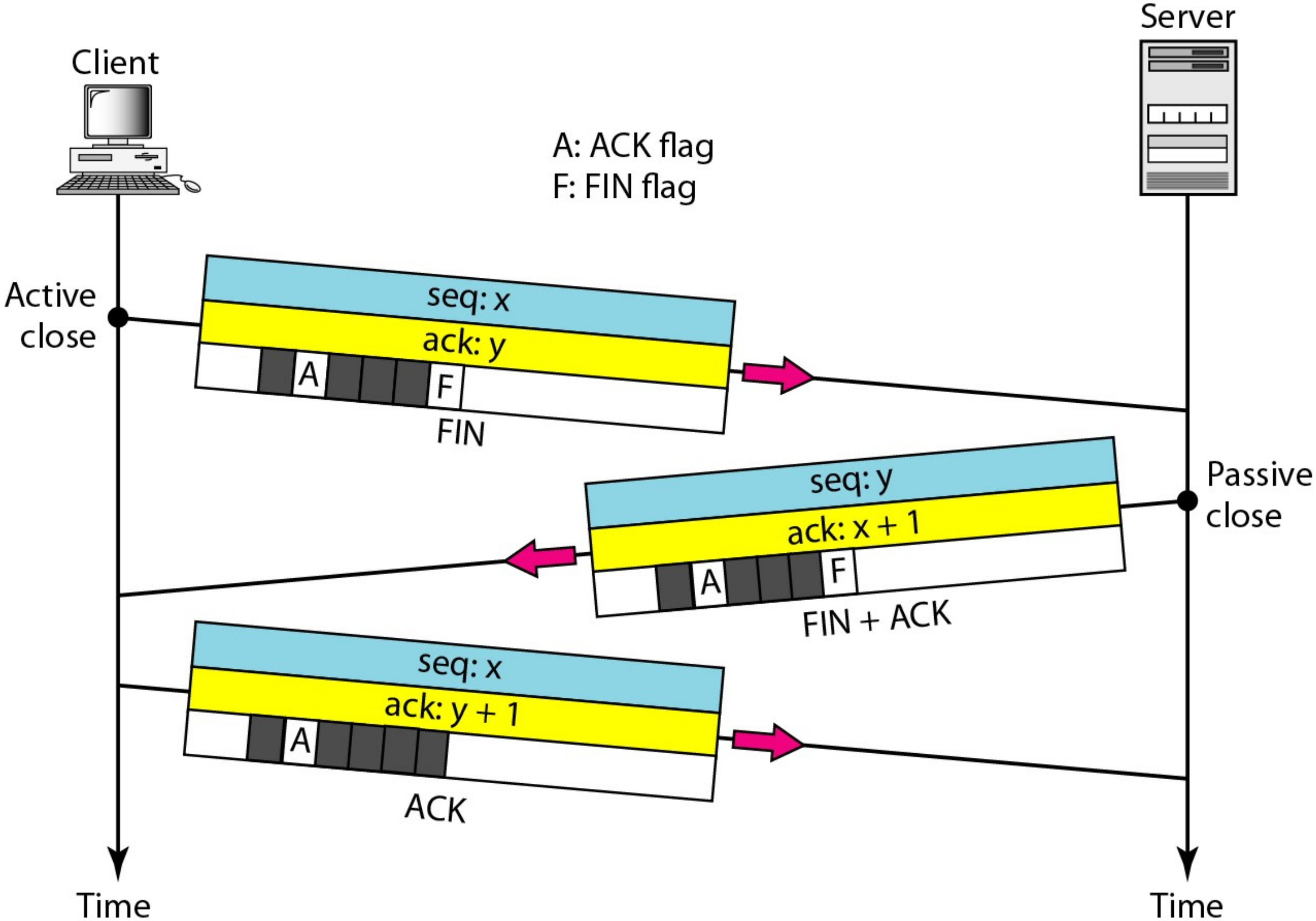


Abrupt disconnection with loss of data.

TCP- Connection Release- The two-army problem.

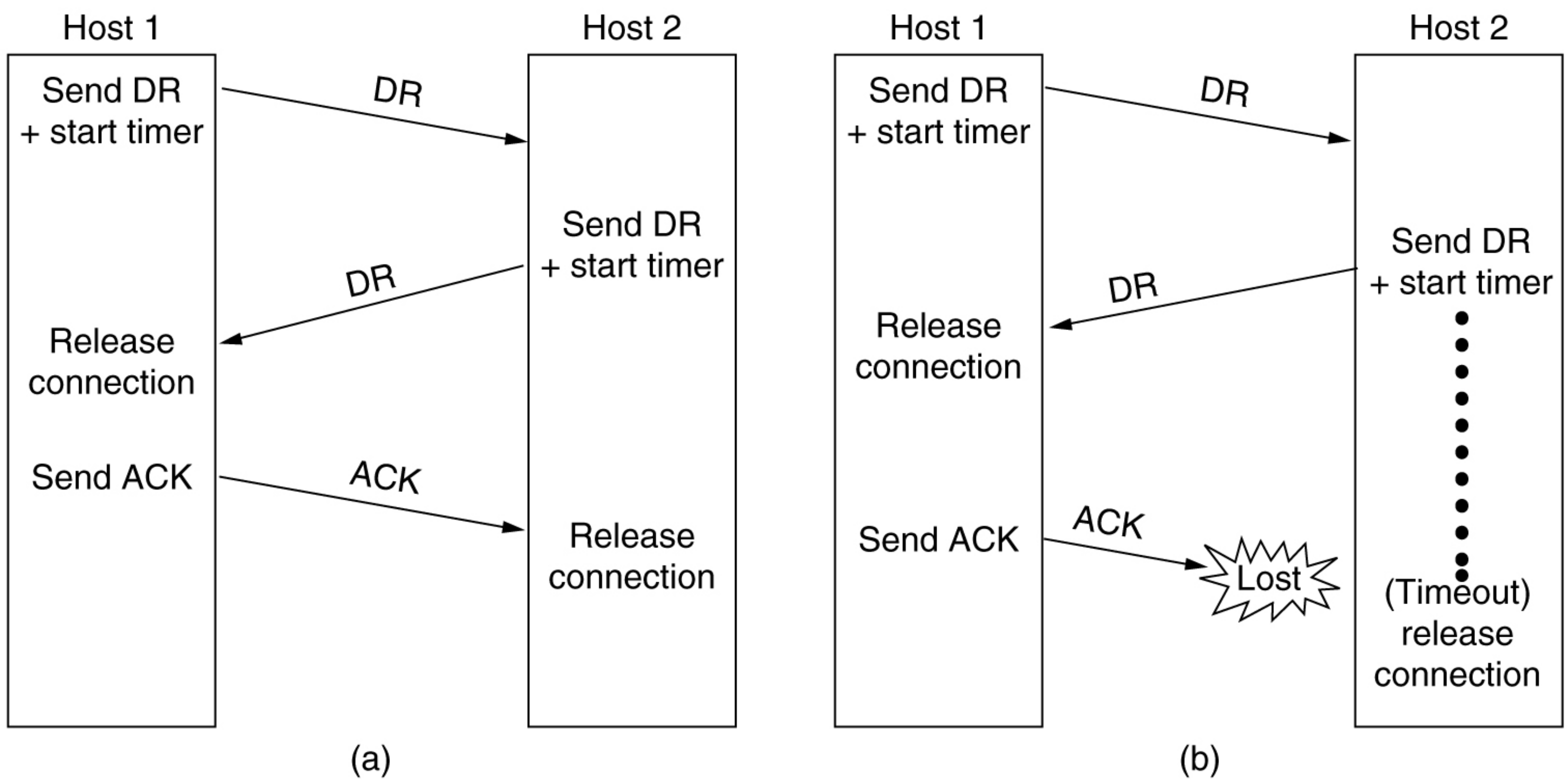


TCP Connection termination using three-way handshaking



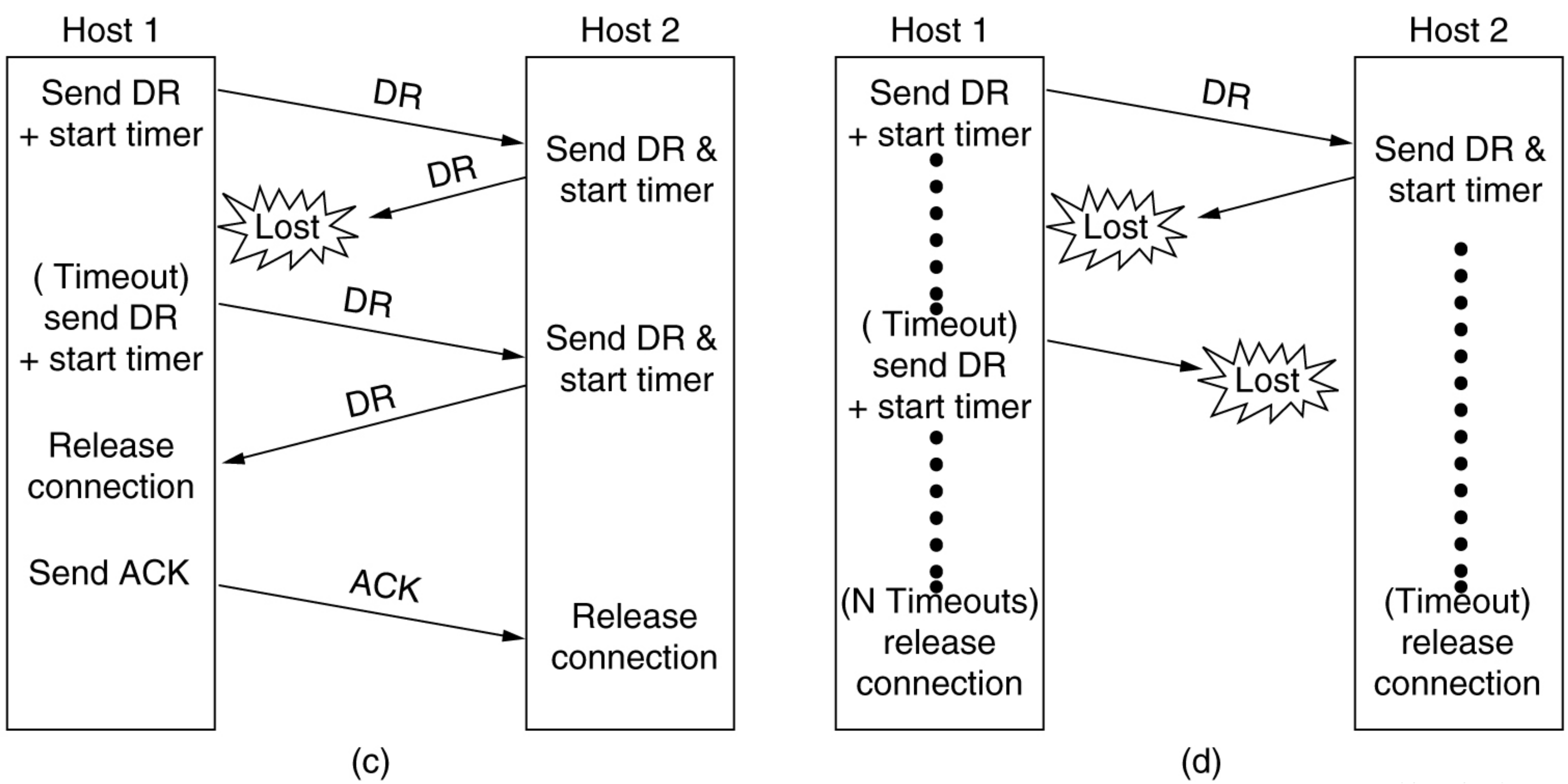
Connection Release Scenarios

Four protocol scenarios for releasing a connection. (a) Normal case of a three-way handshake. (b) final ACK lost.



Connection Release Scenarios…

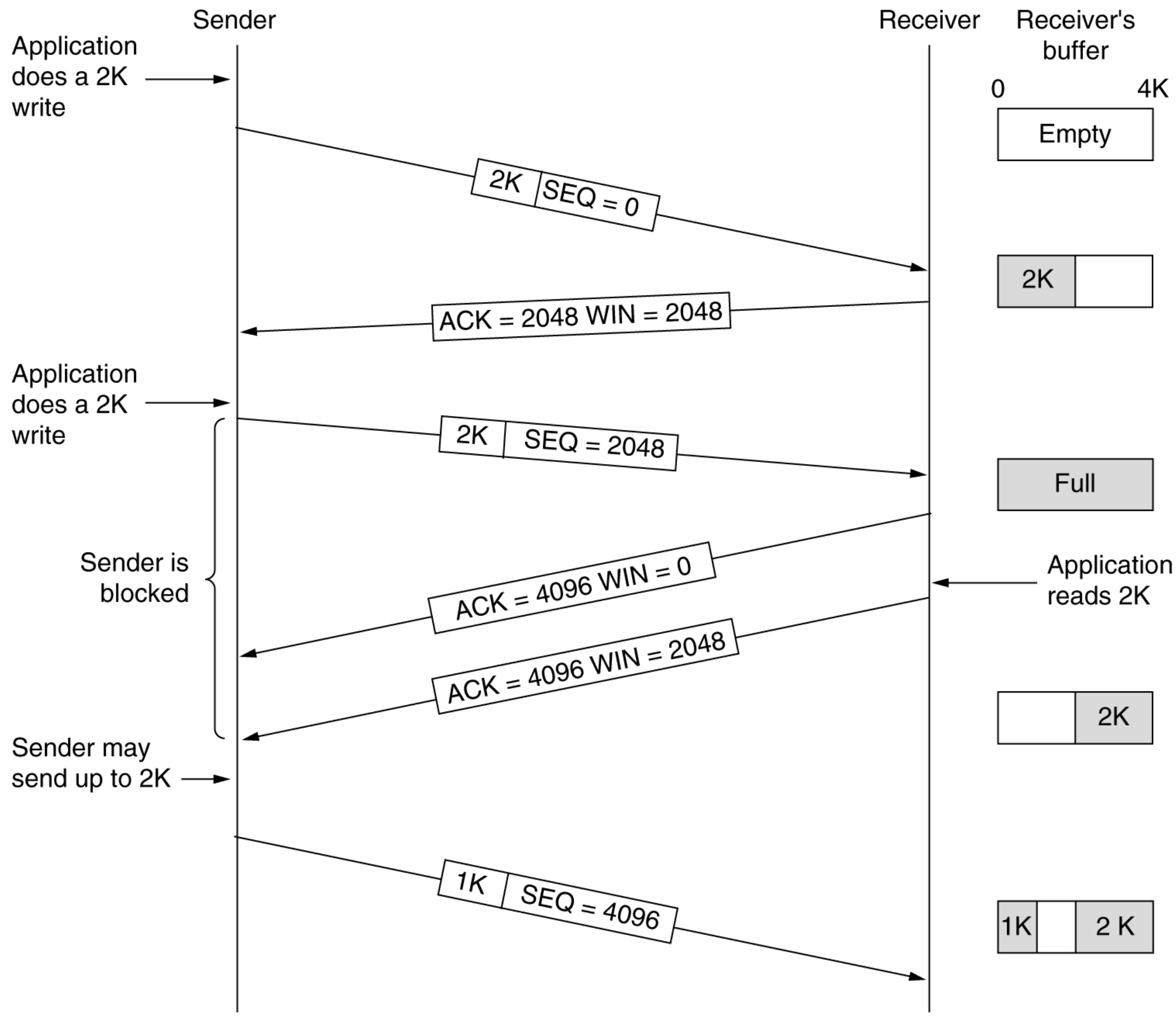
(c) Response lost. (d) Response lost and subsequent DRs lost.



Outline

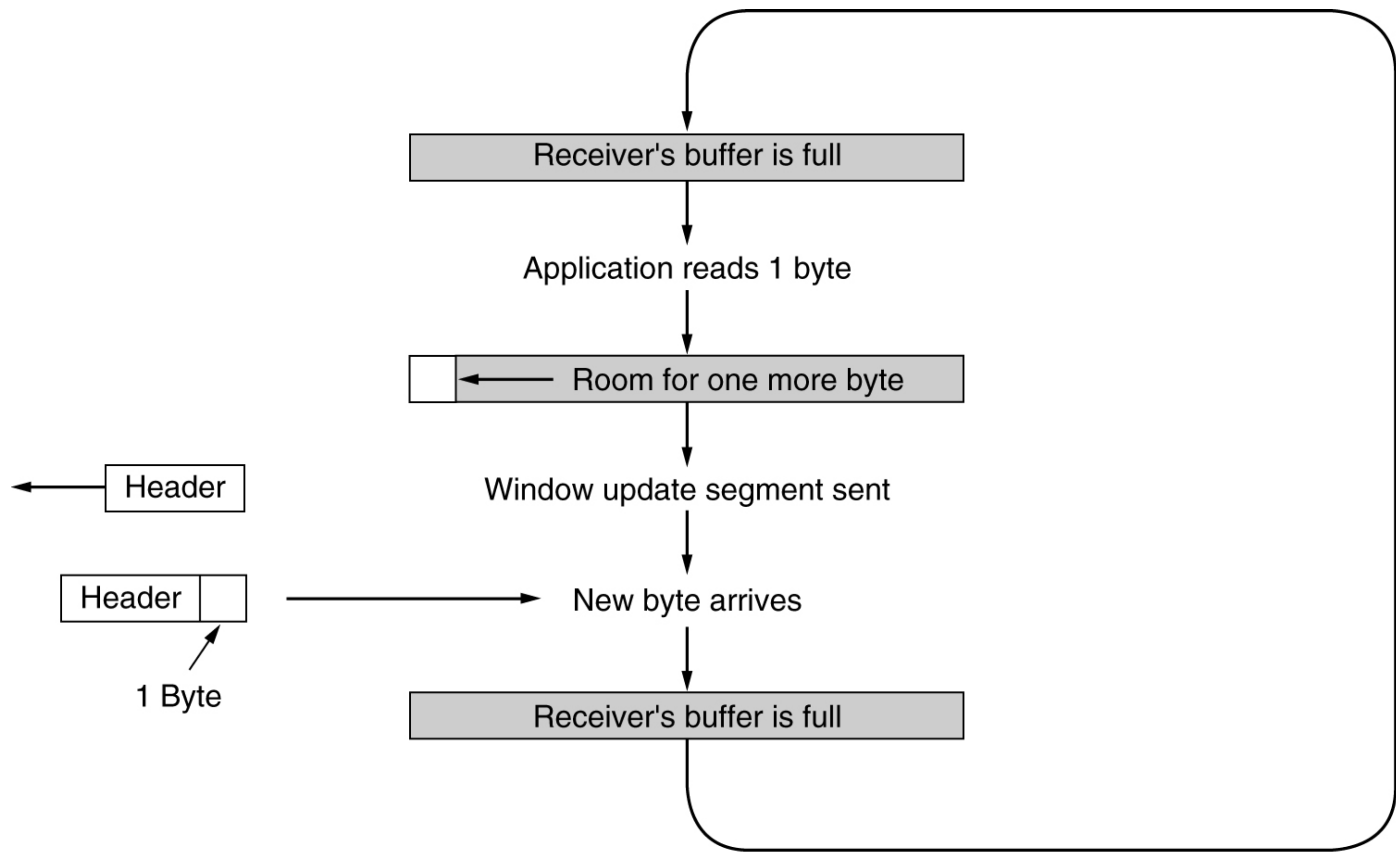
- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- TCP Timer management,
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

TCP Transmission Policy(Flow control)



Window management in TCP.

Silly window syndrome Problem



Silly window syndrome.

Solution to Silly window syndrome Problem

There are two solutions

- Nagle's solution
- Clark's solution

Nagle's algorithm

Purpose is to allow the **sender** TCP to make efficient use of the network, while still being responsive to the sender applications.

Idea:

If application data comes in byte by byte, send first byte only. Then *buffer all application data till until ACK for first byte comes in*.

If network is slow and application is fast, the second segment will contain a lot of data.

Send second segment and buffer all data till ACK for second segment comes in.

An exception to this rule is to always send (not wait for ACK) if enough data for half the receiver window or MSS(Maximum segment size) is accumulated.

Clark's algorithm

Purpose is to prevent the **receiver** from sending a window update for 1byte.

Idea:

Receiver is forced to wait until it has a decent amount of space available

The receiver should not send a window update until it can handle the maximum segment size it declared when the connection was established or until its buffer is half empty, whichever is smaller

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- TCP Congestion Control,
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

TCP timers- TCP uses 4 timers to do its work

The most important of these is the RTO (Retransmission TimeOut). When a segment is sent, a retransmission timer is started. If the segment is acknowledged before the timer expires, the timer is stopped. If, on the other hand, the timer goes off before the acknowledgement comes in, the segment is retransmitted (and the timer os started again).

Persistence timer- It is designed to prevent the following deadlock.

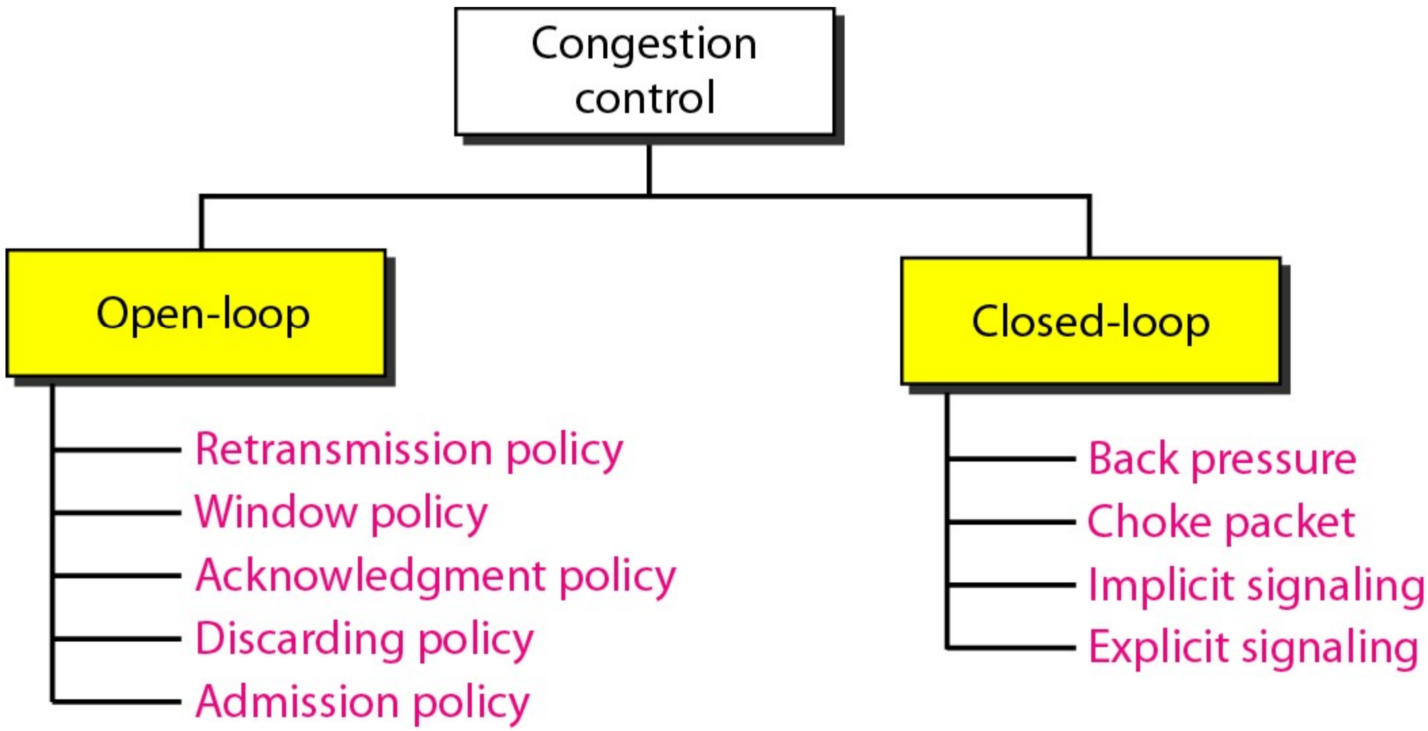
A third timer that some implementations use is the keepalive timer. When a connection has been idle for a long time, the keepalive timer may go off to cause one side to check whether the other side is still there.

The last timer used on each TCP connection is the one used in the TIME WAIT state while closing. It runs for twice the maximum packet lifetime to make sure that when a connection is closed; all packets created by it have died off.

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- **TCP Congestion Control,**
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

Congestion control policies



TCP congestion control

We looked at how TCP handles flow control. In addition we know the congestion happens. The only real way to handle congestion is for the sender to reduce sending rate.

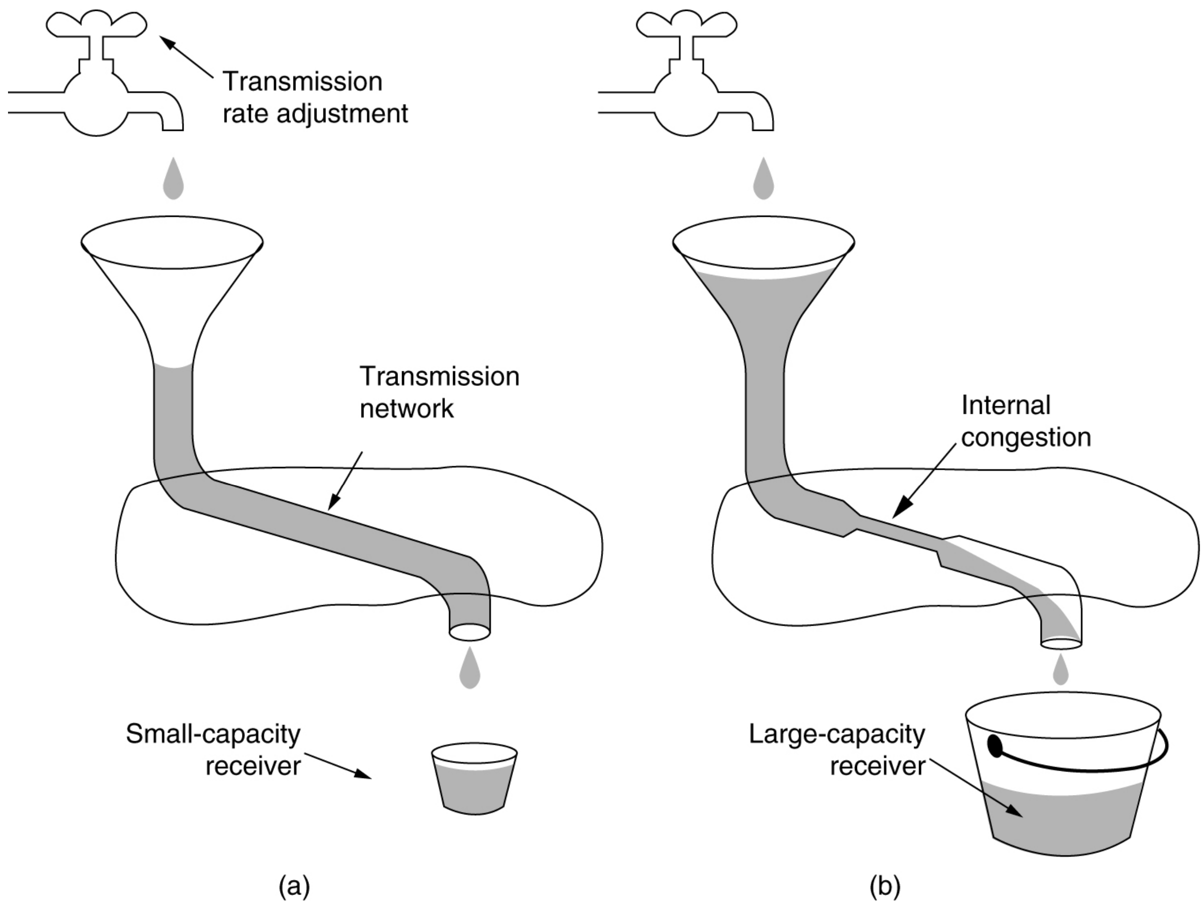
So how does on detect congestion ?

In old days, packets were lost due to transmission errors and congestion. But nowadays, transmission errors are very rare (except for wireless). So, TCP assumes a lost packet as an indicator of congestion.

So does TCP deal with congestion ?

It maintains an indicator of network capacity, called the ***congestion window***

TCP Congestion Control



(a) A fast network feeding a low capacity receiver.
(b) A slow network feeding a high-capacity receiver.

TCP congestion control

In essence TCP deals with two potential problems separately:

Problem		*Solution*
Receiver capacity	→	**Receiver window** (rwnd)
Network capacity	→	**Congestion window** (cwnd)

Each window reflect the number of bytes the sender may transmit. The sender sends the *minimum of these two sizes*. This size is the ***effective window***.

TCP Congestion Control – 3 Stages

TCP uses these stages in updating cwnd.

1. ***Slow start***: Initial state. Rapidly grow cwnd

2. ***Congestion avoidance***: Slowly grow cwnd.

3. ***Congestion detection*** : *(Multiplicative decrease)*

TCP Congestion Control – Slow start

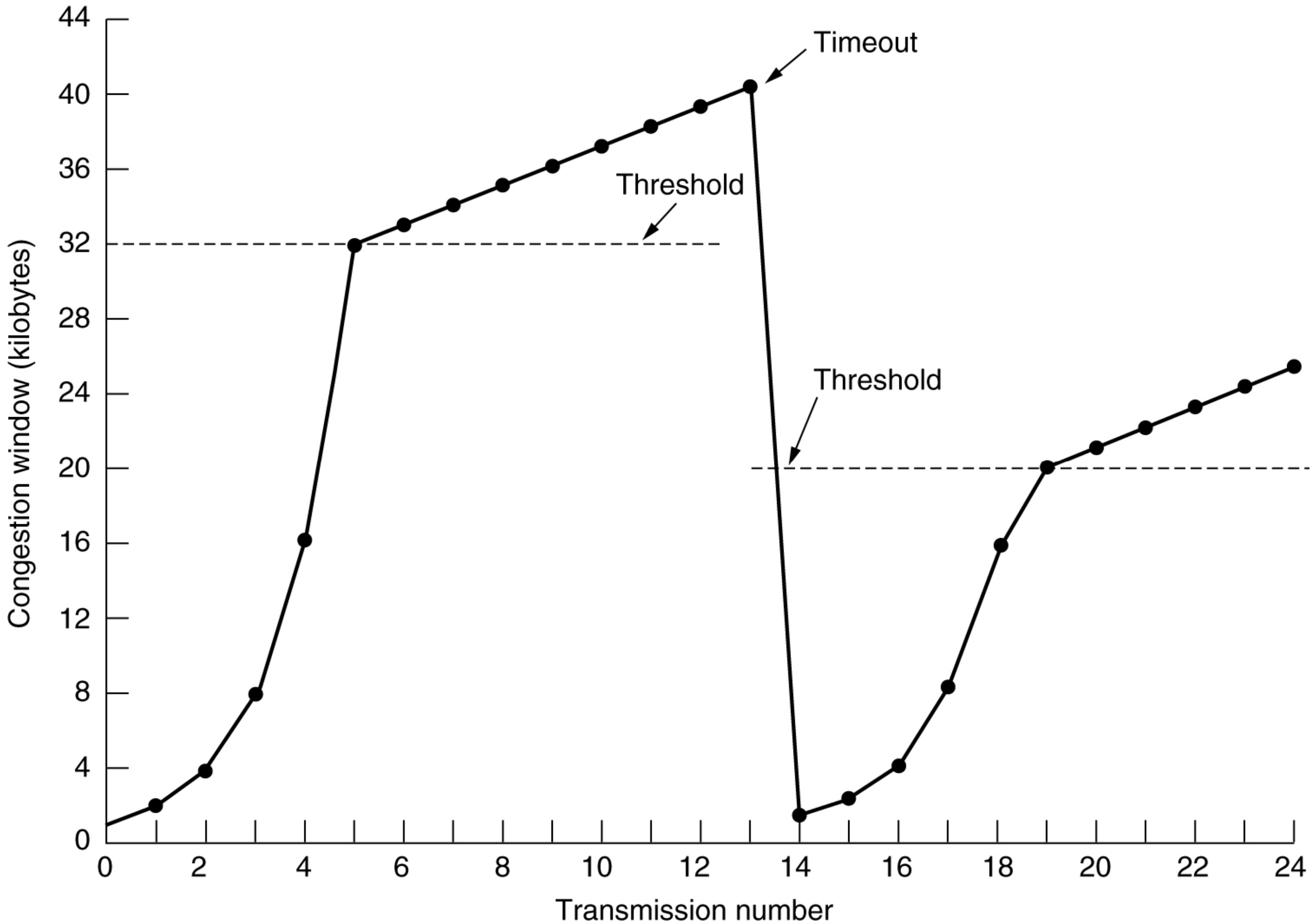
When connection is established , the sender initializes the congestion window to the size of the maximum segment in use on the connection.

It then sends the one maximum segment

If this segment is acknowledged before timeout occurs then it doubles the segment size

This is continued until the timeout occurs or receivers window size is reached

TCP Congestion Control



An example of the Internet congestion algorithm.

TCP Congestion Control-Congestion Avoidance

When the size of congestion window reaches the slow start threshold, the slow start phase stops and the additive phase begins.

TCP Congestion Control-Congestion Detection

If congestion occurs the congestion window size must be decreased.

That means when a timer time outs or when 3 Acks are received the size of the threshold is dropped to ½ (multiplicative decrease)

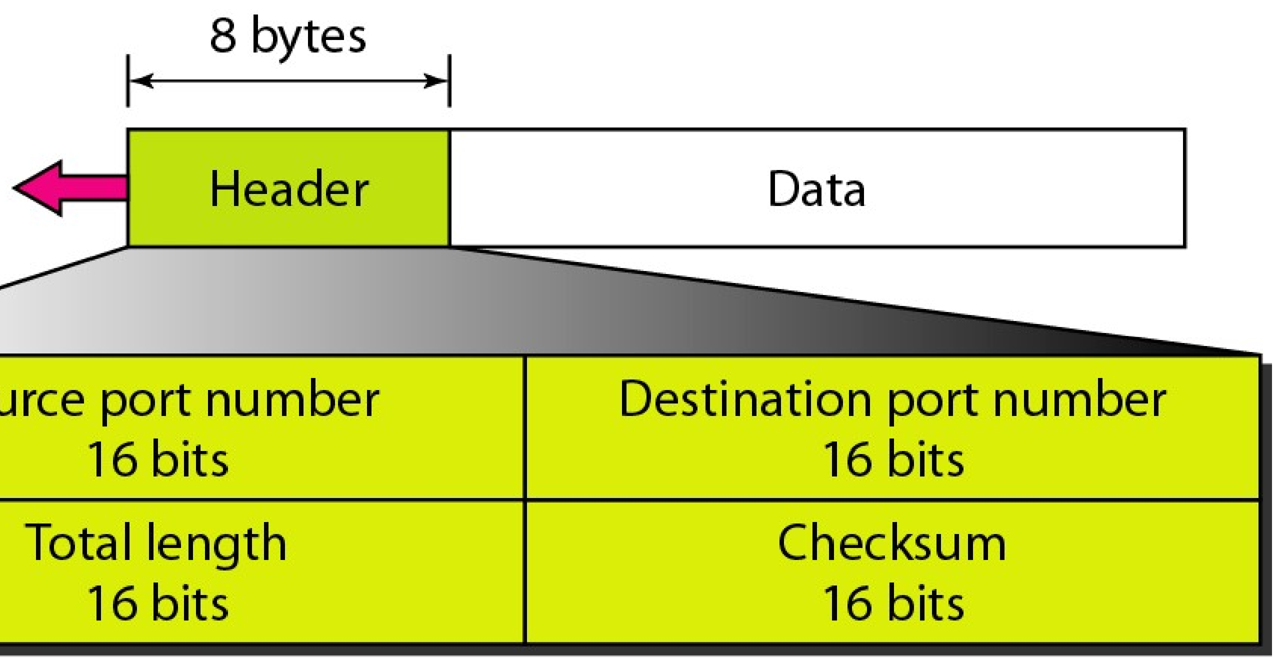
Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- **TCP Congestion Control,**
- **UDP – (Gap Analysis)** ***content beyond syllabus***
- Real Time Transport protocol(RTP),
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

USER DATAGRAM PROTOCOL (UDP)

- *The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.*

User datagram format (UDP Header Format)



UDP Pseudo Header

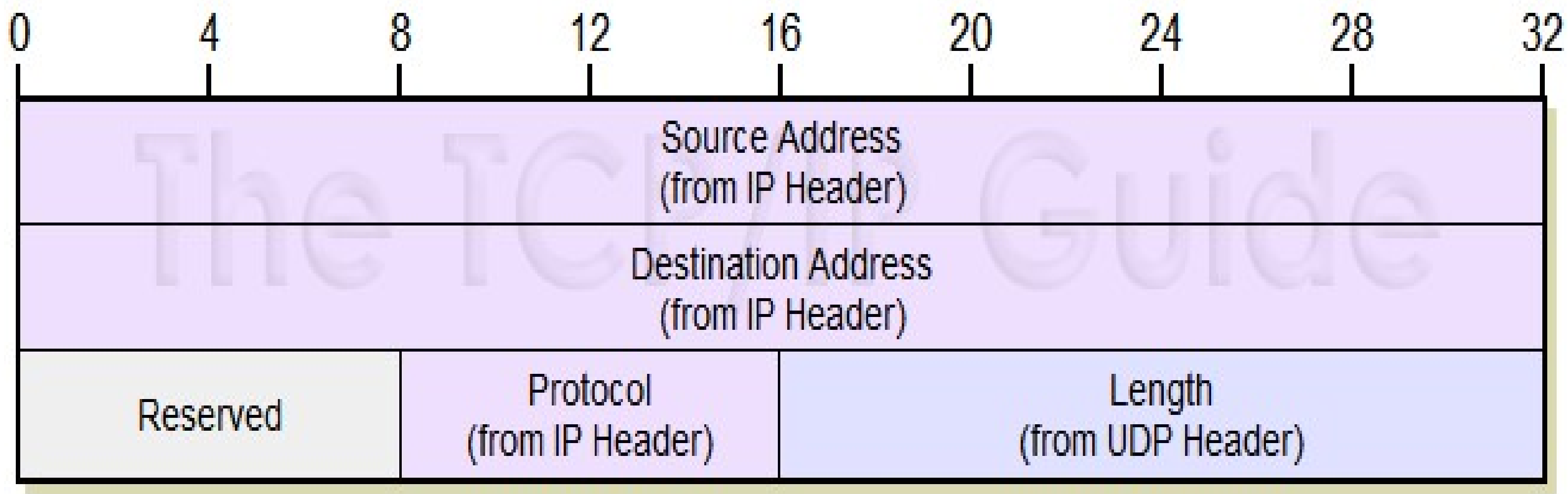
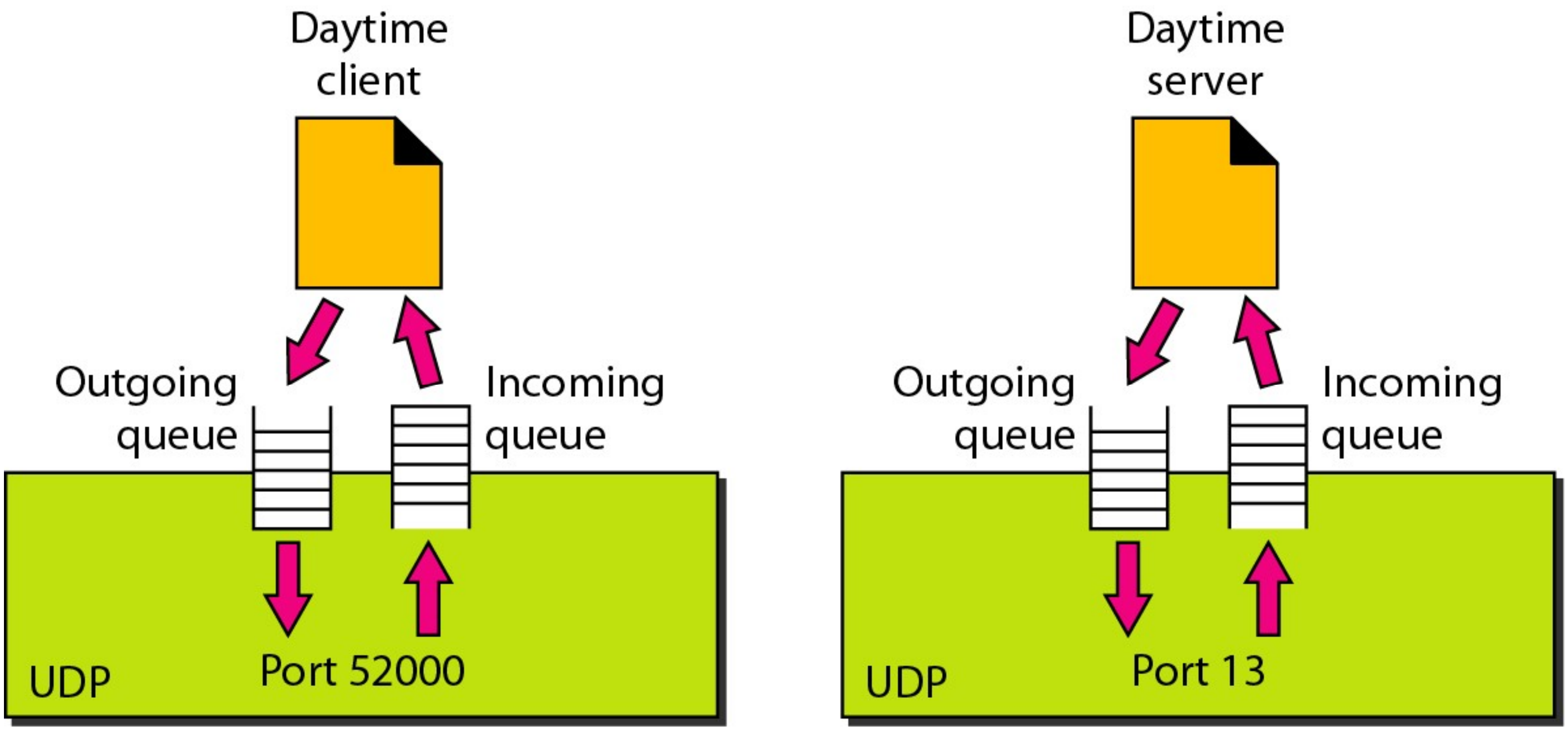


Figure : UDP Pseudo Header Format

UDP Operations

- Connectionless service
- No Flow and error control except checksum
- Encapsulation and Decapsulation of messages in IP datagram
- Queing

Queues in UDP



Uses of UDP

- Simple Request reply communication
- Suitable for process with internal flow and control mechanisms. Eg. TFTP
- The Real-Time Transport Protocol
- Used in route updating protocol like Routing Information Protocol(RIP)
- Remote Procedure Call(RPC)
- Suitable for Multicasting. Multicasting capability is inbuilt in UDP software's

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- **TCP Congestion Control,**
- **UDP – (Gap Analysis)** ***content beyond syllabus***
- **Real Time Transport protocol(RTP),**
- Stream Control Transmission Protocol (SCTP),
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

RTP: A Transport Protocol for Real-Time Applications

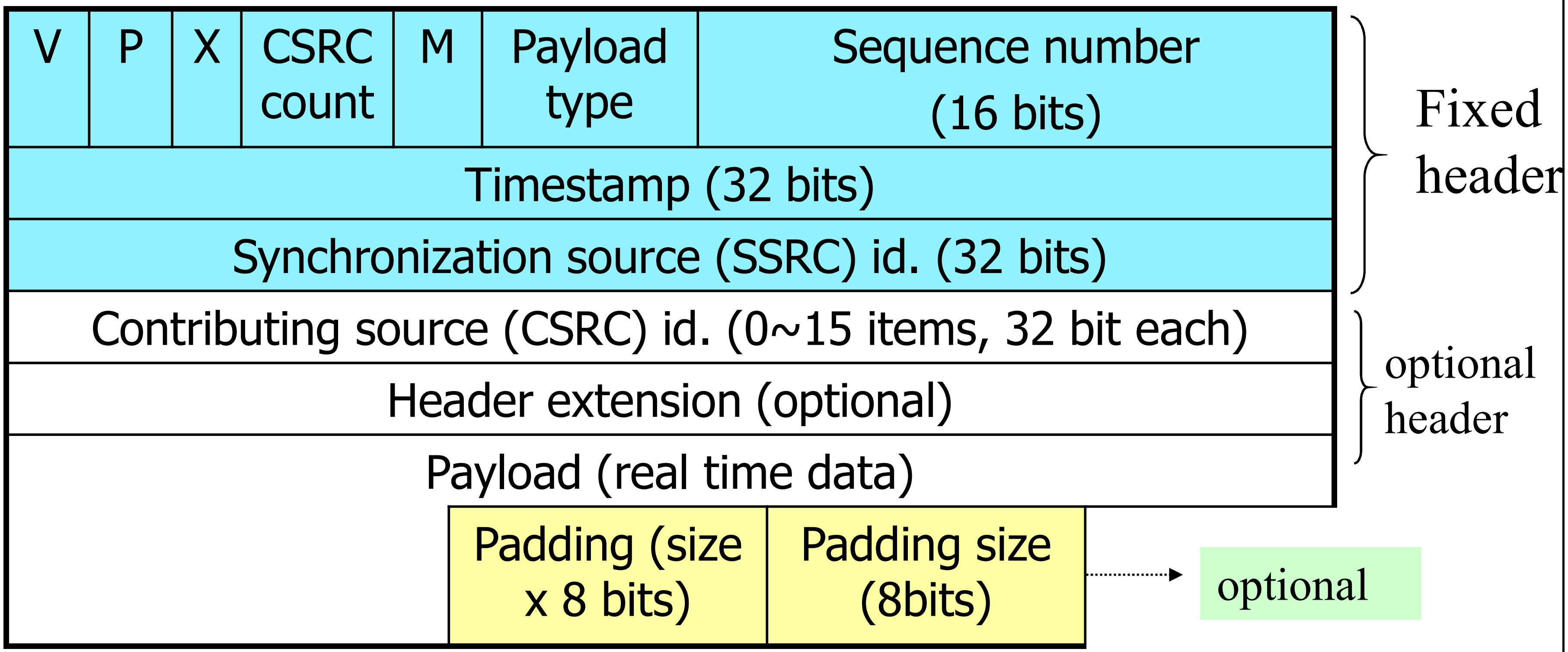
- Internet standard for real-time data
  - Interactive audio, video, and simulation data
- Primarily designed for multi-user multimedia conference
  - Session management
  - Scalability considerations
- Provides end-to-end transport functions for real-time applications
  - Payload type identification
  - Sequence numbering
  - Timestamping
  - Delivery monitoring

Introduction – cont.

- Containing two closely linked parts: data + control
  - **RTP: Real-time transport protocol**
    - Carry real-time data
  - **RTCP: RTP control protocol**
    - QoS monitoring and feedback
    - Session control
- Architecture

<table>
<tr><td colspan="2">Applications</td></tr>
<tr><td colspan="2">RTP & RTCP</td></tr>
<tr><td rowspan="2">Other transport and network protocols</td><td>UDP</td></tr>
<tr><td>IP</td></tr>
</table>

RTP – packet format



- Version (V, 2bits): =2
- Padding(P, 1bit): If set, last byte of payload is padding size
- Extension(X, 1bit): If set, variable size header extension exists

RTP - header

- CSRC count (4 bits): number of Contributors, max 16 can be possible
- Marker (1 bit): defined in *profile*, mark end of data
- Payload type (7 bits): Audio/Video encoding scheme
- Sequence number: random initial value, increase by one for each RTP packet; for loss detection and seq. restoration
- SSRC: identify source; chosen randomly and locally; **collision** needs to be resolved
- CSRC list: id. of contributing sources, inserted by *mixer*

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- **TCP Congestion Control,**
- **UDP – (Gap Analysis)** ***content beyond syllabus***
- **Real Time Transport protocol(RTP),**
- **Stream Control Transmission Protocol (SCTP),**
- Quality of Service (QoS),
- Differentiated services,
- TCP and UDP for Wireless.

SCTP

- *Stream Control Transmission Protocol (SCTP) is a*
  - *reliable,*
  - *message-oriented*
  - *transport layer protocol.*
- *SCTP, however, is mostly designed for Internet applications that have recently been introduced.*
- *These new applications need a more sophisticated service than TCP can provide.*

SCTP is a message-oriented, reliable protocol that combines the best features of UDP and TCP.

Comparison

- UDP: Message-oriented, Unreliable
- TCP: Byte-oriented, Reliable
- SCTP
  - Message-oriented, Reliable
  - Other innovative features
    - Association, Data transfer/Delivery
    - Fragmentation,
    - Error/Congestion Control

Some SCTP applications

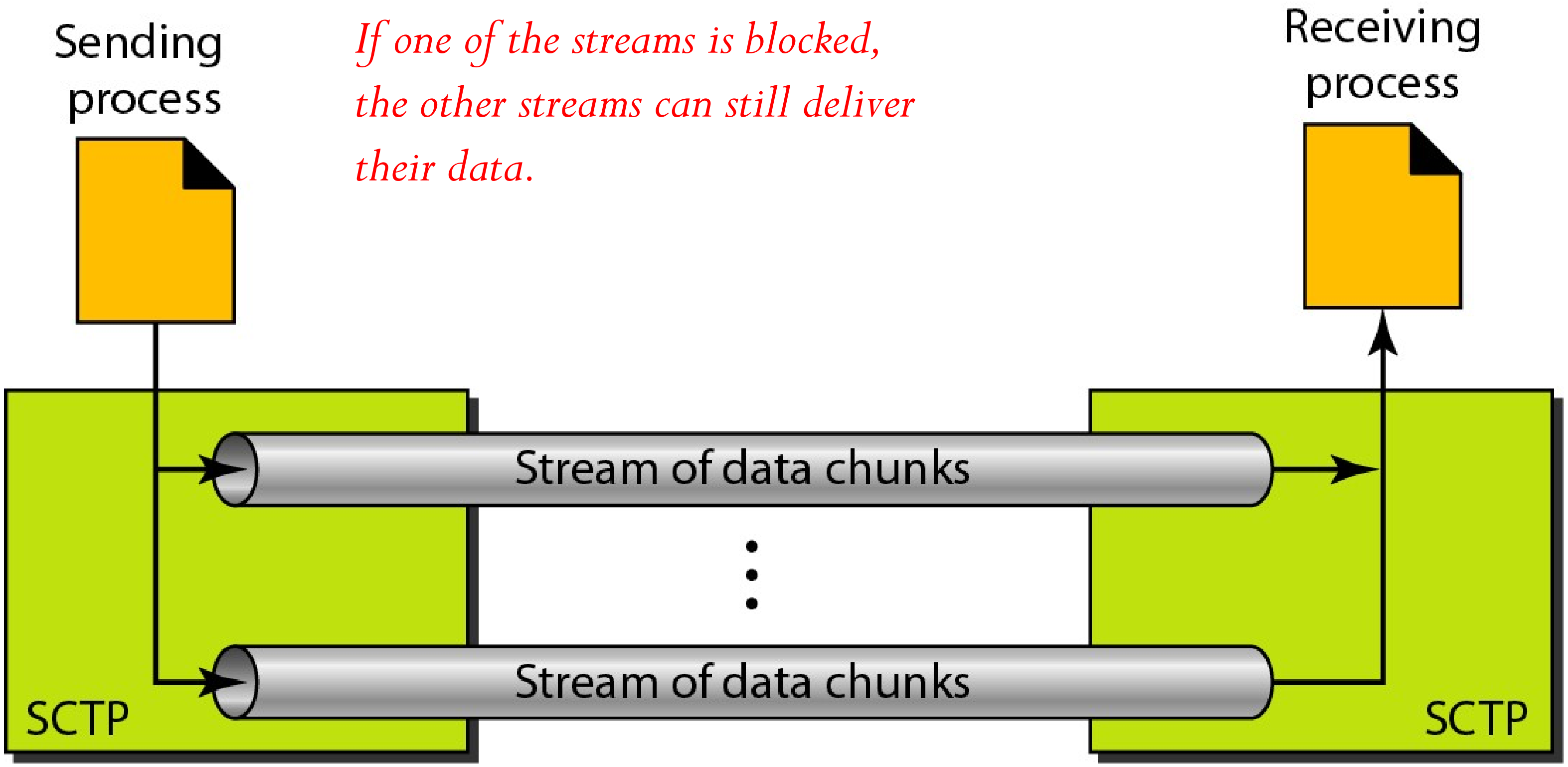
Protocol	*Port Number*	*Description*
IUA	9990	ISDN over IP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony

Services of SCTP

- Process-to-Process Communication
- Multiple Streams
- Multihoming
- Full-Duplex Communication
- Connection-Oriented Service
- Reliable Service

Multiple-stream concept

If one of the streams is blocked, the other streams can still deliver their data.

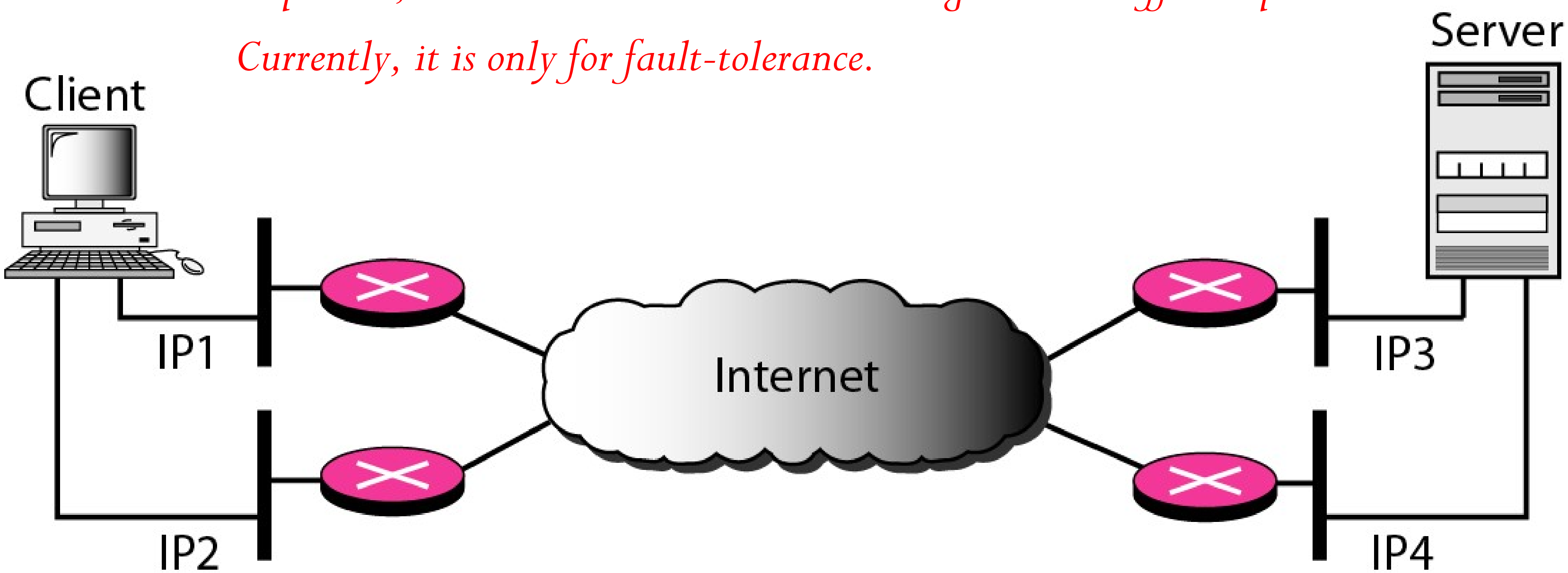


An association in SCTP can involve multiple streams.

Multihoming concept

At present, SCTP does not allow load sharing between different path.

Currently, it is only for fault-tolerance.



SCTP Features

- Transmission Sequence Number (TSN)
- Stream Identifier (SI)
- Stream Sequence Number (SSN)
- Packets
- Acknowledgment Number
- Flow Control
- Error Control
- Congestion Control

In SCTP, a data chunk is numbered using a TSN.

To distinguish between different streams, SCTP uses an SI.

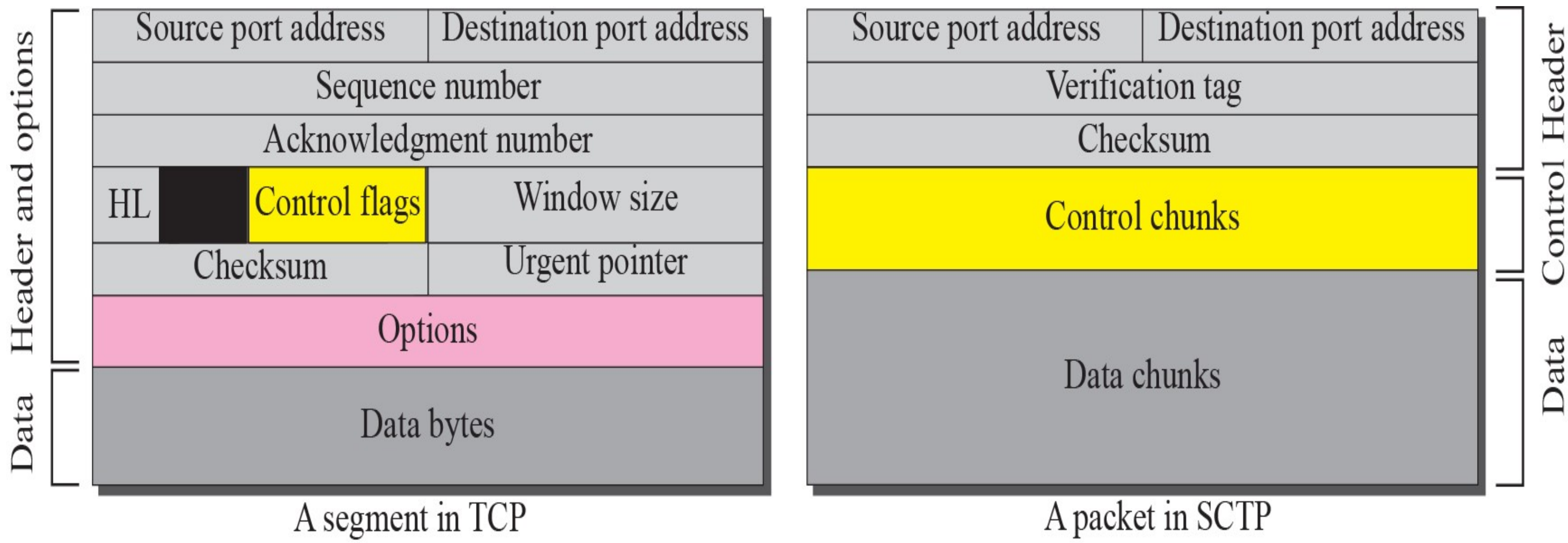
To distinguish between different data chunks belonging to the same stream, SCTP uses SSNs.

Comparison between UDP,TCP and SCTP

UDP	TCP	SCTP
Message oriented protocol	Byte oriented protocol	Message oriented protocol
Preserve message boundaries	Does not Preserve message boundaries	Preserve message boundaries
Unreliable	Reliable	Reliable
No congestion and flow control	Have congestion and flow control	Have congestion and flow control
Each message follows different route so no sequencing	Each message follows same route so have in sequence data delivery	have in sequence data delivery
Port no 17	Port no 6	Port no 132

Comparison between a TCP segment and an SCTP packet

TCP has segments; SCTP has packets.

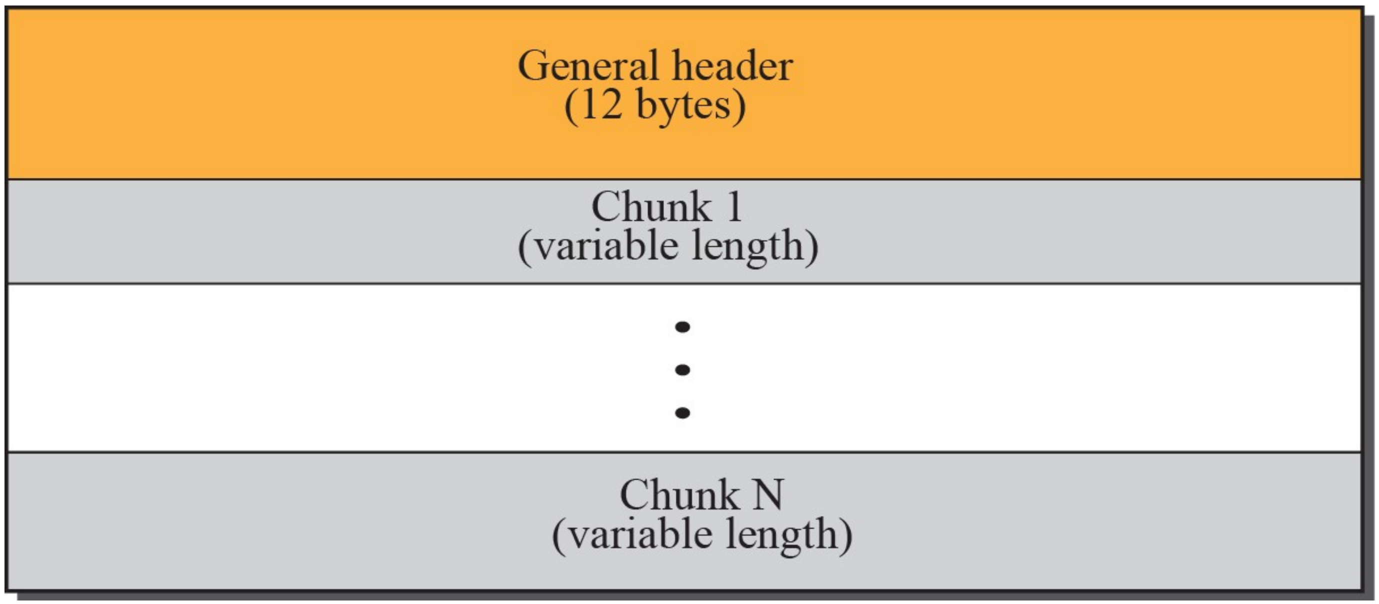


SCTP PACKET FORMAT

- In this section, we show the format of a packet and different types of chunks.
- An SCTP packet has a mandatory general header and a set of blocks called chunks.
- There are two types of chunks:

1. control chunks and
2. data chunks.

SCTP packet format



In an SCTP packet, control chunks come before data chunks.

General header (Common layout of a chunk)

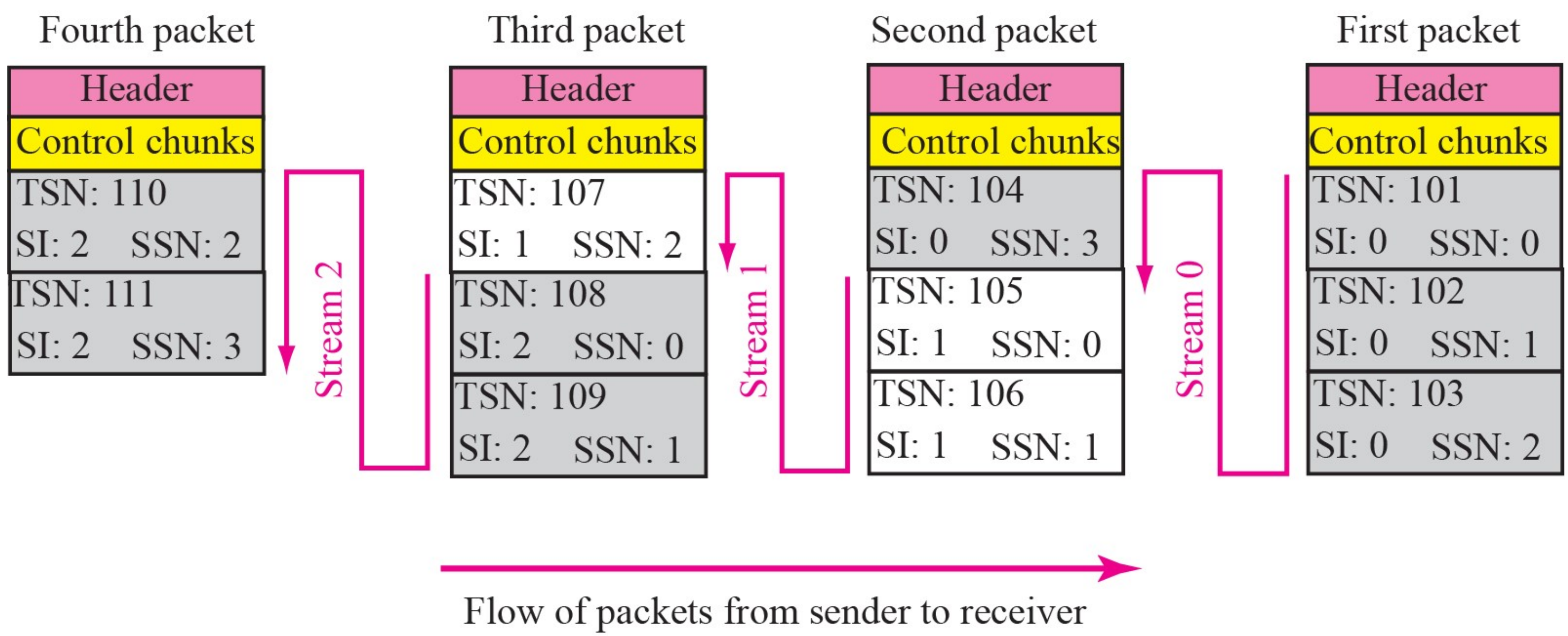
Source port address 16 bits	Destination port address 16 bits
Verification tag 32 bits	
Checksum 32 bits	

In SCTP, control information and data information are carried in separate chunks.

Data chunks are identified by three identifiers: TSN, SI, and SSN.
TSN is a cumulative number identifying the association; SI defines the stream; SSN defines the chunk in a stream.

In SCTP, acknowledgment numbers are used to acknowledge only data chunks; control chunks are acknowledged by other control chunks if necessary.

Packet, data chunks, and streams



Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- **TCP Congestion Control,**
- **UDP – (Gap Analysis)** ***content beyond syllabus***
- **Real Time Transport protocol(RTP),**
- **Stream Control Transmission Protocol (SCTP),**
- **Quality of Service (QoS),**
- Differentiated services,
- TCP and UDP for Wireless.

QoS Parameters

- Reliability
- Jitter
- Delay
- Bandwidth

Requirements

Reliability- *Reliability* is concerned with the ability of a *network* to carry out a desired operation according to its specifications

Jitter- **Jitter** is defined as a variation in the delay of received packets.

Delay- is the amount of time required to transmit packets.

Bandwidth- amount of information that can be transmitted over a **network** in a given amount of time

Requirements

How stringent the quality-of-service requirements are.

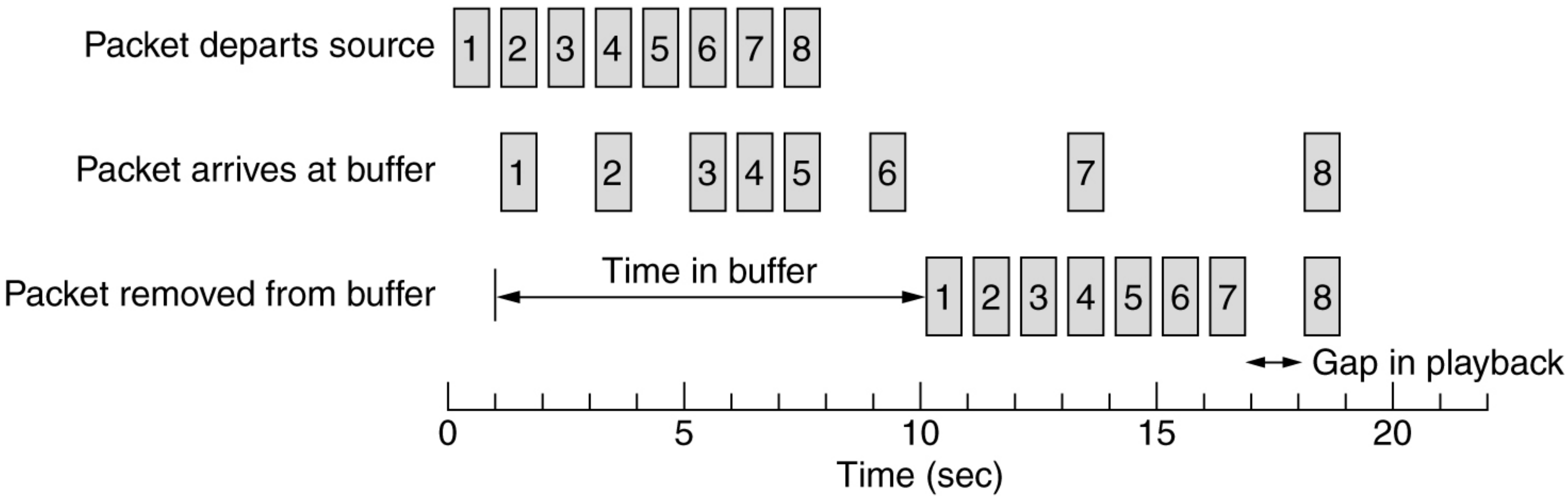
Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Techniques to achieve Good QoS

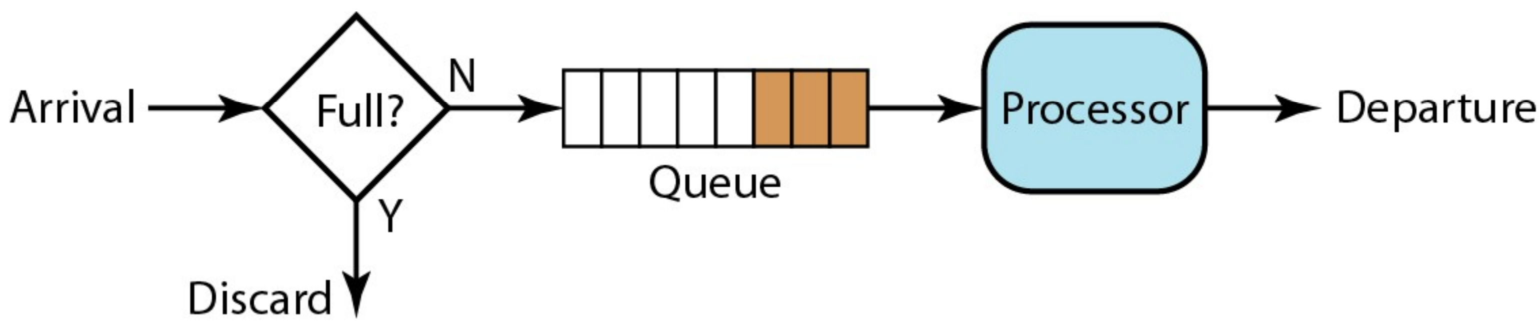
- Buffering
- Scheduling
  - *FIFO queue*
  - *Priority queuing*
  - *Weighted fair queuing*
- Traffic Shaping
  - *Leaky bucket algorithm*
  - *Token bucket algorithm*
- Resource reservation

Buffering

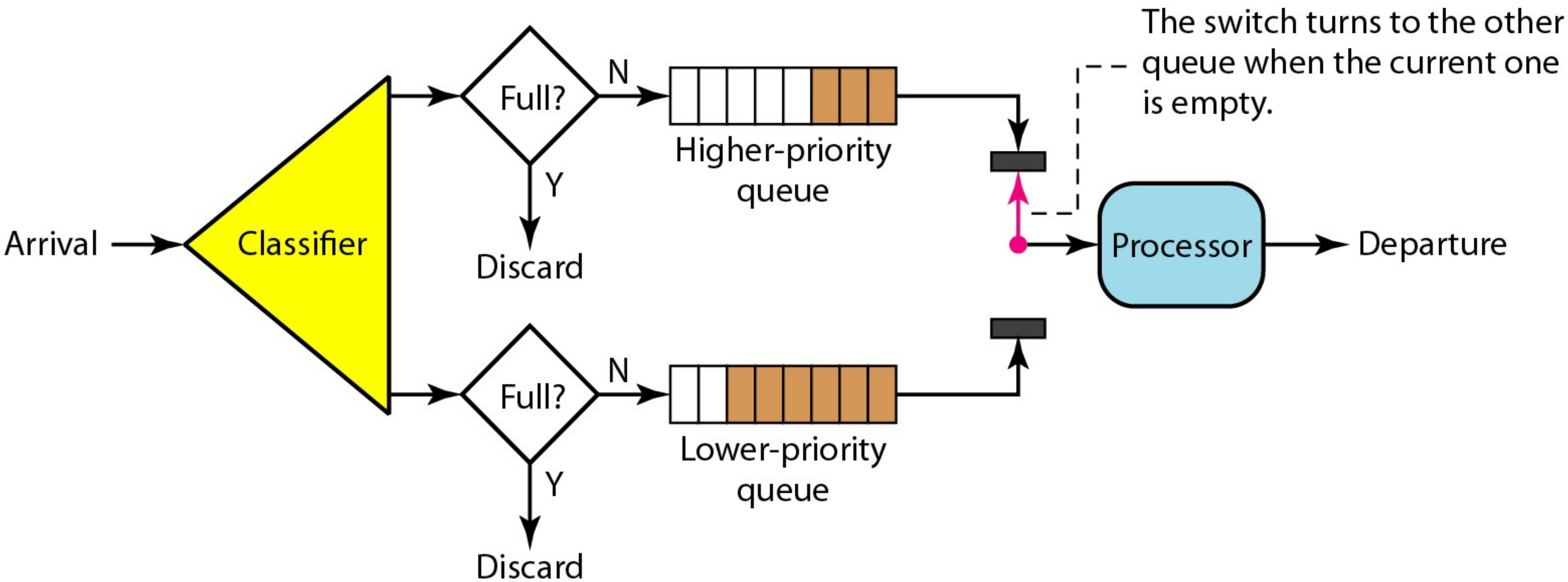
Smoothing the output stream by buffering packets.



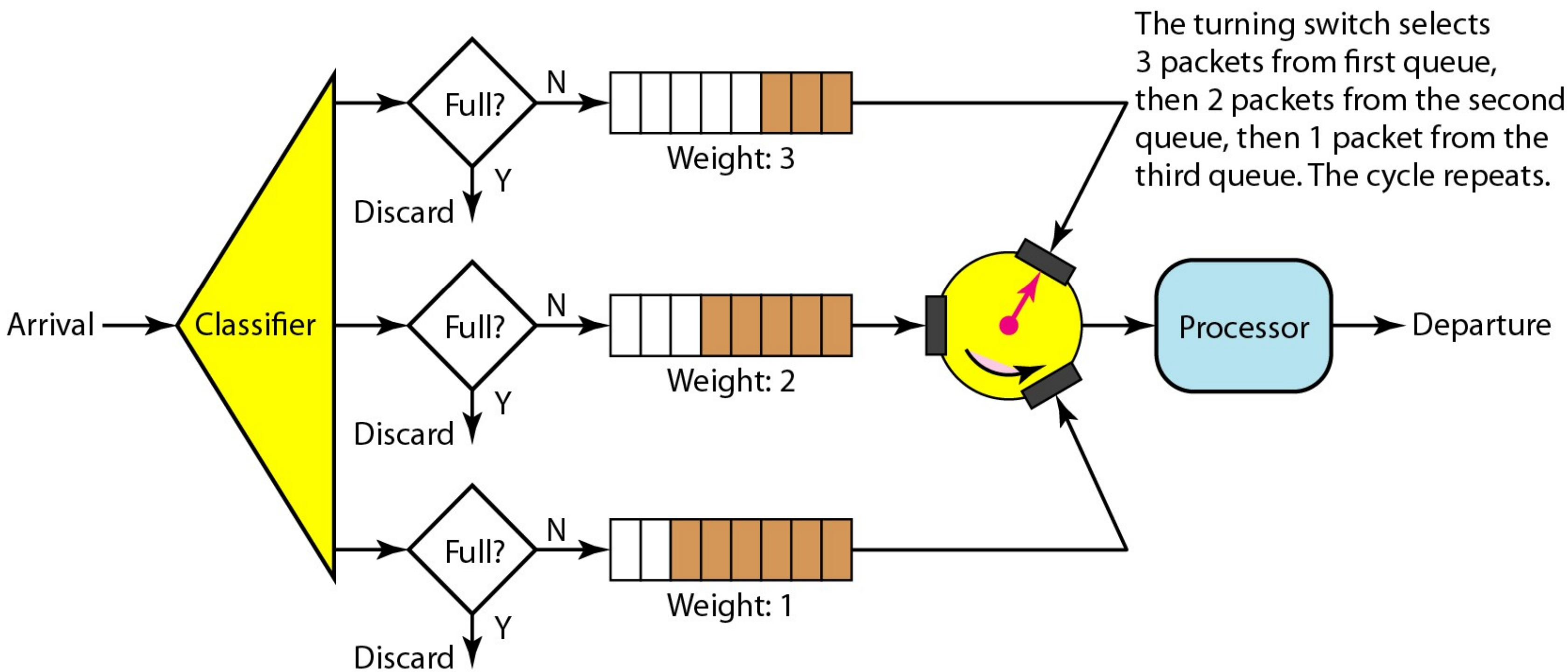
FIFO queue



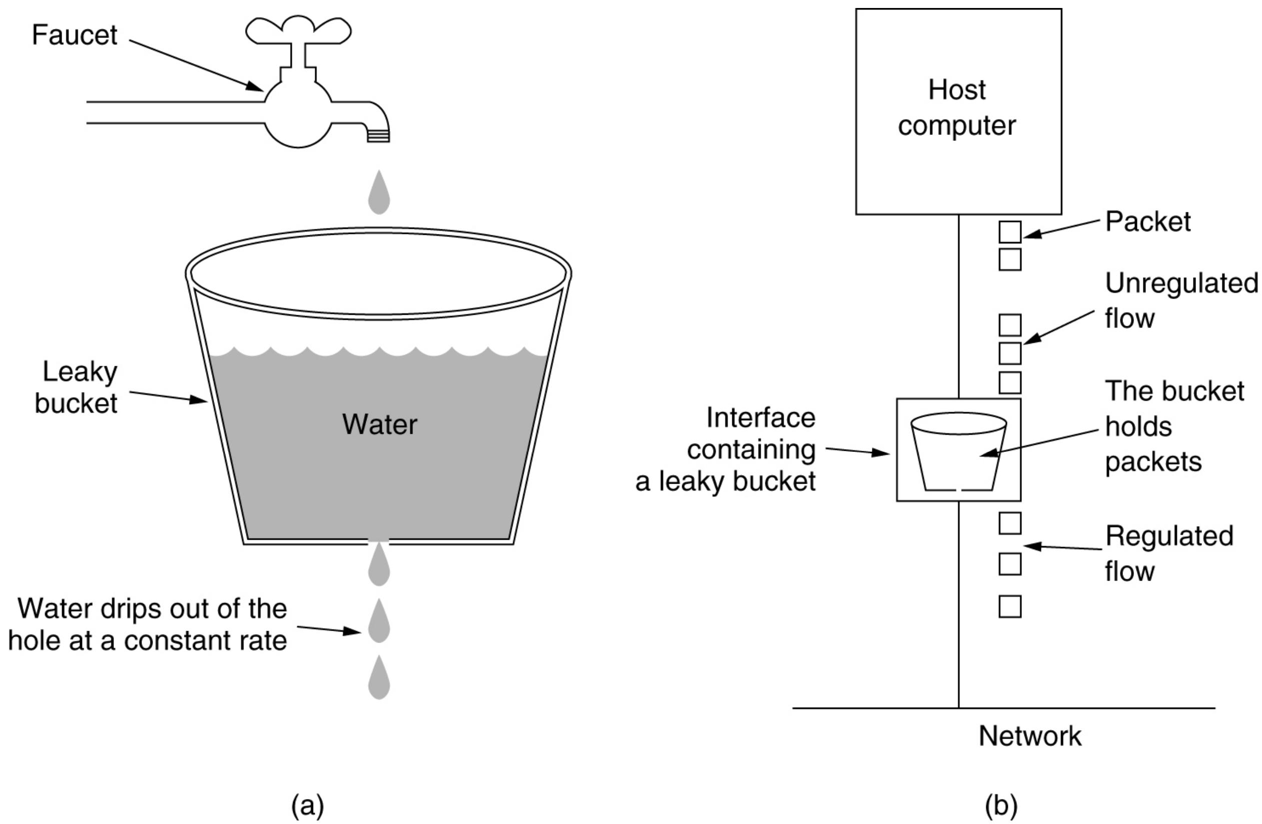
Priority queuing



Weighted fair queuing

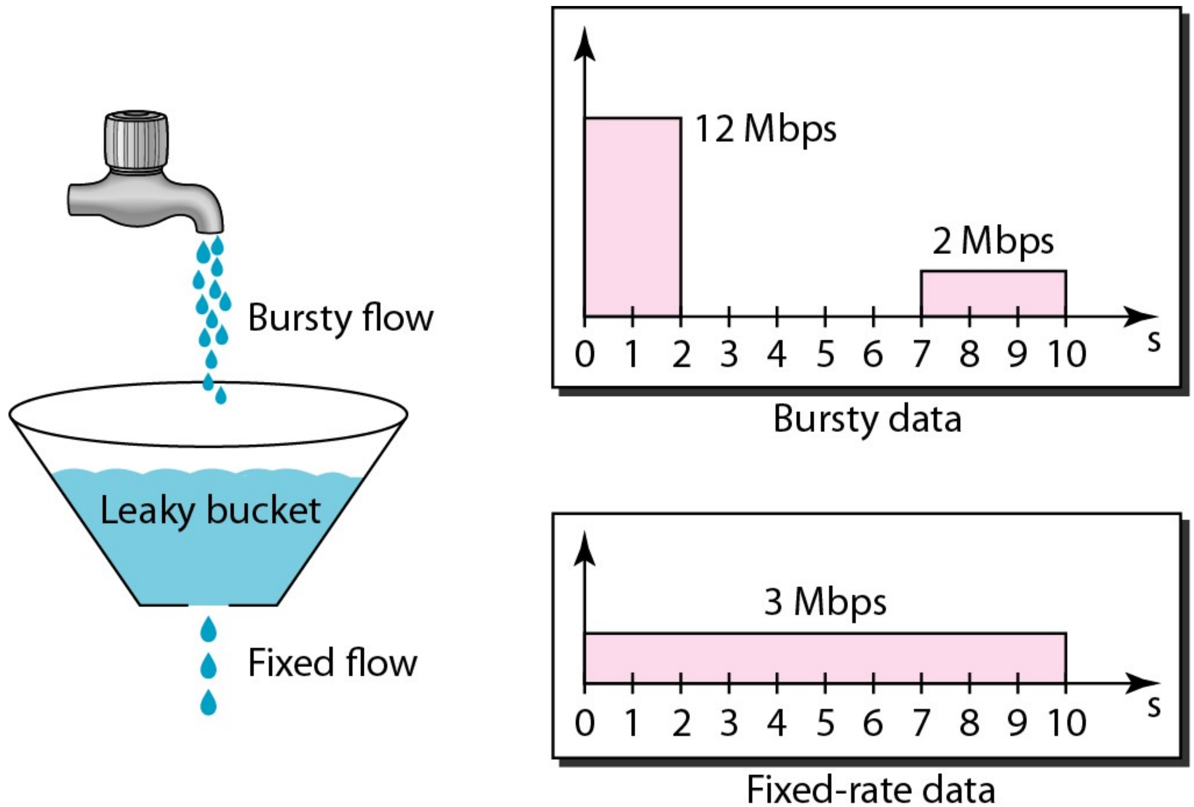


The Leaky Bucket Algorithm

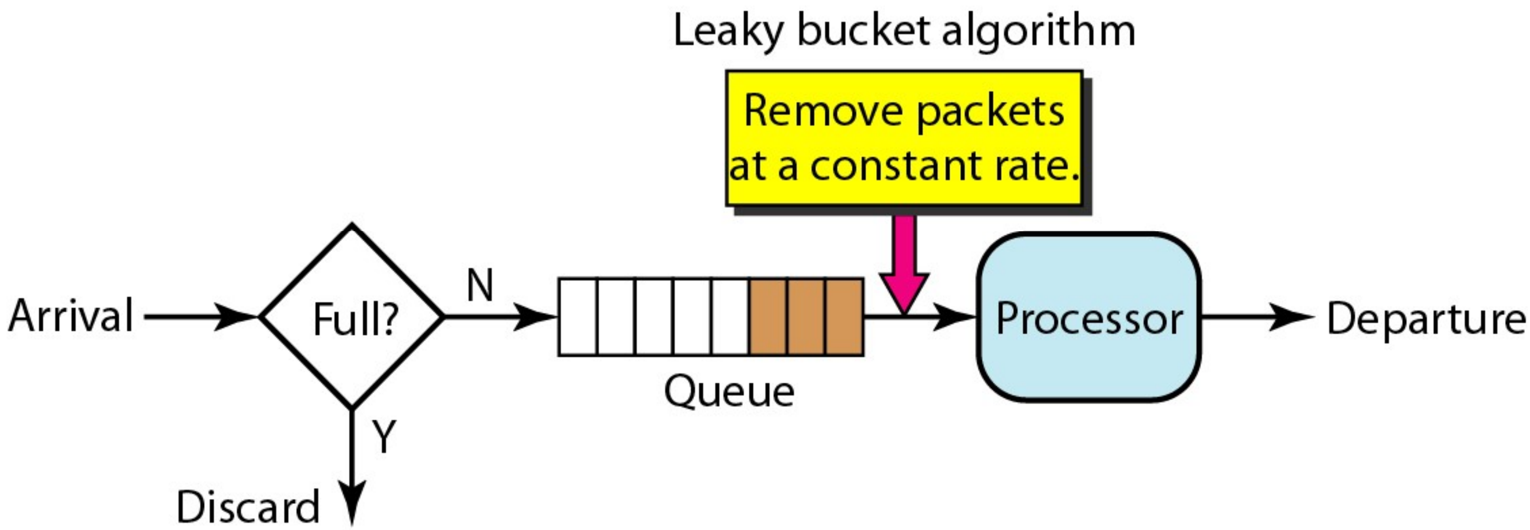


(a) A leaky bucket with water. (b) a leaky bucket with packets.

The Leaky Bucket Algorithm



Leaky bucket implementation



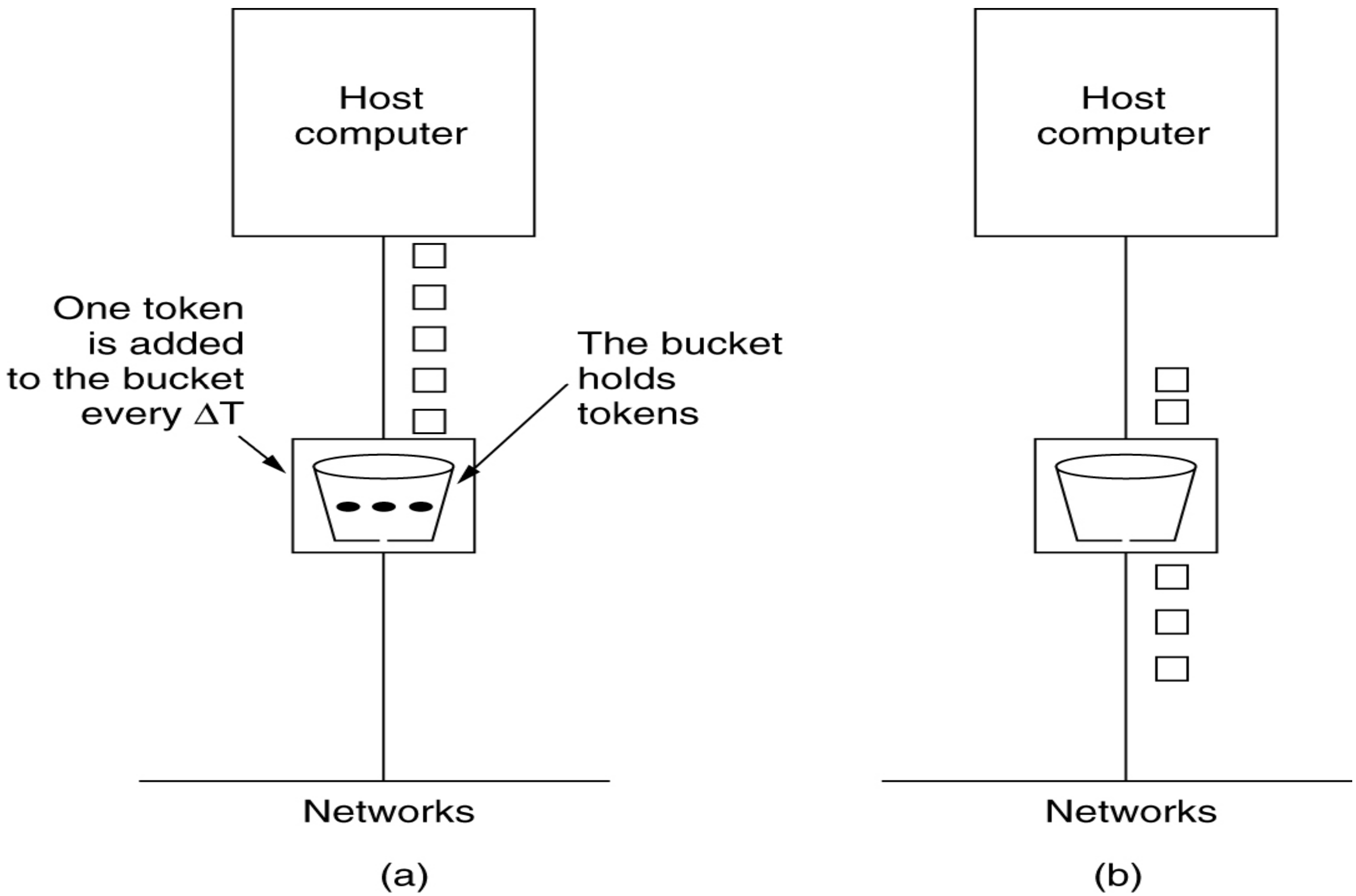
Note

A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

Note

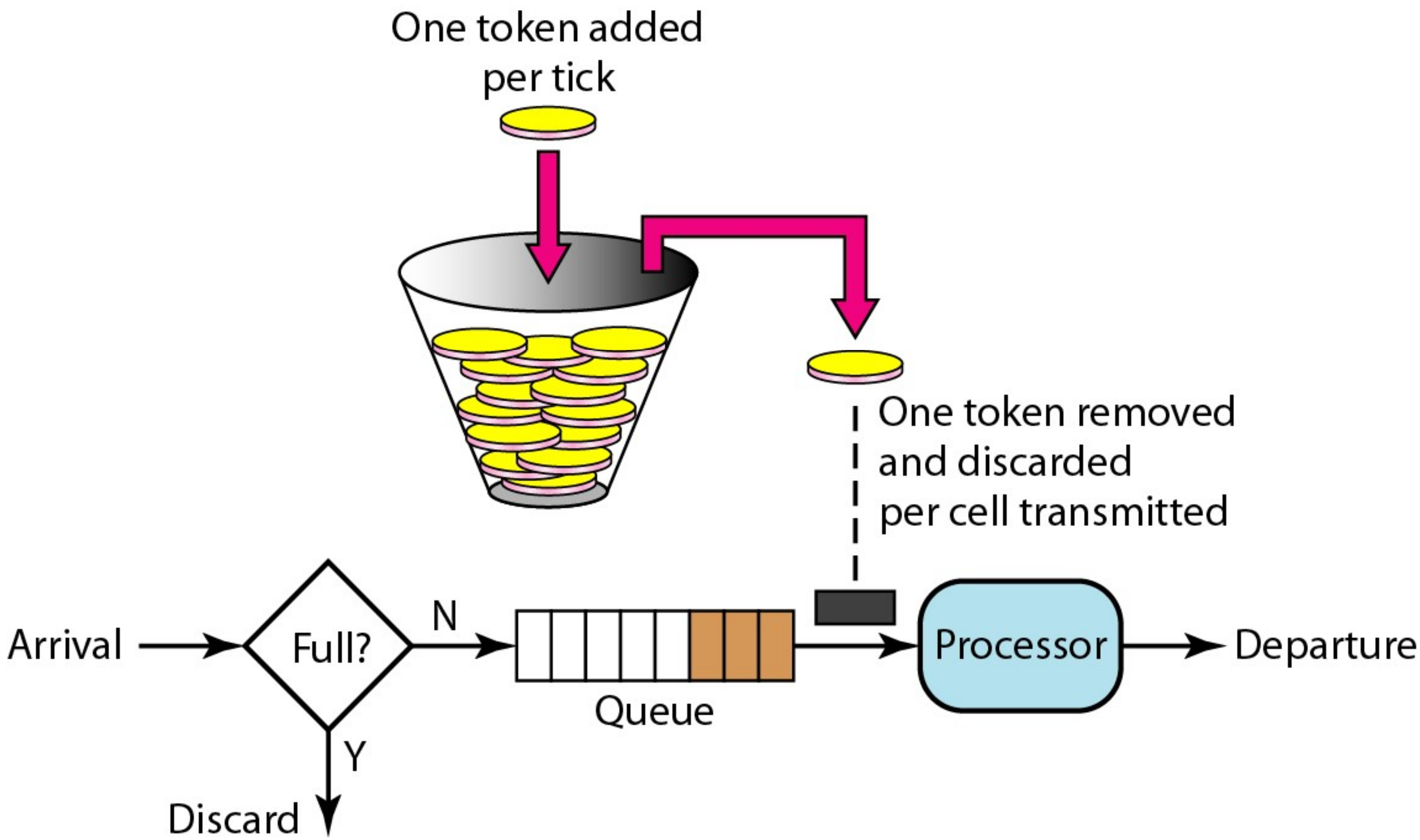
The token bucket allows bursty traffic at a regulated maximum rate.

The Token Bucket Algorithm



(a) Before. (b) After.

Token bucket algorithm implementation



Admission Control

An example of flow specification.

Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- **TCP Congestion Control,**
- **UDP – (Gap Analysis)** ***content beyond syllabus***
- **Real Time Transport protocol(RTP),**
- **Stream Control Transmission Protocol (SCTP),**
- **Quality of Service (QoS),**
- **Differentiated services,**
- TCP and UDP for Wireless.

Two models to provide Quality of Service

- Integrated Services
- Differentiated Services

Integrated Services

- Flow based QoS model
- Which means used need to create a flow, a kind of virtual circuit from source to destination and inform all routers about the resource requirement
- This kind of reservation of resources is done by a protocol called RSVP(Resource Reservation Protocol)

Integrated Services

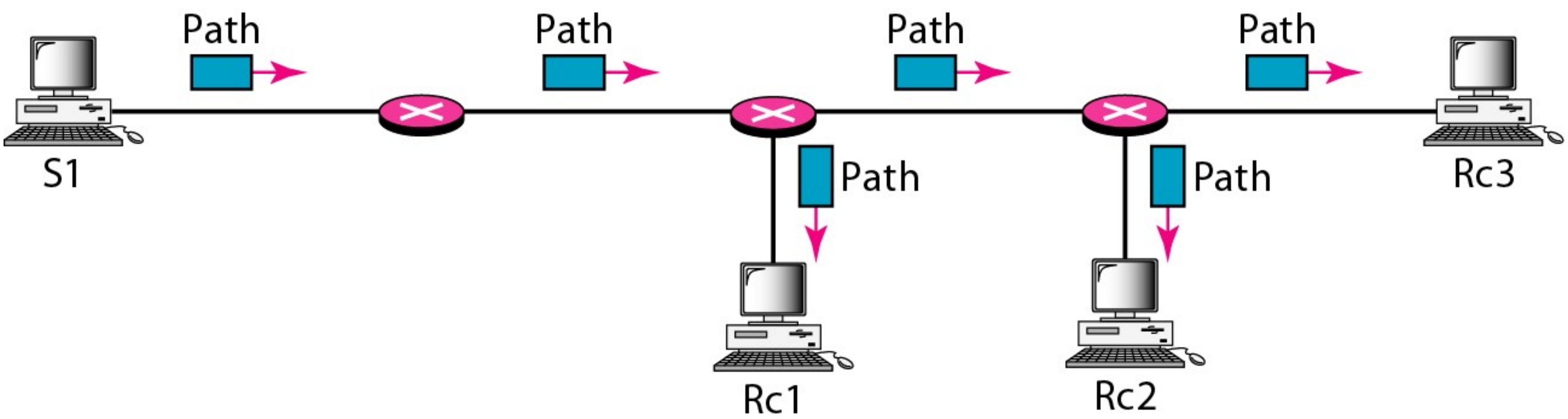
- Resource reservation means reserve how much buffer, bandwidth etc is needed.
- When a router receives flow specification from an application, it decides to admit or deny the service
- Two classes of service is defined for Integrated serviced

1. Guaranteed Service Class(For real time application)
2. Controlled-load Service(For application require reliablility)

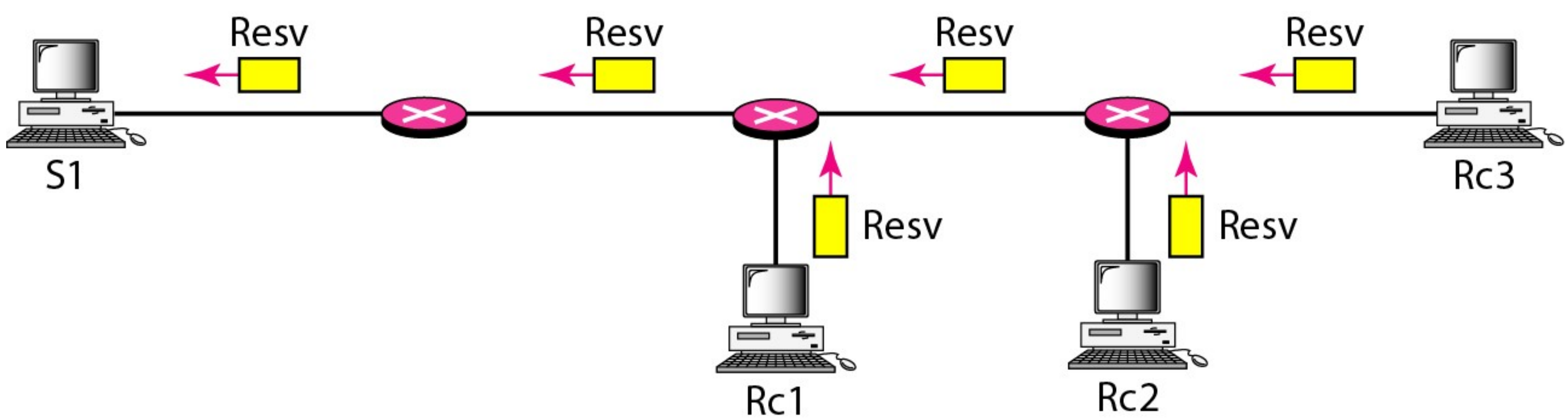
RSVP-The Resource ReSerVation Protocol

- The **Resource Reservation Protocol** (**RSVP**) is a Transport layer protocol designed to reserve resources across a network for an Integrated service network.
- RSVP operates over an IPV4 or IPV6 and provides resource reservations for multicast or unicast data flows
- RSVP can be used by either host or routers to request or deliver specific levels of quality of service (QoS) for application data streams or flows.
- RSVP defines how applications place reservations and how they can give up the reserved resources once the need for them has ended.

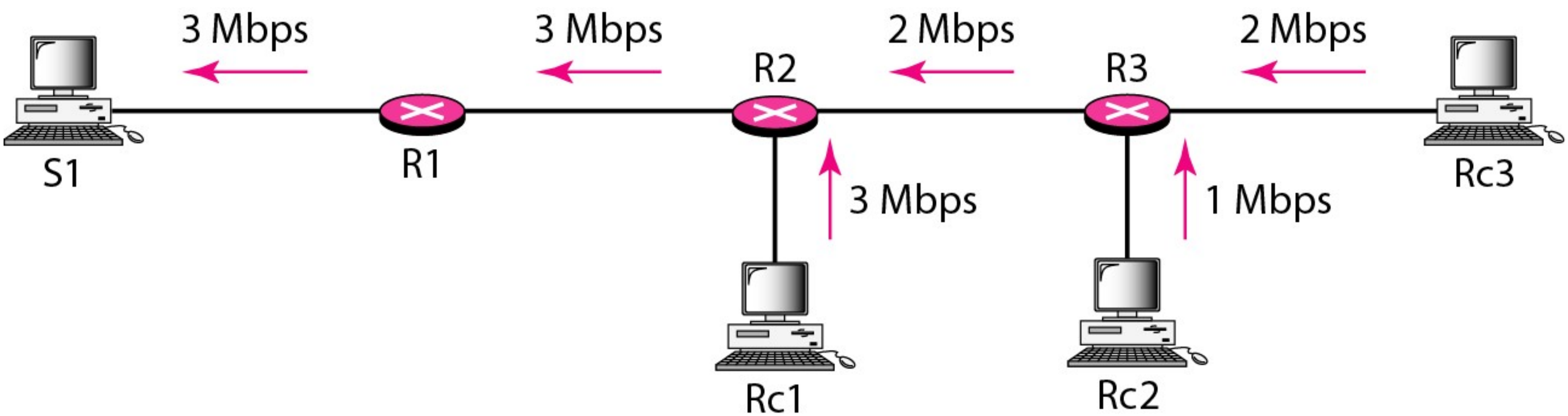
Path messages



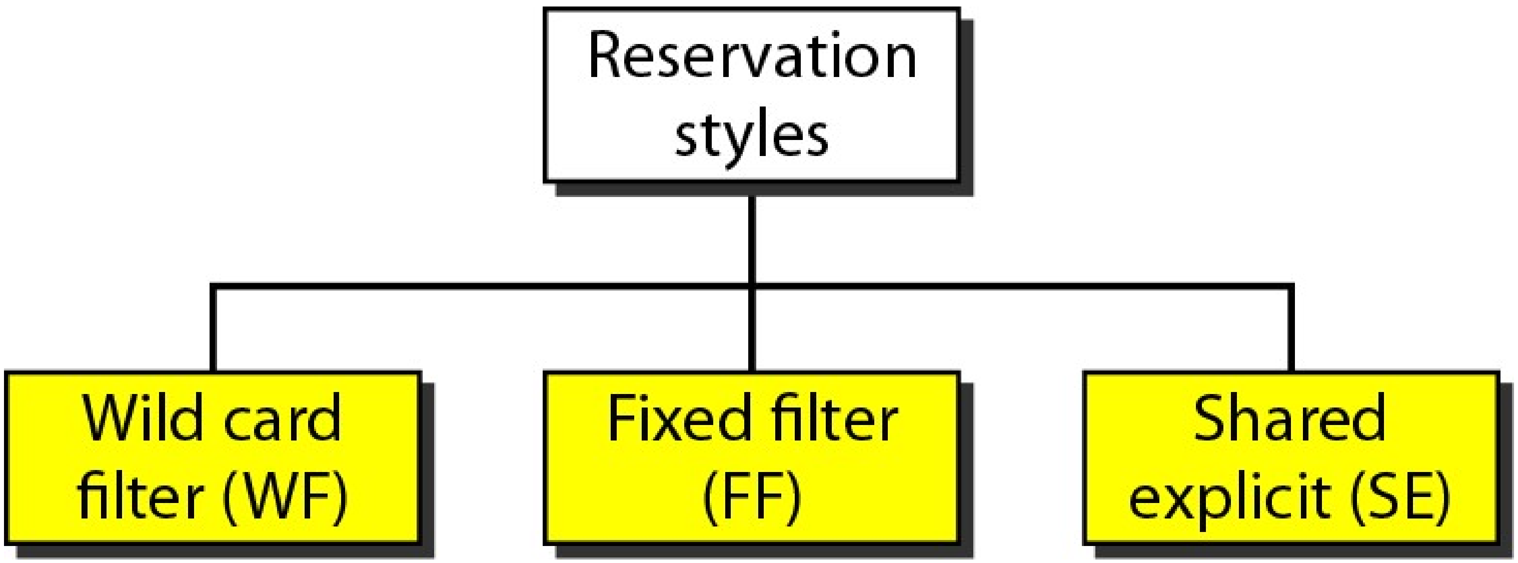
Resv messages



Reservation merging



Reservation styles



Problems with Integrated Services

Scalability: Each router keep information for each flow. So does not possible to scale more

Service Type Limitation: Only two types of services are provided guaranteed and control based

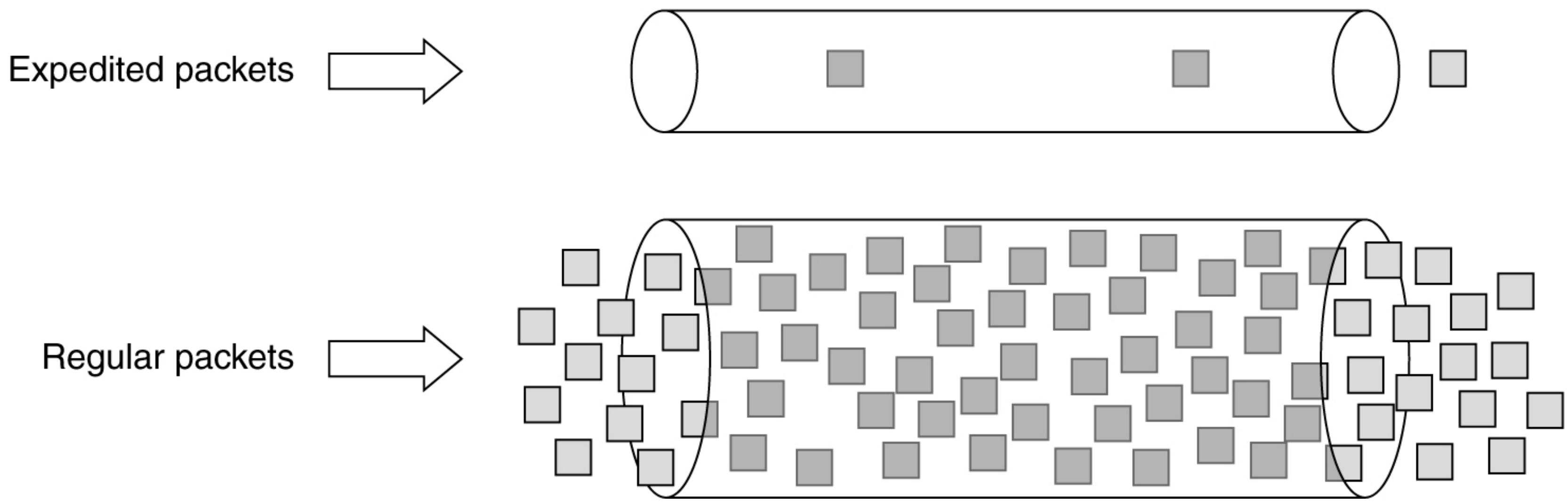
Differentiated Services

- Handles shortcomings of Integrated Services .
- In differentiated model router does not store information about flows.
- No advance reservation is required
- It is a Class based service model
- Each packet contains a field called DS field
- It has two types of models

1. Expedited forwarding
2. Assured Forwarding

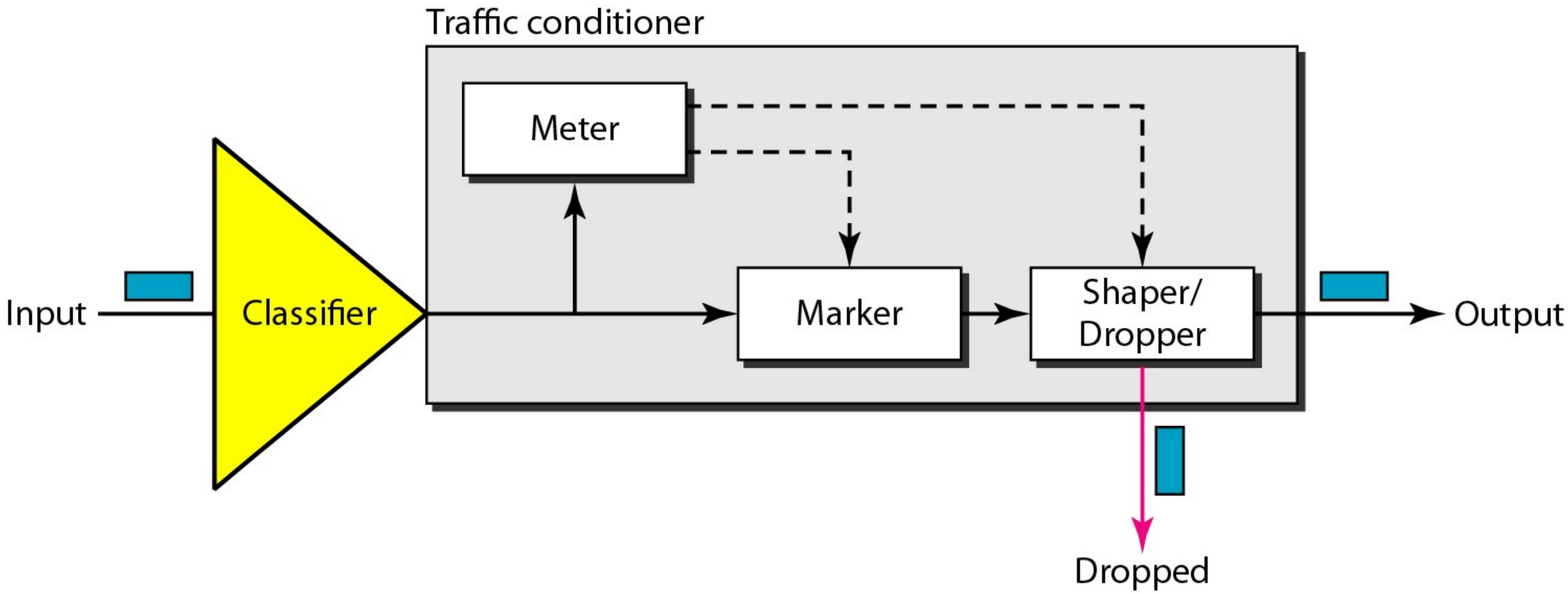
DS field

Expedited Forwarding



- In this model two classes of service is available:
  - 1>Regular
  - 2> Expedited
- Expedited packets experience a traffic-free network.

Assured Forwarding



- There are 4 priority classes , each having 3 discard policies like low,medium and high.
- Traffic controller have Classifier,Marker and Shaper/Dropper
- Packet is classified according to priority, then marked according to their class .
- Shaper/dropper filter these packet that may drop or delay the packet.

Outline

- **Services,**
- **Addressing**
- **Berkley Sockets,**
- **Multiplexing,**
- **TCP**
- **Connection establishment,**
- **Connection release,**
- **Flow control and buffering,**
- **TCP Timer management,**
- **TCP Congestion Control,**
- **UDP – (Gap Analysis)** ***content beyond syllabus***
- **Real Time Transport protocol(RTP),**
- **Stream Control Transmission Protocol (SCTP),**
- **Quality of Service (QoS),**
- **Differentiated services,**
- **TCP and UDP for Wireless.**

TCP over Wireless : outline

- TCP over Wireless: Problems
- TCP over Wireless: Solutions/Schemes
  - Split TCP
    1. Indirect TCP
    2. Selective repeat protocol
    3. Mobile TCP
  - TCP-aware link layer
    1. Snoop
    2. WTCP
  - Link layer protocol
  - End-to-end protocol
    1. Selective Acknowledgement
    2. Explicit Loss Notification

TCP over Wireless: Problems

- TCP has been optimized for wired networks.
- Any packet loss is considered to be the result of network congestion and the congestion window size is reduced drastically as a precaution.
- Sources of errors in wireless links:

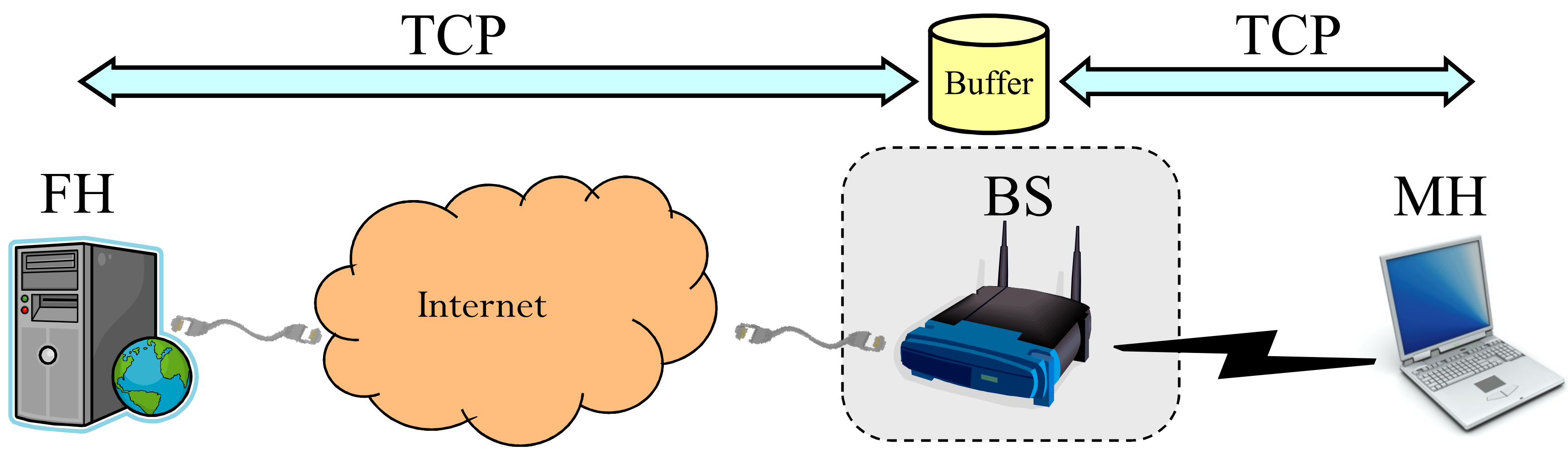
1. Due to hands off between cells
2. Packet losses due to futile transmissions
3. Packet losses due to transmission errors in wireless links

TCP over Wireless : outline

- TCP over Wireless: Problems
- TCP over Wireless: Solutions (Schemes)
  - Split TCP
    1. Indirect TCP
    2. Selective repeat protocol
    3. Mobile TCP
  - TCP-aware link layer
    1. Snoop
    2. WTCP
  - Link layer protocol
  - End-to-end protocol
    1. Selective Acknowledgement
    2. Explicit Loss Notification

Split TCP: Indirect TCP

- I-TCP splits end-to-end TCP connection into two connections
  - Fixed host to BS
  - BS to mobile host
- Two TCP connections with independent flow/congestion control contexts
- Packets buffered at BS

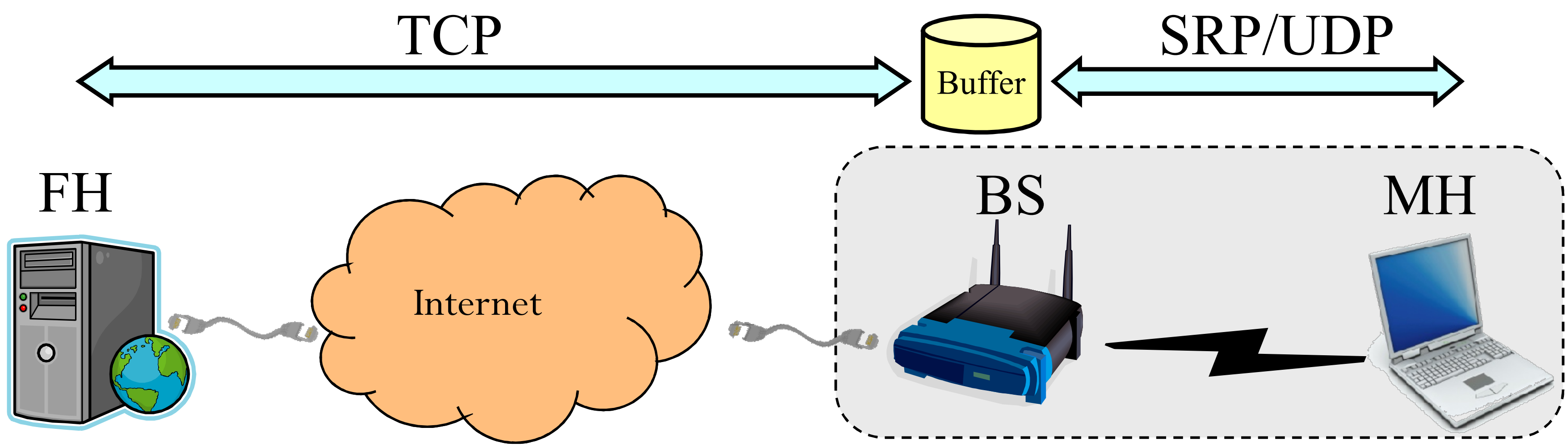


Split TCP: Indirect TCP

- Pros
  - Separates flow and congestion control of wireless and wired
    --higher throughput at sender
- Cons
  - Breaks TCP end-to-end semantics
    - Ack at FH does not mean MH has received the packet
  - BS failure causes loss of data
    - Neither FH nor MH can recover the data
  - On path change, data has to be forwarded to new BS
  - Wireless part is the bottleneck

Split TCP: Selective Repeat Protocol

- Similar to I-TCP but uses SRP/UDP (Selective Repeat Protocol over UDP) over wireless link, Improving End-to-End Performance of TCP over Mobile Internetworks
- Pros
  - Better performance over wireless links
- Cons
  - All cons of I-TCP except last one



Split-TCP: Mobile TCP

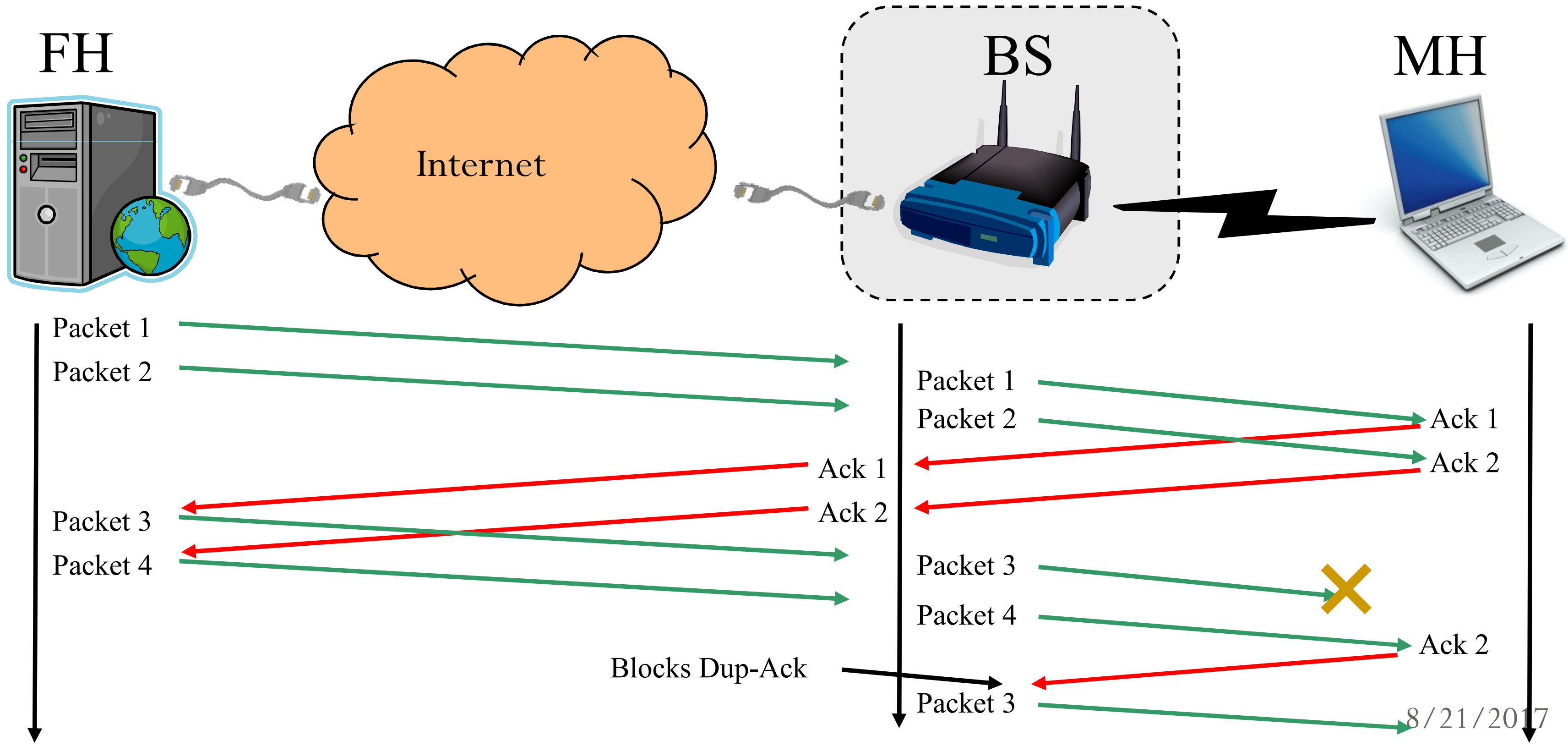
- Similar to I-TCP but tries to keep TCP end-to-end semantics
- No buffering , no retransmission at base station BS.
- BS only monitors all packets and only acks the last packet after it is received by MH
- Pros
  - Data will be recovered eventually after BS failure
  - BS buffer does not overflow
- Cons
  - Worse performance
  - Still not exactly the TCP semantics

TCP over Wireless : outline

- TCP over Wireless: Problems
- TCP over Wireless: Solutions (Schemes)
  - Split TCP
    1. Indirect TCP
    2. Selective repeat protocol
    3. Mobile TCP
  - TCP-aware link layer
    1. Snoop
    2. WTCP
  - Link layer protocol
  - End-to-end protocol
    1. Selective Acknowledgement
    2. Explicit Loss Notification

TCP-aware Link Layers: Snoop

- Link layer is aware of TCP traffic
- BS caches data and monitors acks. Retransmits on duplicate acks and drops duplicate acks



TCP-aware Link Layers: Snoop

- Pros
  - No modification to FH and MH
  - BS only keeps soft state—BS failure does not break TCP
- Cons
  - Does not work with encrypted packets
  - Does not work if data packets and acks traverse different paths
  - Increases RTT—high timeout

TCP-aware Link Layers: WTCP

- Similar to Snoop
- WTCP corrects RTT by modifying the timestamp in return acks