Transport Layer

Textbook: Computer Networks (Tanenbaum)

Transport Layer

Provides end-to-end transport of messages between remote applications

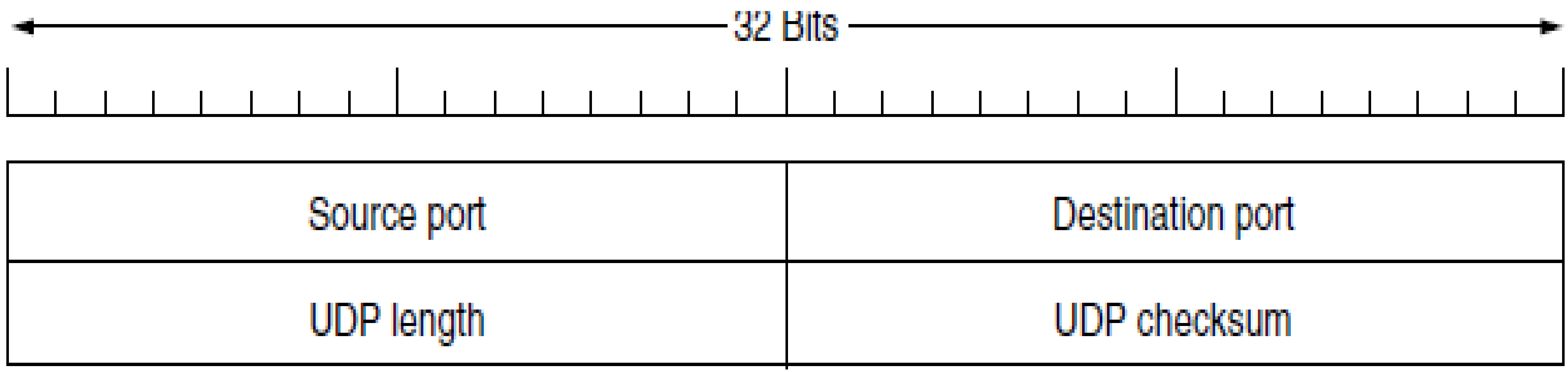
Good abstraction for application developers to enable cooperation between heterogeneous networks

Works at the end users as opposed to network services which operate at the routers

Connection-based (e.g. TCP) vs. connectionless communication (e.g. UDP)

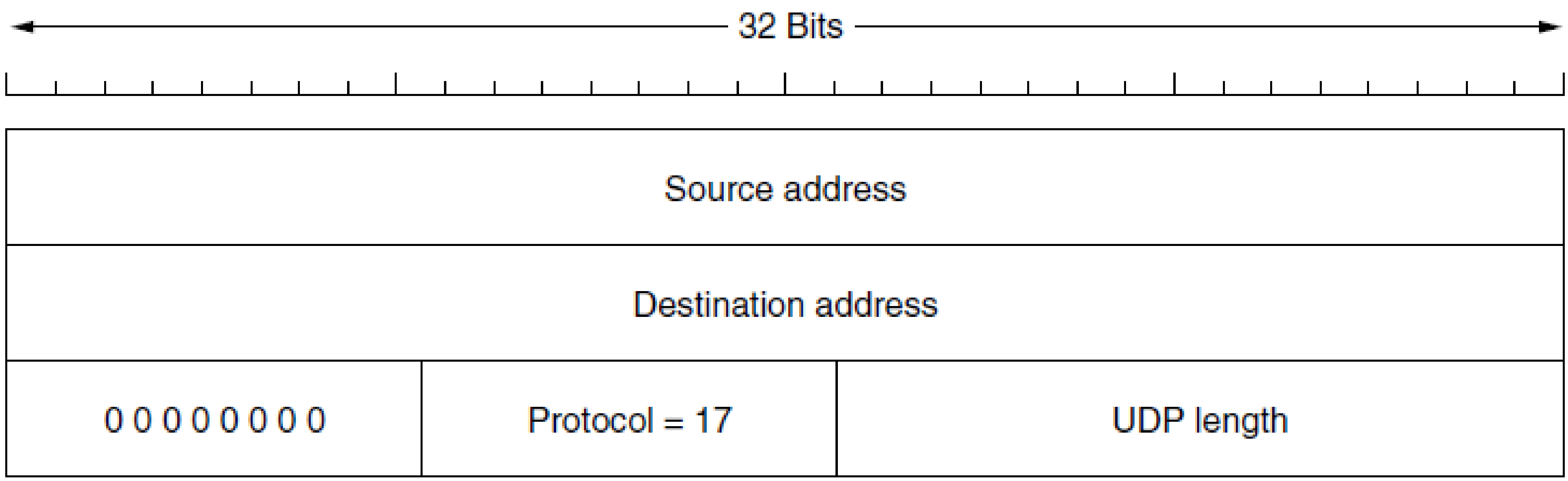
UDP (User Datagram Protocol)

UDP



- UDP transmits segments, header 8 bytes
- 16-bits for port number (0.. 65,535).
- Port numbers to identify to which application the message should go.
- source port needed when a reply is expected.
- segment length includes the header (max length is 65,515 to accommodate for IPv4 pseudo header (see next slide))

IPv4 pseudo header



- IPv4 source and destination addresses
- Protocol = 17 for UDP

-IPv4 pseudo header is 12 bytes and UDP header is 8 bytes thus UDP max length is 65535 – 20 = 65515.

UDP does and does not

Does
- end-to-end error detection

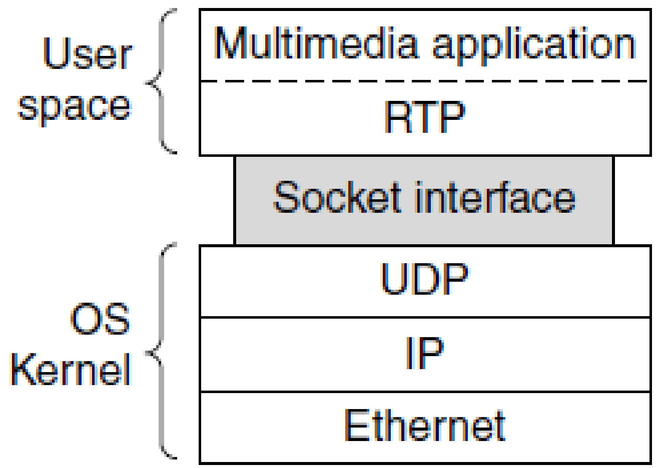
Does not
- retransmission of corrupt packets
- congestion control
- flow control

Good for applications that want to handle “Does not” themselves, gives them more control. (e.g. client/server with exchange of short messages where it is better to use UDP to avoid TCP's initial setup complexity)
→ e.g. DNS

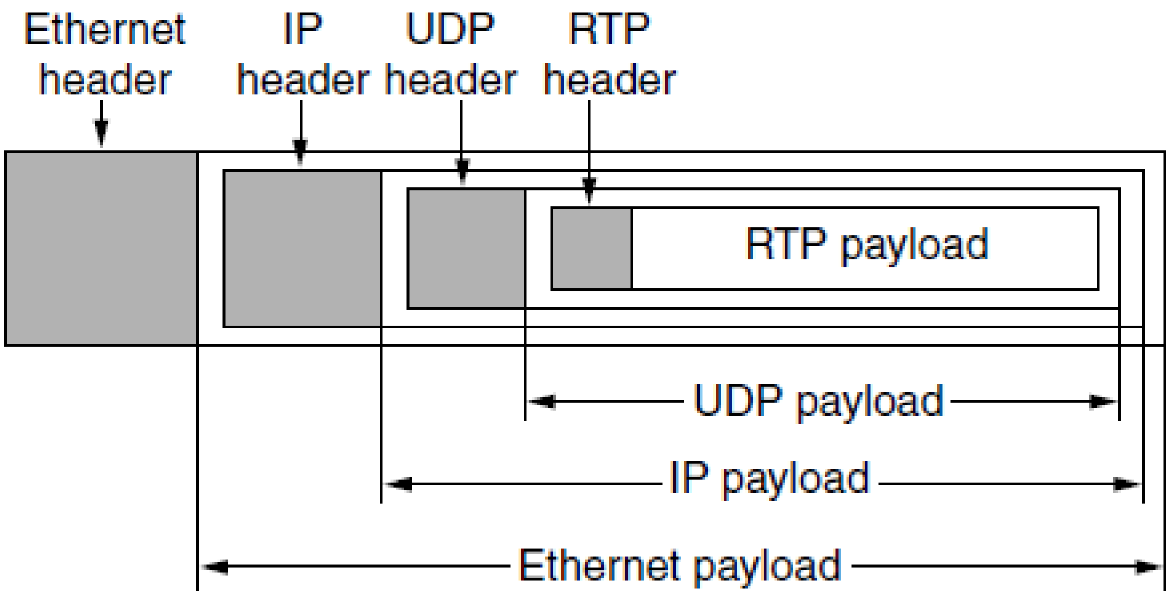
Well known UDP port numbers

Port	Protocol
7	Echo
9	Discard
11	Users
13	Daytime
17	Quote
19	Chargen
53	Nameserver
67	Bootps
68	Bootpc
69	TFTP
111	RPC
123	NTP

Realtime Transport Protocol (RTP)



RTP in the protocol stack



Packet nesting

- The success of Internet Radio, TV, Telephony, Video-conferencing, … lead to the need for having a generic rather than many specific real-time transport protocols.
- RTP is a library in the user space. Composed of two parts: one for transportation and the second for playing media.
- RTP is a transport protocol that is implemented at the application layer

Realtime Transport Protocol (RTP)

- No guarantee about delivery. Packets may be lost, or delayed.

- If a packet is missing, the application runs interpolation to compensate for the loss. Retransmission not helpful as may arrive late.

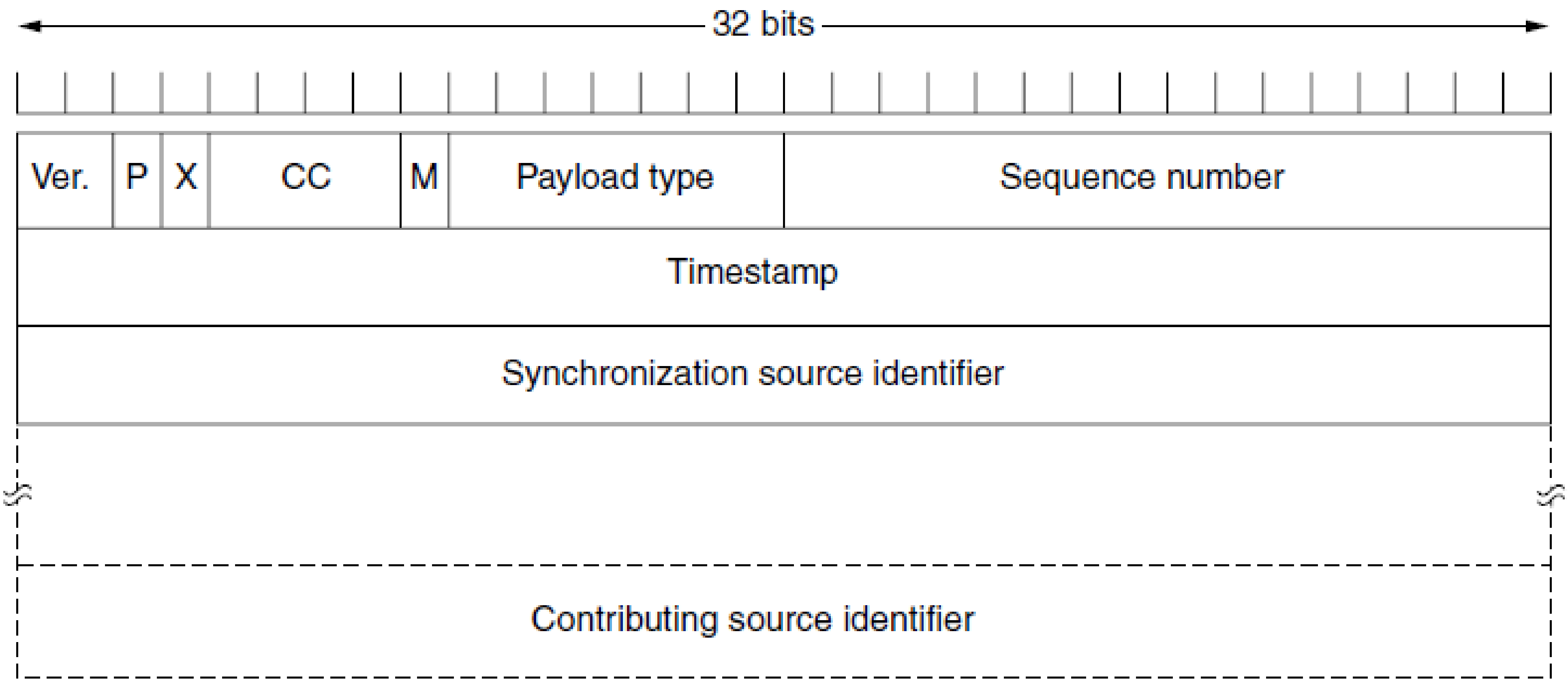
- Defines profiles (e.g. single audio stream)

- Specifies encoding scheme (e.g. H.264, H.265) for each profile.

- Uses time-stamping (here difference between timestamps) to allow applications to buffer packets to avoid delay variation effects.

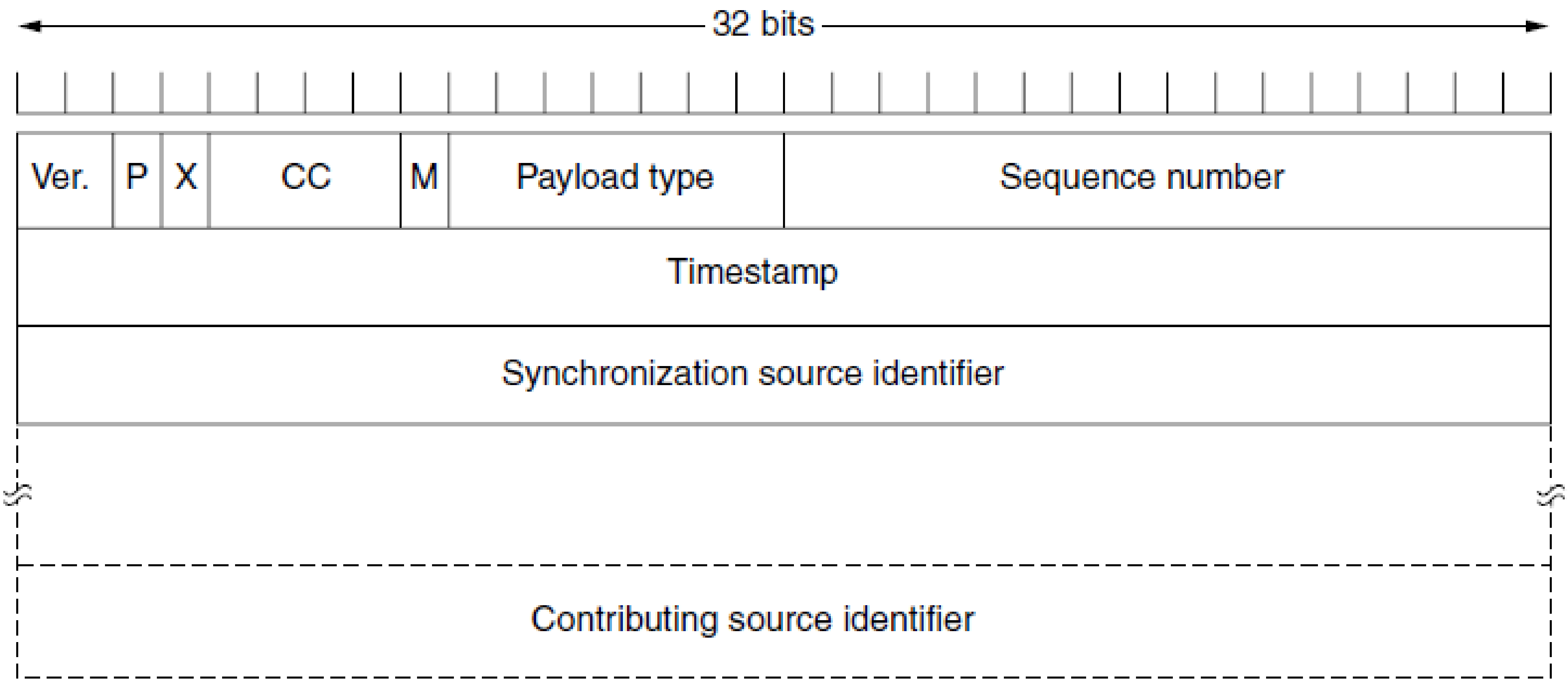
- Time-stamping also helps synchronising different media (e.g. a TV program may have one video stream and two audio streams)

Realtime Transport Protocol (RTP)



- Ver. RTP Version
- P means that packet has been padded
- X means extension byte added.
- CC tells how many contributing sources are used.
- M application-specific: can mark the start of video, audio,

Realtime Transport Protocol (RTP)



- Payload type: which encoding algorithm has been used (e.g H.264)
- Sequence number: to figure out if a packet has gone missing.
- Synchronization source id: the stream the packet belongs to.

Real Time Control Protocol (RTCP)

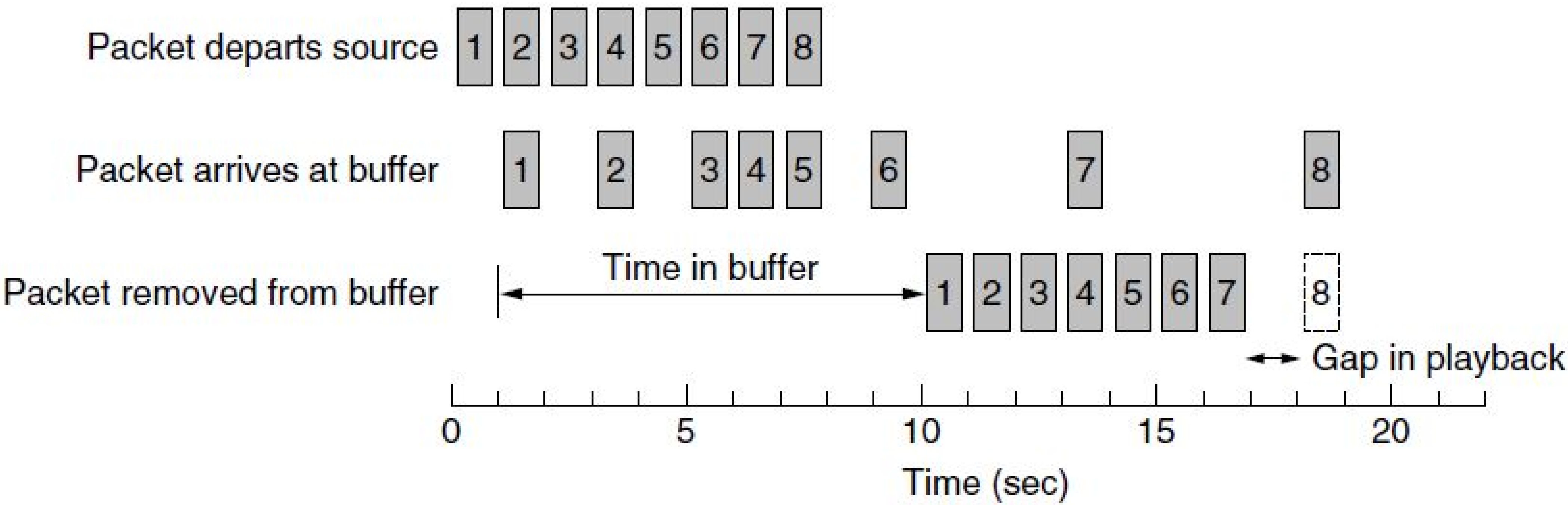
The RTP Control Protocol (RTCP) is a sister protocol of the Real-time Transport Protocol (RTP).

The primary function of RTCP is to provide feedback on the quality of service (QoS) in media distribution by periodically sending statistics information to participants in a streaming multimedia session.

RTCP provides out-of-band statistics and control information for an RTP session. Statistics include: delay, jitter (variation in delay), bandwidth, congestion, …

→ increase data rate when network is functioning well, or decrease the data rate or use a different Codec when it is not

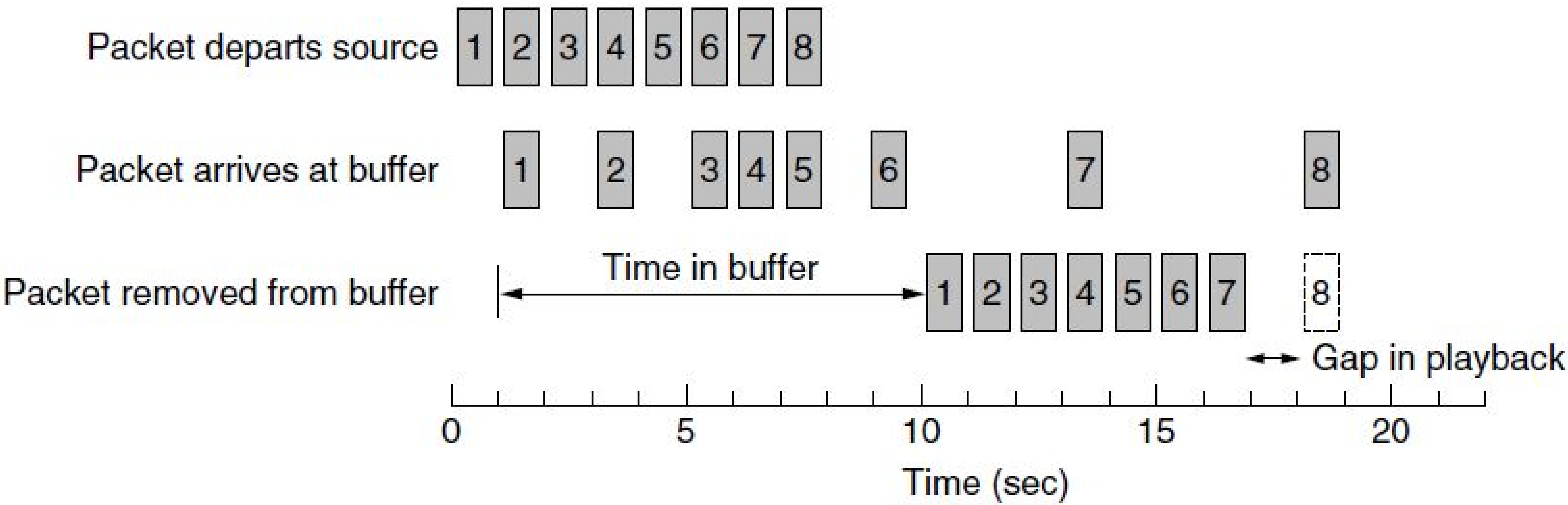
Buffering and Jitter Control



- Even if media is injected regularly, irregularity may happen during transit causing jitters. If played out as they arrive, they cause jerky video frames and unintelligible audio
  → buffer packets

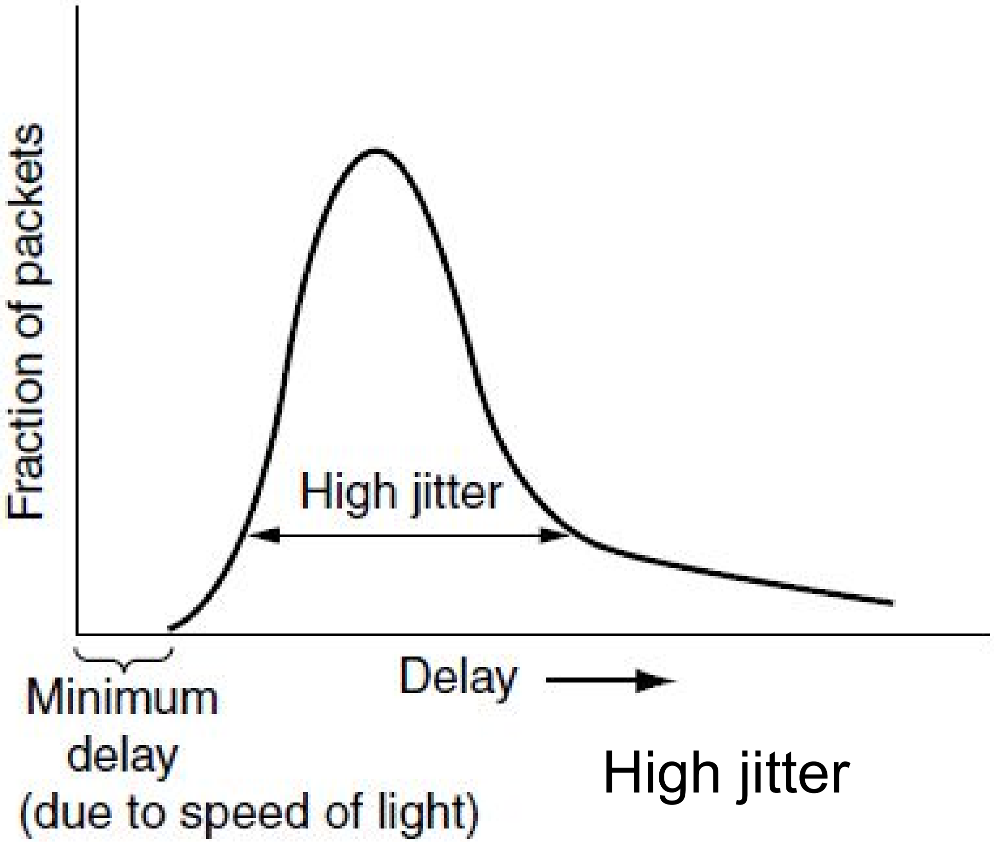
- Packet 8 still arrives late. Either discard (voice-over-IP) or pause the video and wait (video streaming)

Buffering and jitter control

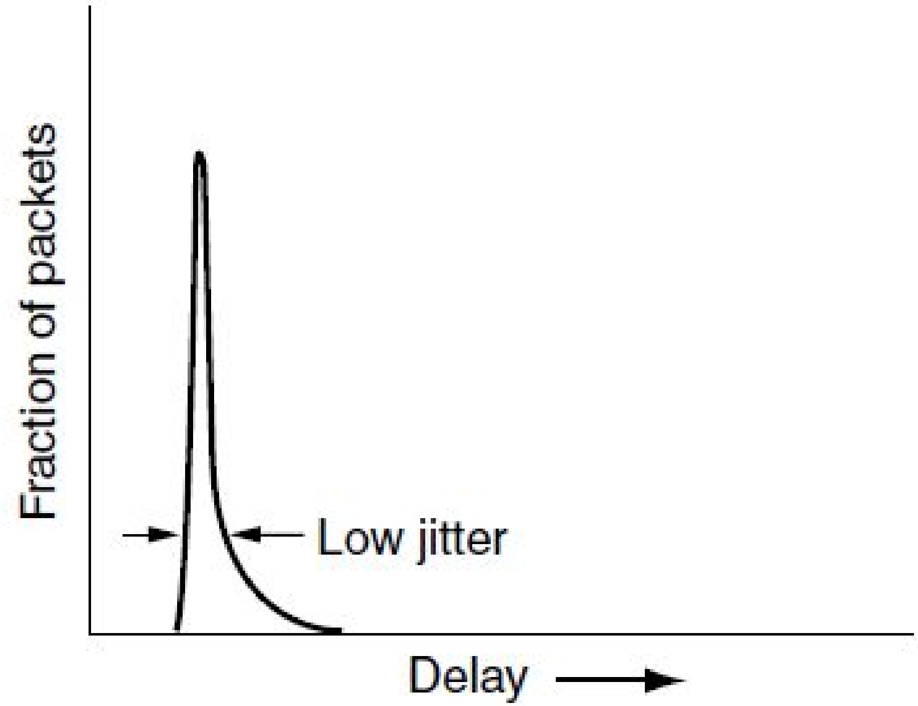


- For video streaming using a 10s buffer can be reasonable to absorb network delays of packet that are not dropped.
- For interactive applications (e.g. video-conferencing), a smaller buffer is needed for responsiveness.
- Playback point: how long to buffer?

Playback point: how long to buffer?



High jitter



Low jitter

- The longer we wait the more packets we get. Applications can adapt their playback point. The marker bits indicates the beginning of a talkspurt.
- Beginning of talkspurts are good opportunities to adjust the playout delay at the receiver

TCP (Transmission Control Protocol)

Transmission Control Protocol

- TCP accepts data streams from local processes. TCP breaks a stream into pieces of a max of 64KB (in practice 1460B to fit in a single Ethernet frame)

- IP does not know the capacity of the network. TCP should tune the speed of packet transmission to increase the bandwidth but not to cause congestion.

- TCP retransmits packets that time out without ACK.

- TCP reorders packets to reconstruct the stream.

- TCP connection between sender and receiver is identified by the port numbers.

Well known TCP ports

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

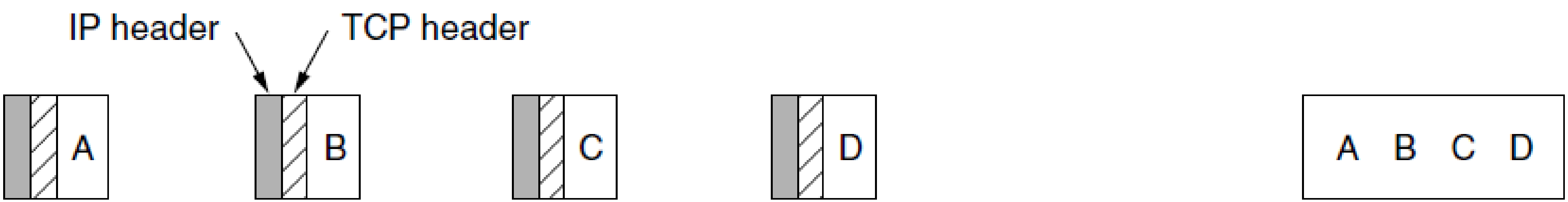
- Port numbers below 1024. Usually can be only started by root.
E.g. to retrieve email using IMAP, connect to server on port 143.

TCP is point to point

- TCP works on full duplex.

- TCP does not support multicast or broadcast. Only point to point is possible.

TCP is byte stream



Four 512-byte segments sent in separate IP packets

The 2048 data is delivered in a single call

- TCP may buffer data to collect larger amount and send it at once

- Use TCP_NODELAY to expedite transmission of urgent packets

TCP segments

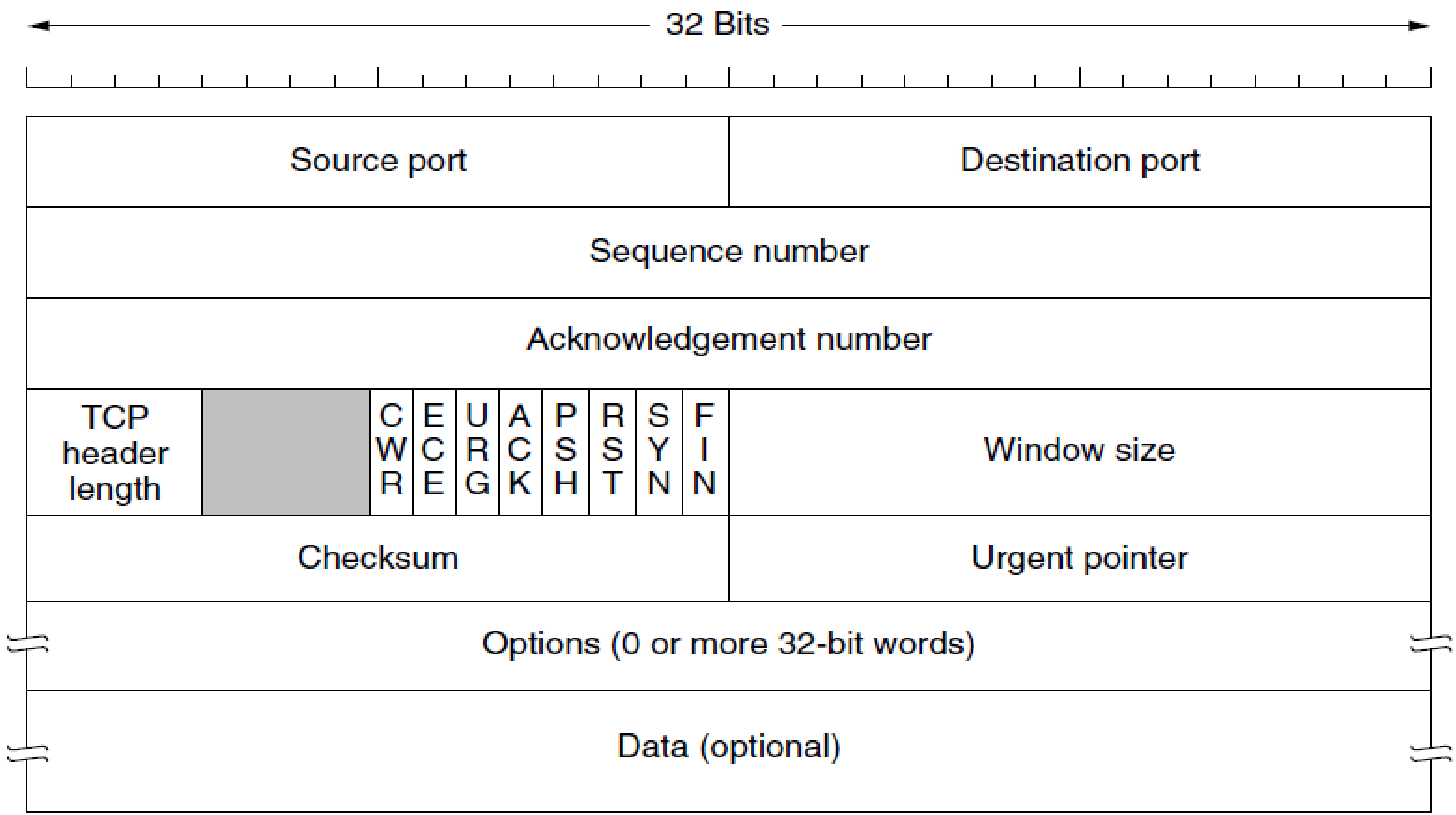
- TCP exchanges data in the form a segments

- Each segment has a 20 byte header

- Each segment must not exceed IP payload (65515 bytes) … but practically limited by the link layer. Each link has a MTU (Maximum Transfer Unit) … in Ethernet MTU is 1500 bytes.

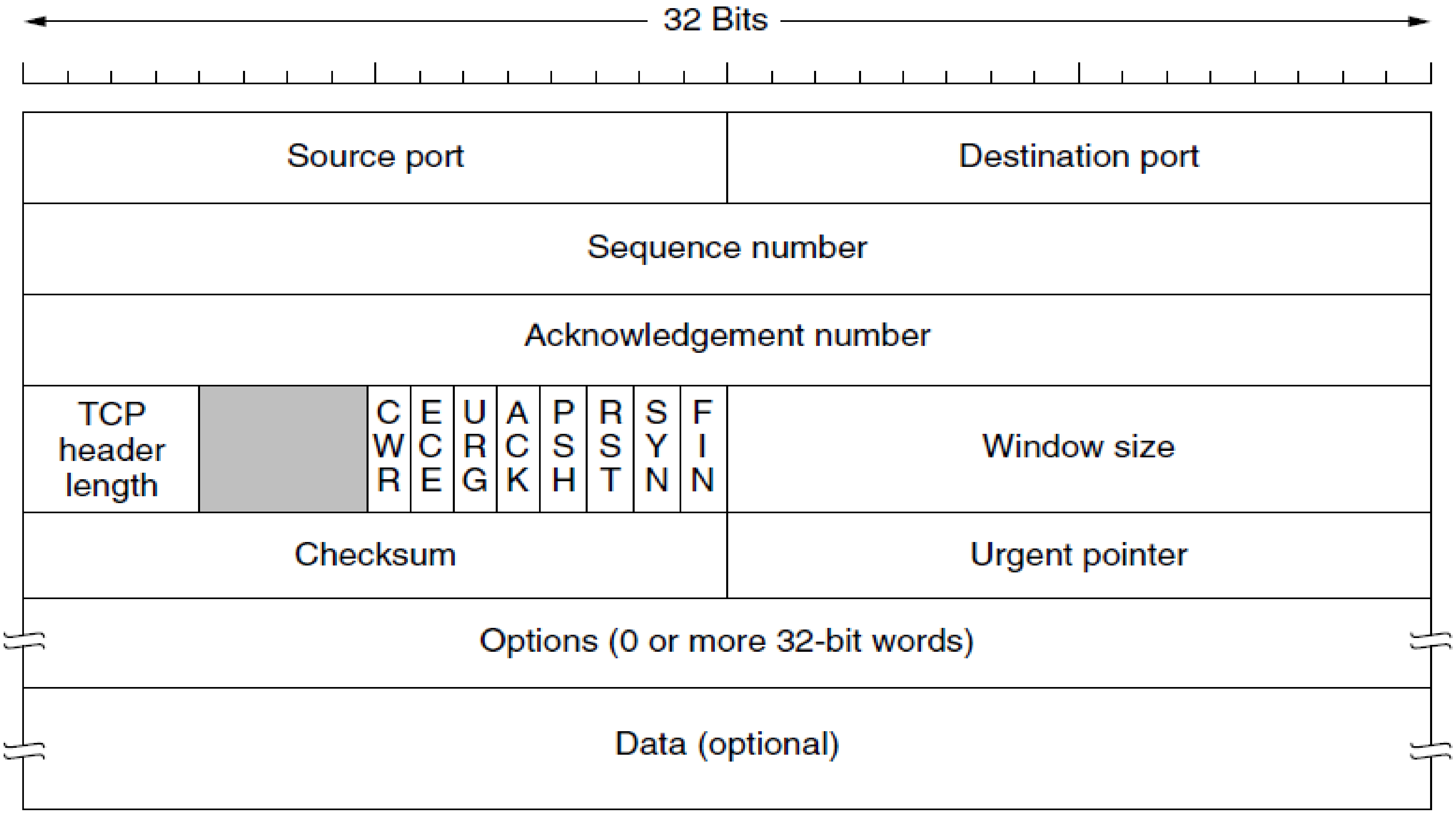
- Still TCP segment can be fragmented if they pass through links with smaller MTU which leads to performance degradation
  → modern TCP uses path MTU discovery

TCP segment header



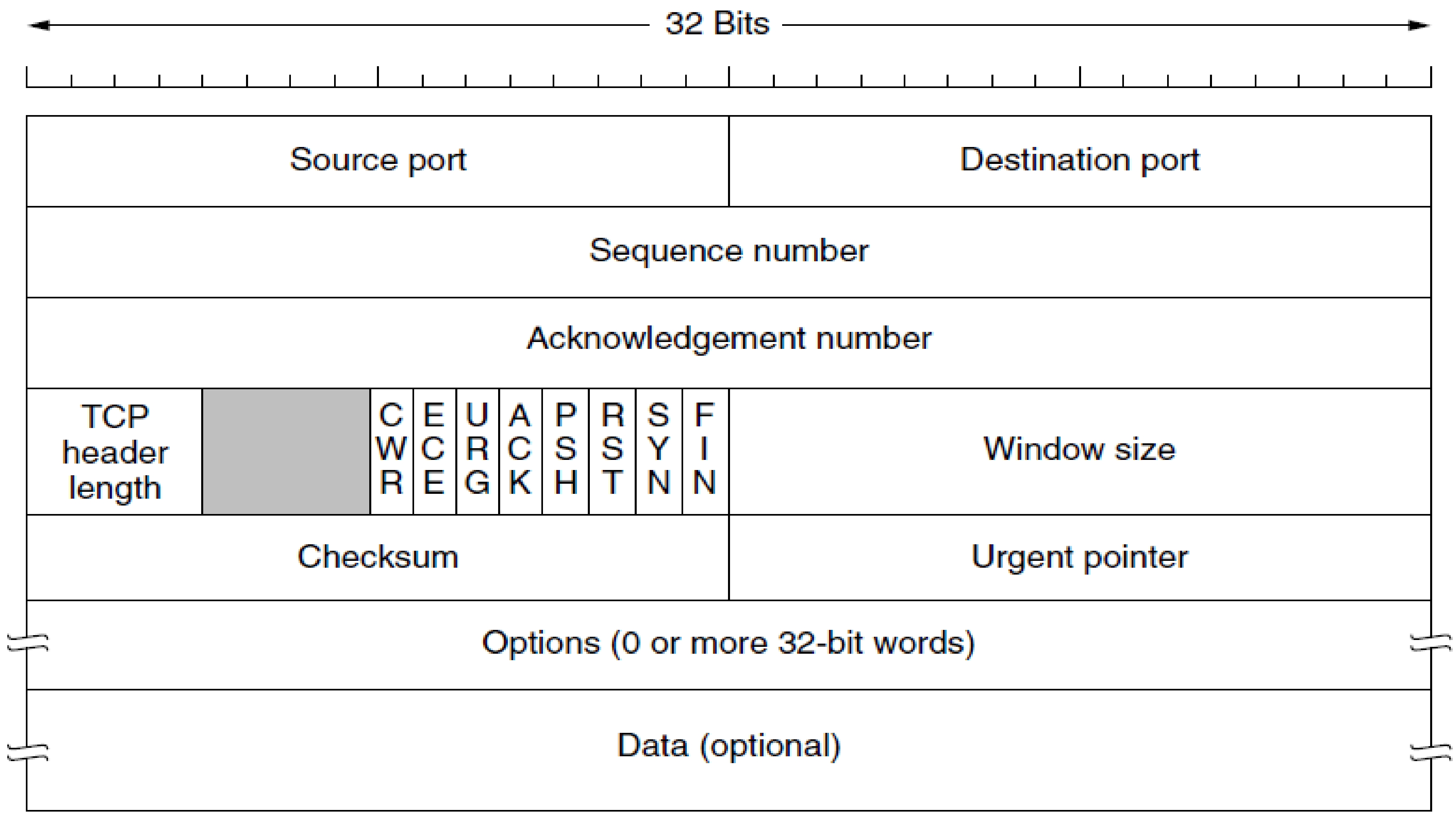
Acknowledgement number specifies the next in order byte expected, not the last byte correctly received.
TCP header length is needed because, 'Options' has variable length

TCP segment header



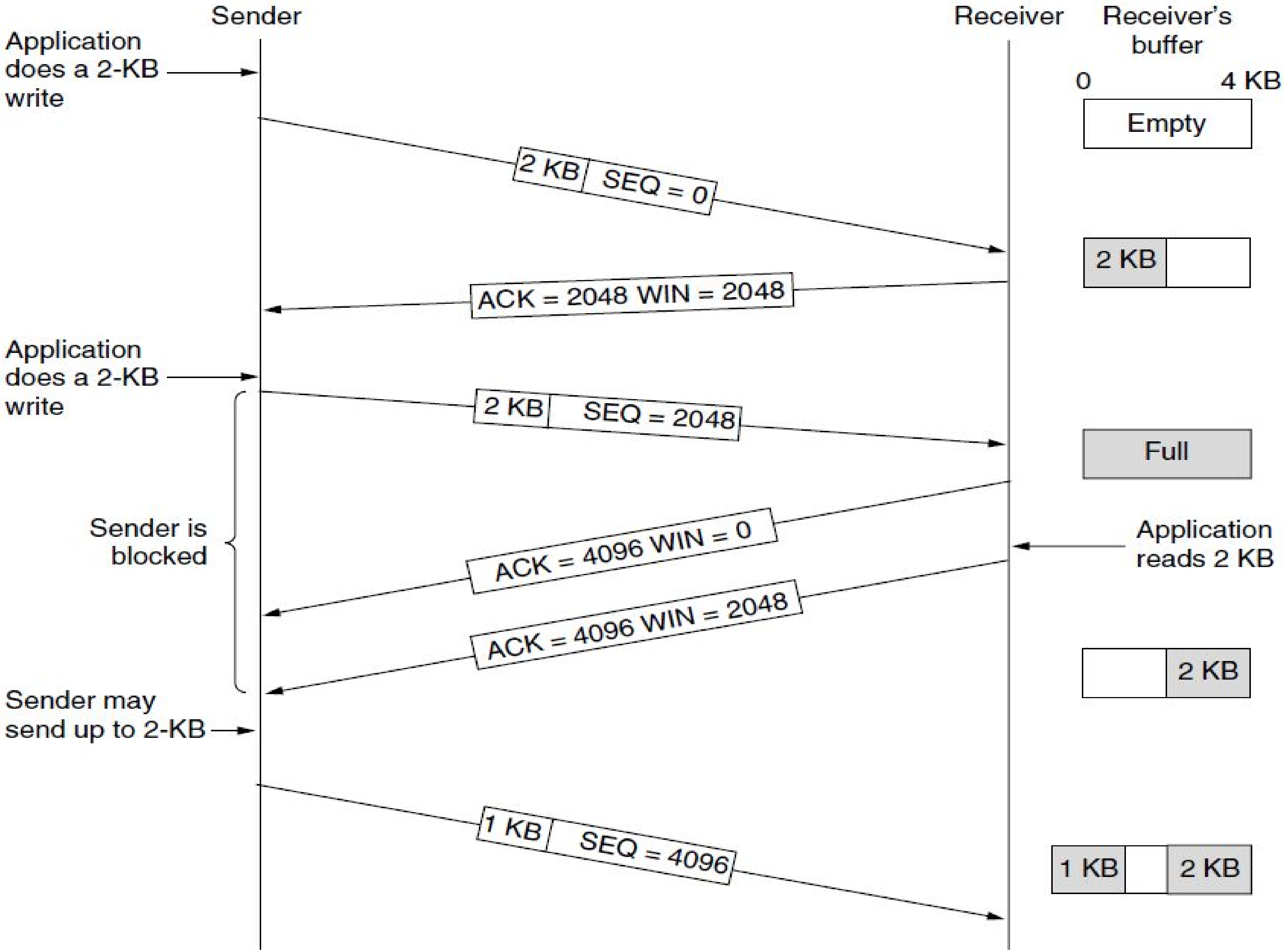
RST: used to reset the connection (e.g. when a problem arises)
SYN: used to establish a connection
FIN: used to release the connection

TCP segment header

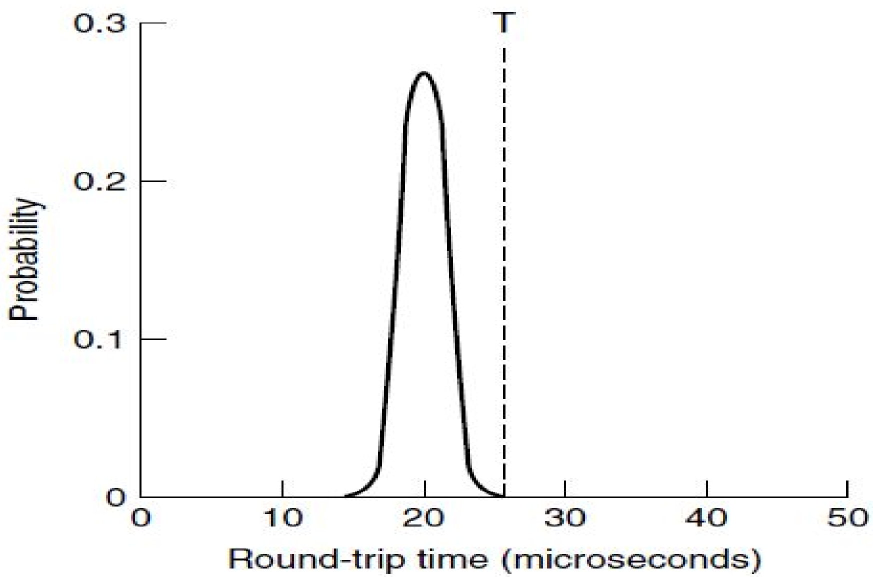


Window size: how many bytes may be sent after the acked byte
Checksum: same as in UDP

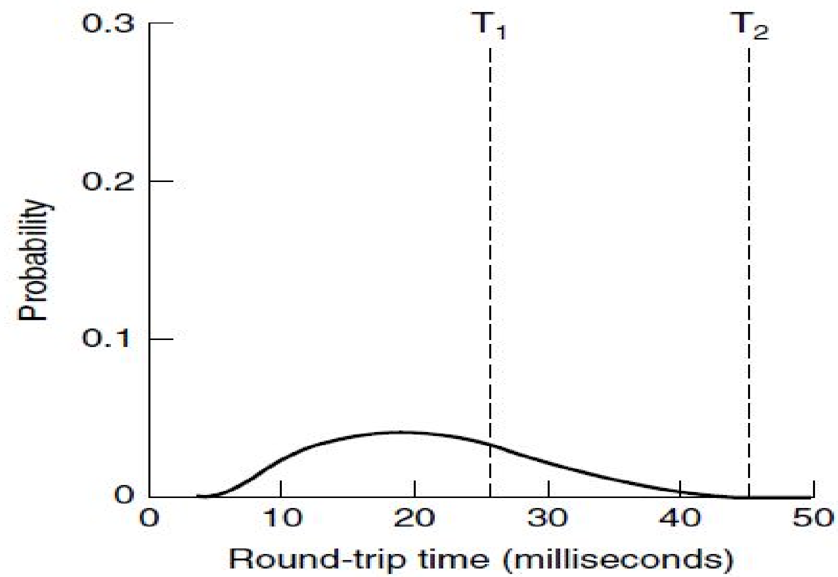
TCP window management



TCP timer management



PDF of ACK arrival time at link layer



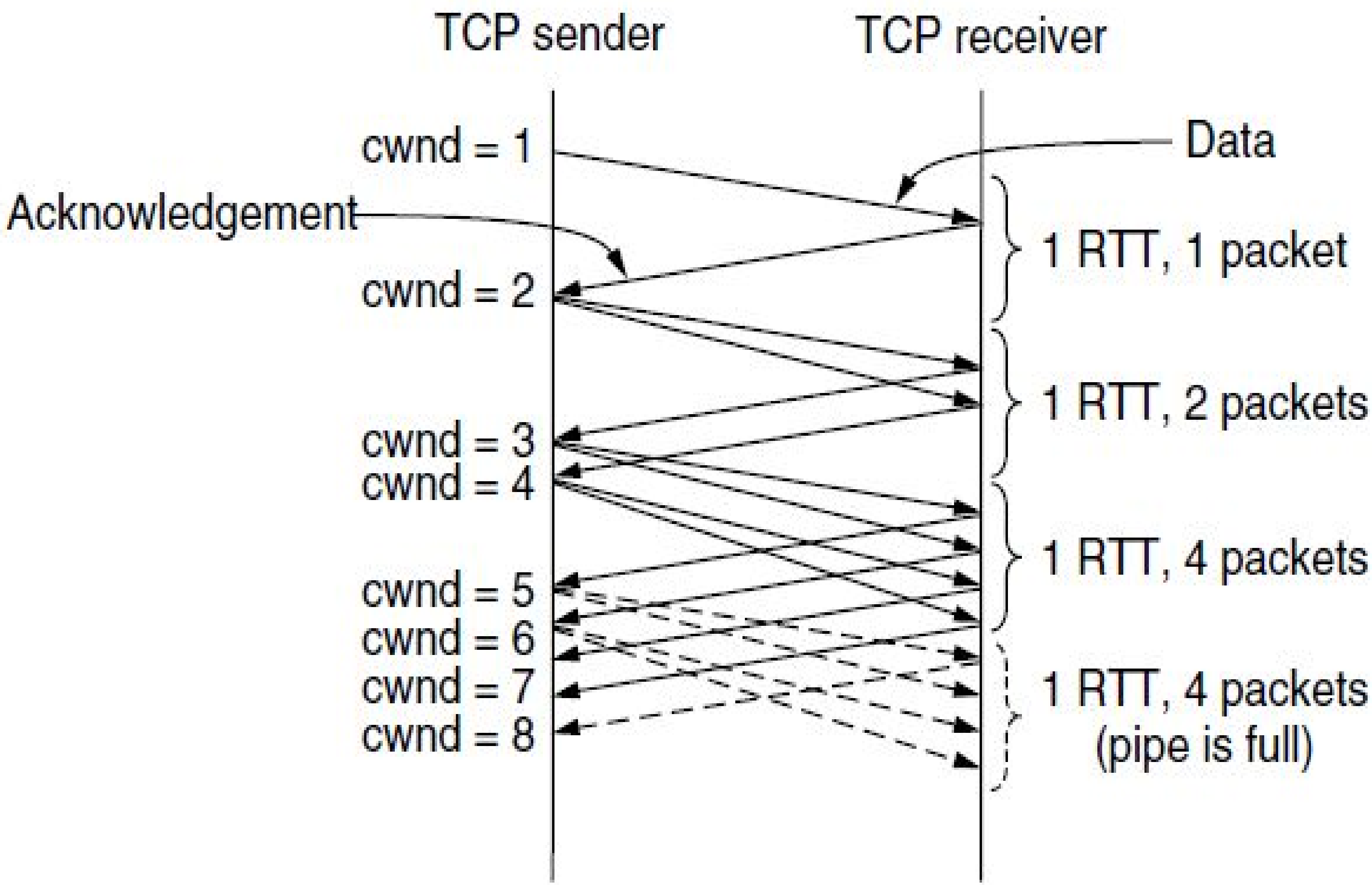
PDF of ACK arrival time at transport layer

If time is too short (e.g. T1) → unnecessary retransmissions
If time is too long (e.g. T2) → time wasted before a retx demand

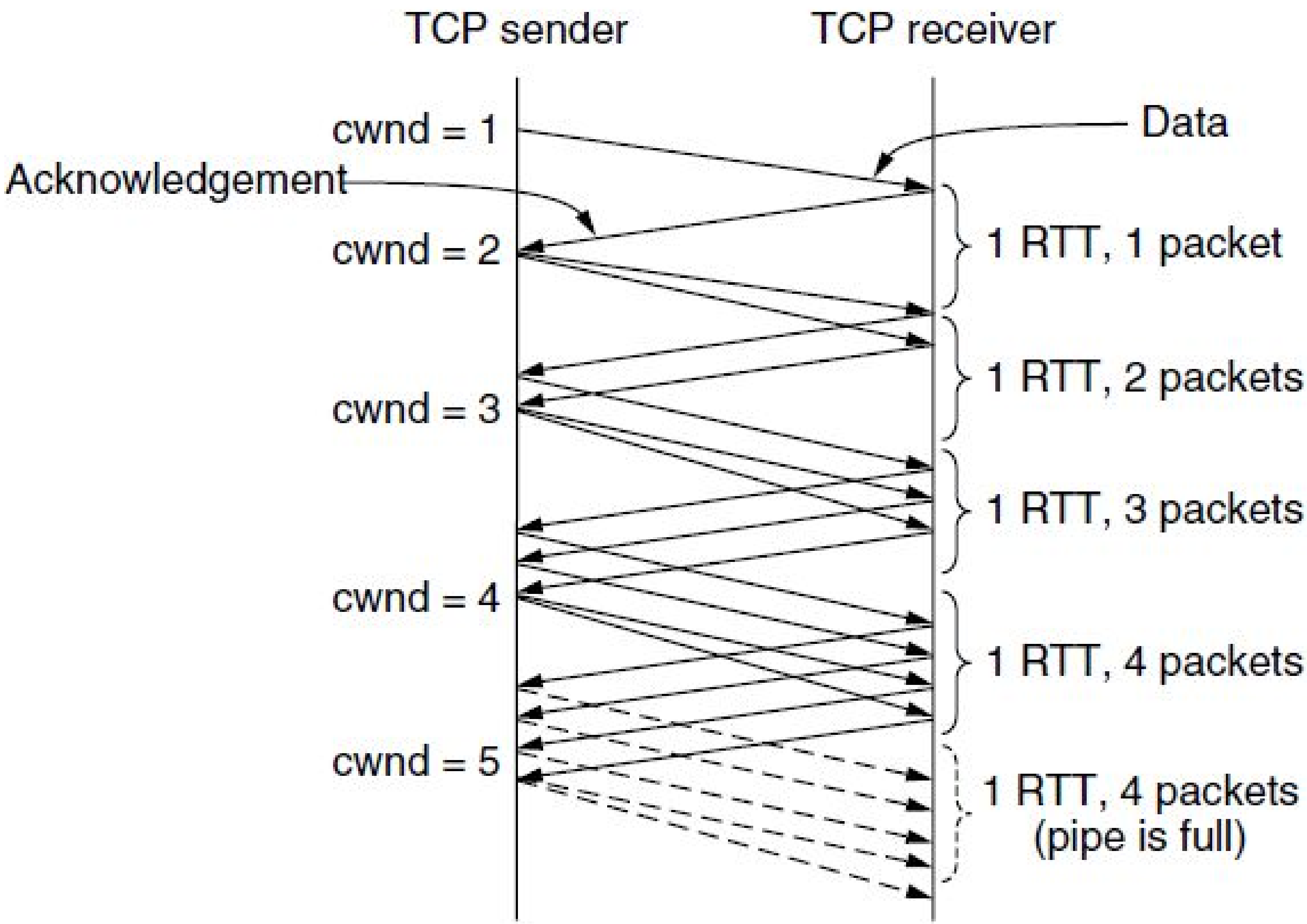
Use EWMA (Exponentially Waiting Moving Average)

$$SRTT = \alpha\ SRTT + (1 - \alpha)\ R$$

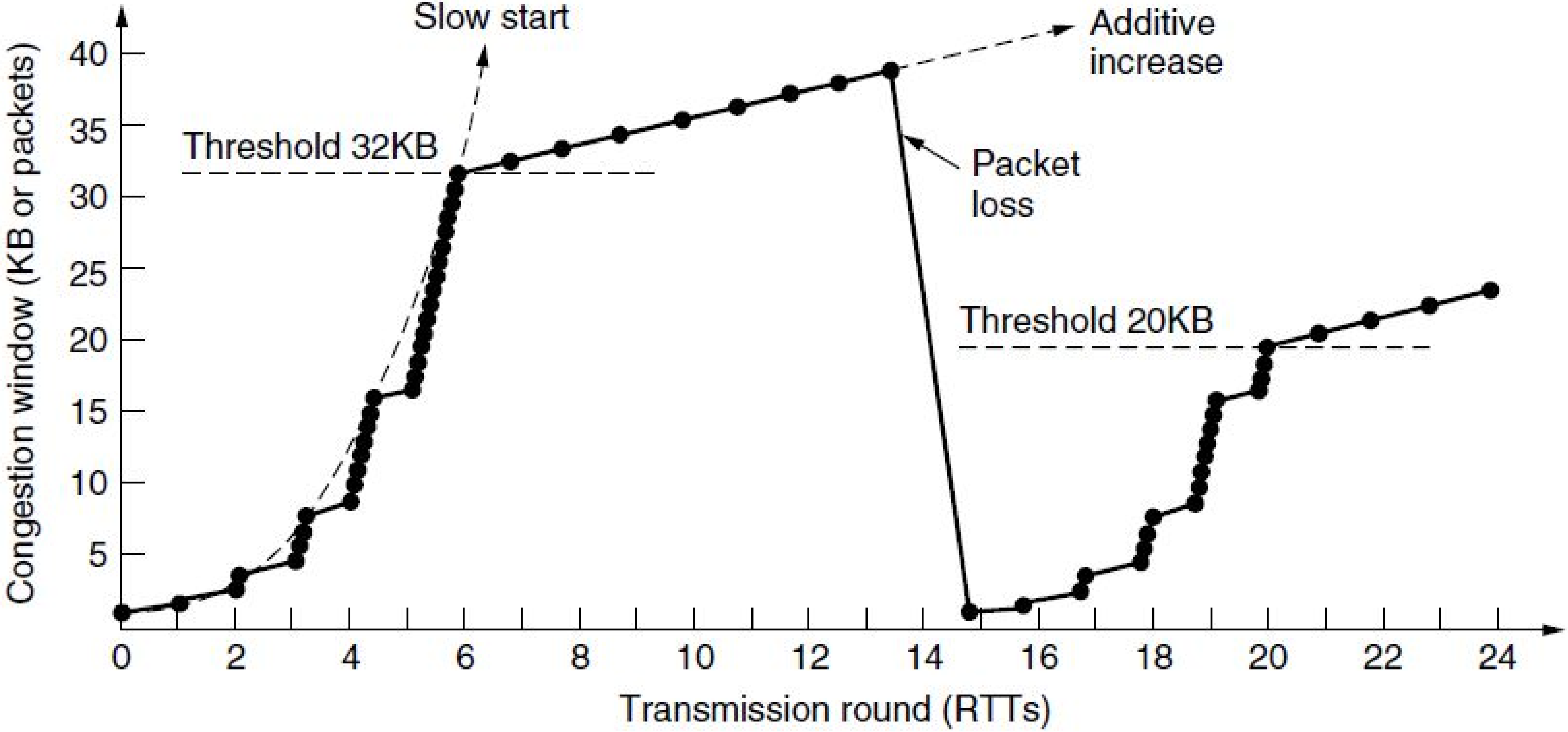
TCP Slow Start



TCP Additive Increase

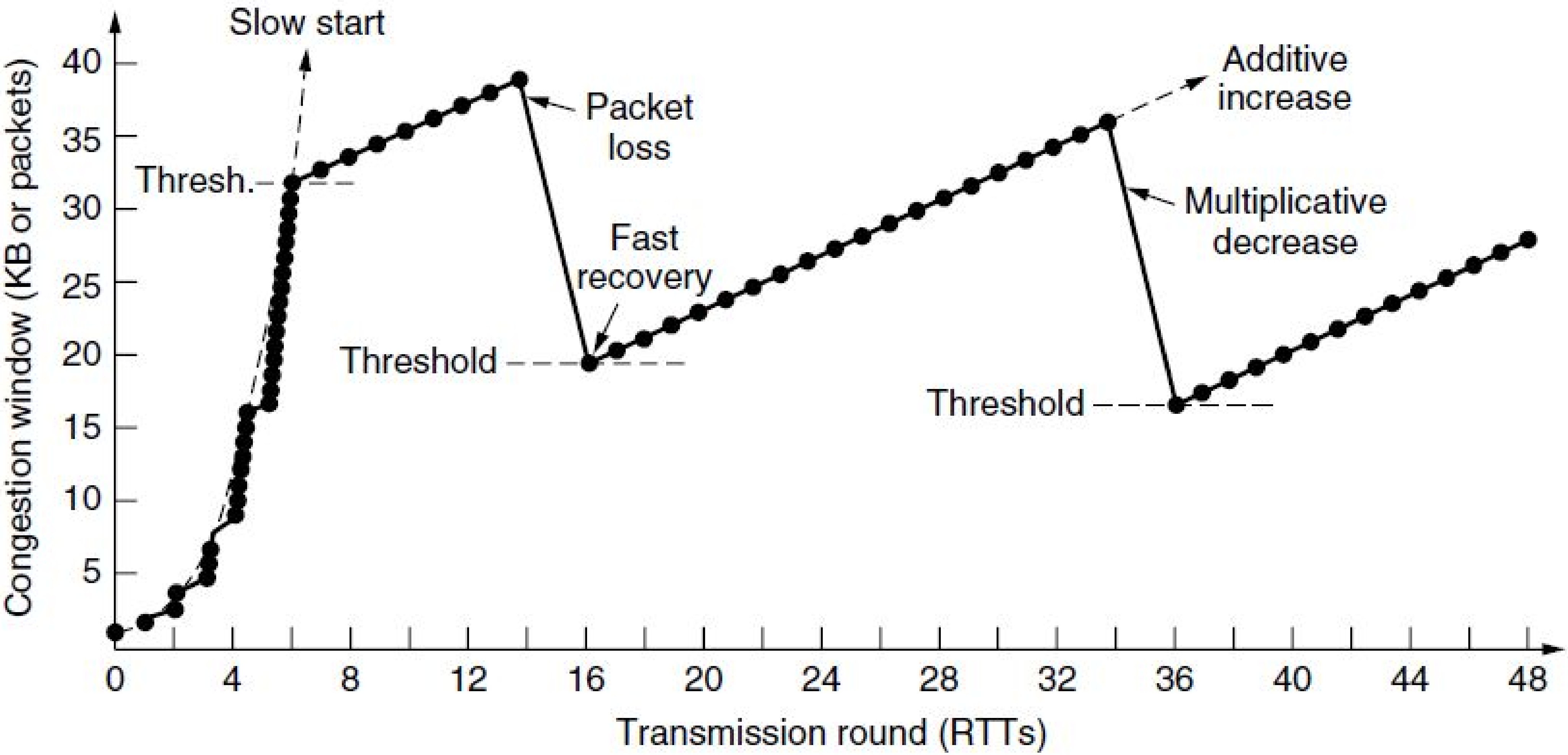


TCP Tahoe



Slow start
Additive increase
Threshold 32KB
Packet loss
Threshold 20KB
Congestion window (KB or packets)
40
35
30
25
20
15
10
5
0
2
4
6
8
10
12
14
16
18
20
22
24
Transmission round (RTTs)

TCP Reno



Slow start
Packet loss
Thresh.
Fast recovery
Threshold
Additive increase
Multiplicative decrease
Threshold
Congestion window (KB or packets)
40
35
30
25
20
15
10
5
0
4
8
12
16
20
24
28
32
36
40
44
48
Transmission round (RTTs)