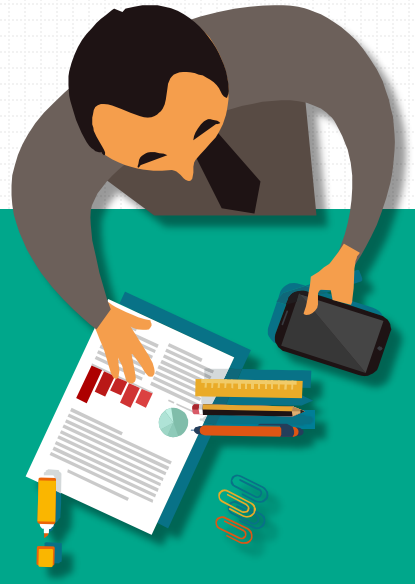
協定基礎
(Protocol Basis)

異質多網多媒體服務

國立臺北科技大學電子工程系
授課教師：李昭賢 副教授
電子郵件：chlee@ntut.edu.tw
校內分機：2288



學習目標 Outline

- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
  - Reliable, In-order Delivery
  - Congestion Control
  - Flow Control

User Datagram Protocol (UDP)

NTUT NESL

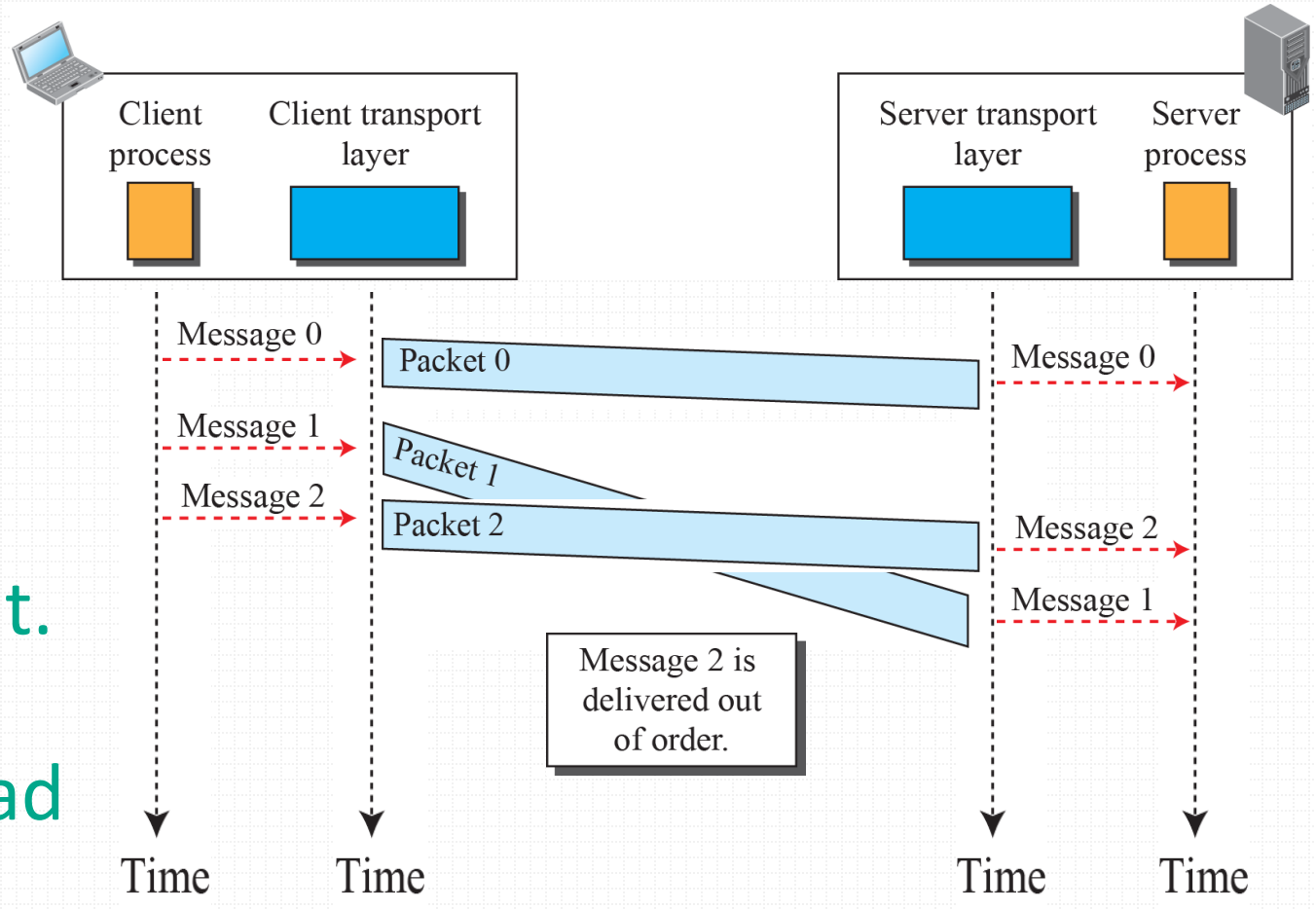
User Datagram Protocol (UDP)

- RFC 768

  - Also known as STD 6.

User Datagram Protocol (UDP)

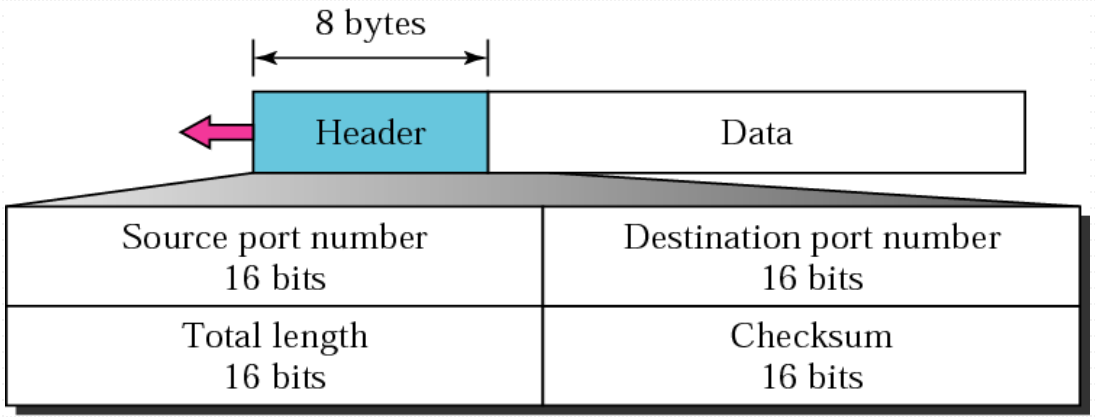
- "no-frills" and "bare-bones"
- "best-effort" service
  - Packet loss.
  - Delivered out of order to applications.
- Connectionless
  - No connection establishment.
  - No connection state.
  - Small packet header overhead
  - Each UDP segment handled independently of others

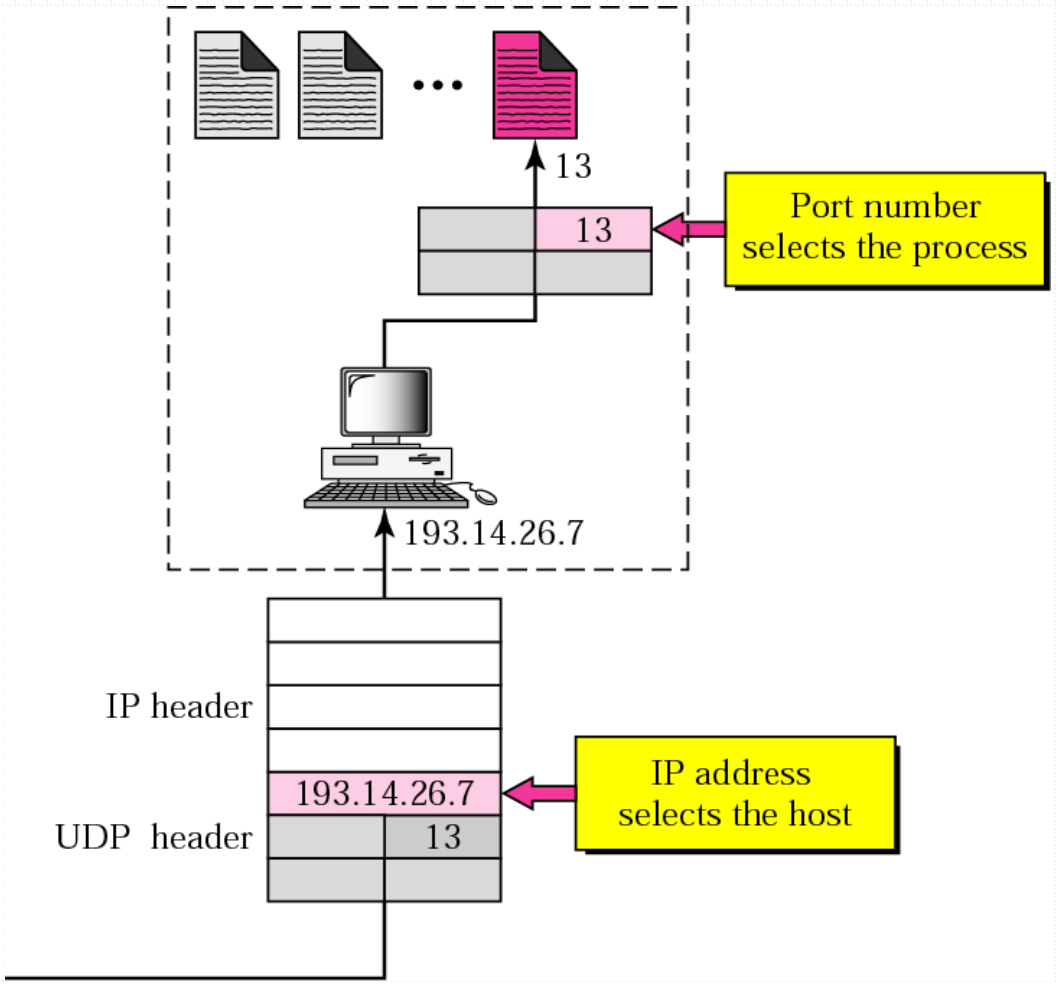


圖片來源：Computer Networks - A Top Down Approach，McGraw-Hill出版

User Datagram Protocol (UDP)

- UDP socket identified by 2-tuple
  - (src IP, src port)
  - (dst IP, dst port)





圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

總結

- Often used for streaming multimedia applications
  - loss tolerant, rate sensitive
- Other usages
  - DNS, SNMP, etc.
- Reliable transfer over UDP
  - add reliability at application layer
  - application-specific error recovery

Transmission Control Protocol (TCP)

NTUT NESL

Transmission Control Protocol (TCP)

- RFC 793

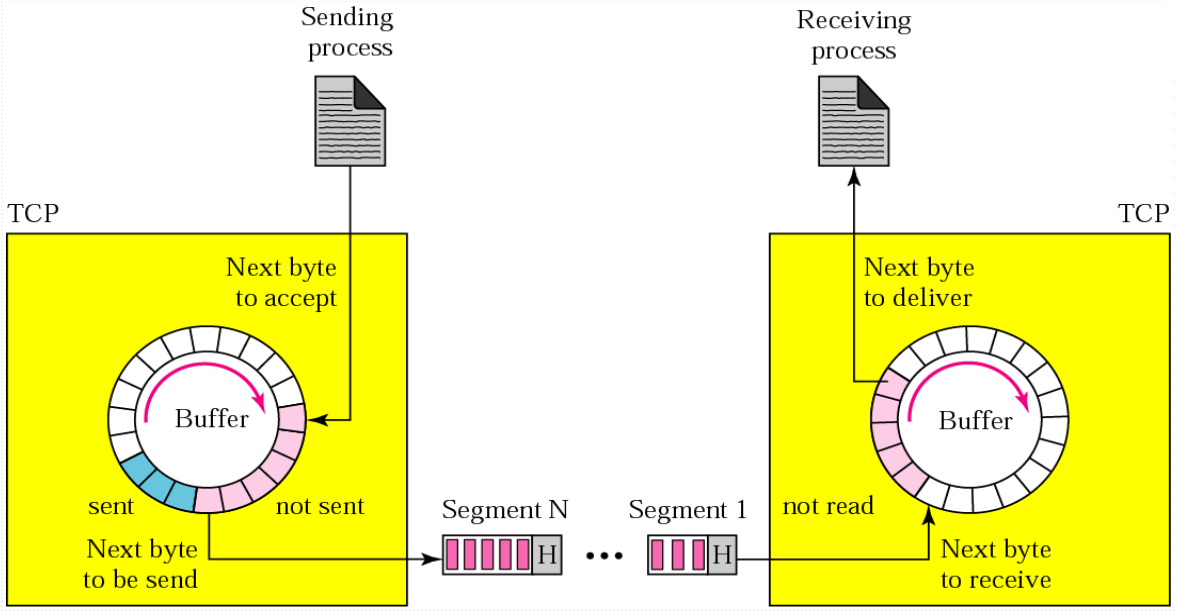
  - Also known as STD 7.
  - Obsoletes RFC 761
  - Updated by RFC 6093, RFC 3168, RFC 1122, RFC 6528

Transmission Control Protocol (TCP)

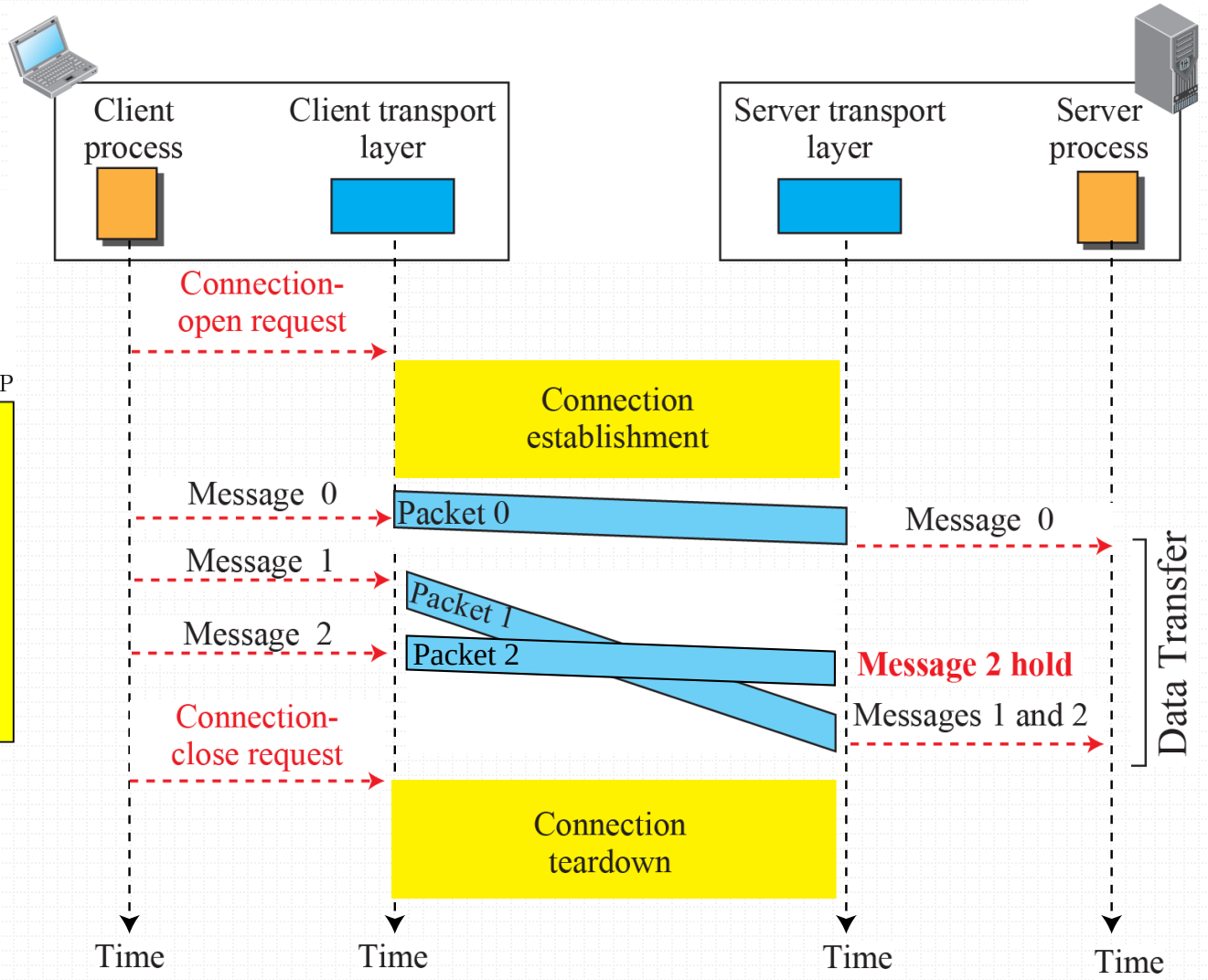
- Connection-oriented
  - Three-way handshaking is initialized before data exchange.
  - Four-way handshaking is triggered after data exchange.
- Reliable, in-order byte steam
  - Point-to-point：one sender & one receiver
  - no “message boundaries”
- Flow Control
  - Sender will not overwhelm receiver.
- Congestion Control
  - Too many data are transmitted beyond the network bandwidth.

- Send Buffer
  - Maximum Segment Size (MSS)
  - Sliding Window size

Transmission Control Protocol (TCP)

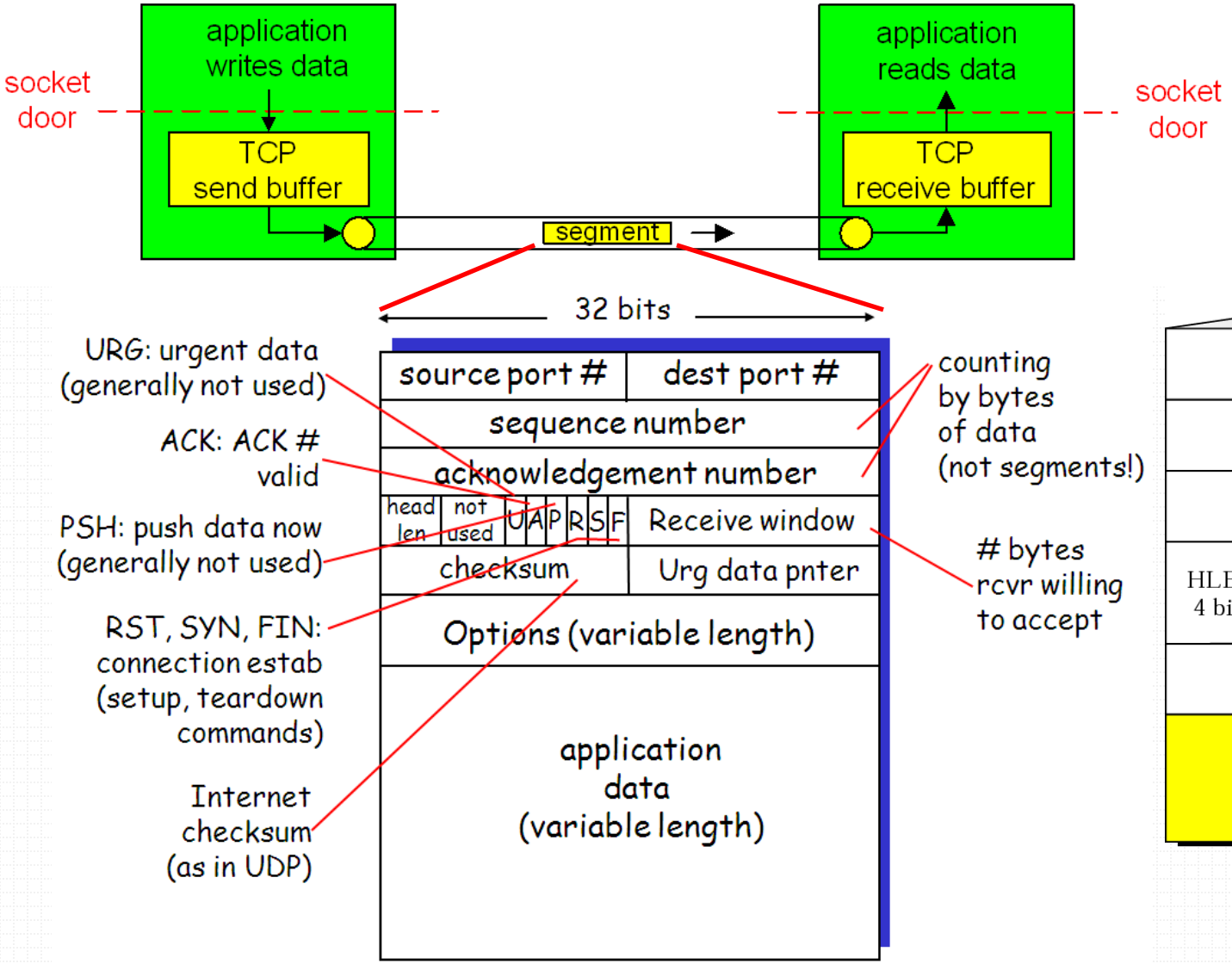


圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

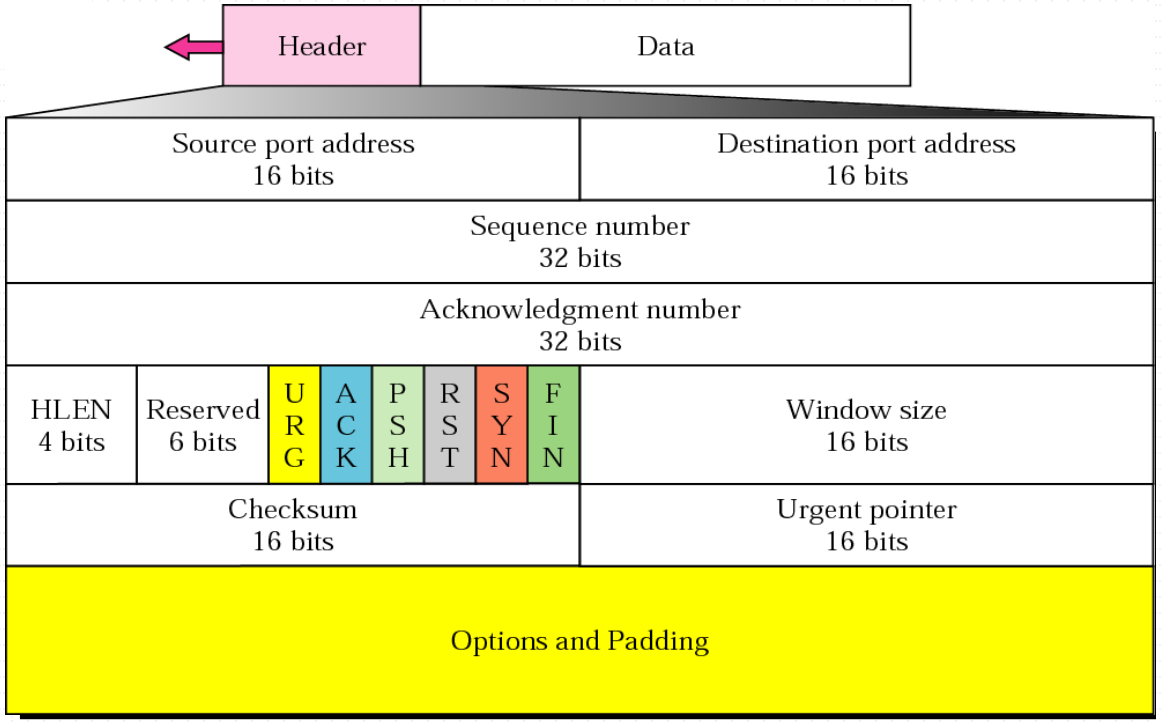


圖片來源：Computer Networks - A Top Down Approach，McGraw-Hill出版

TCP Packet Format



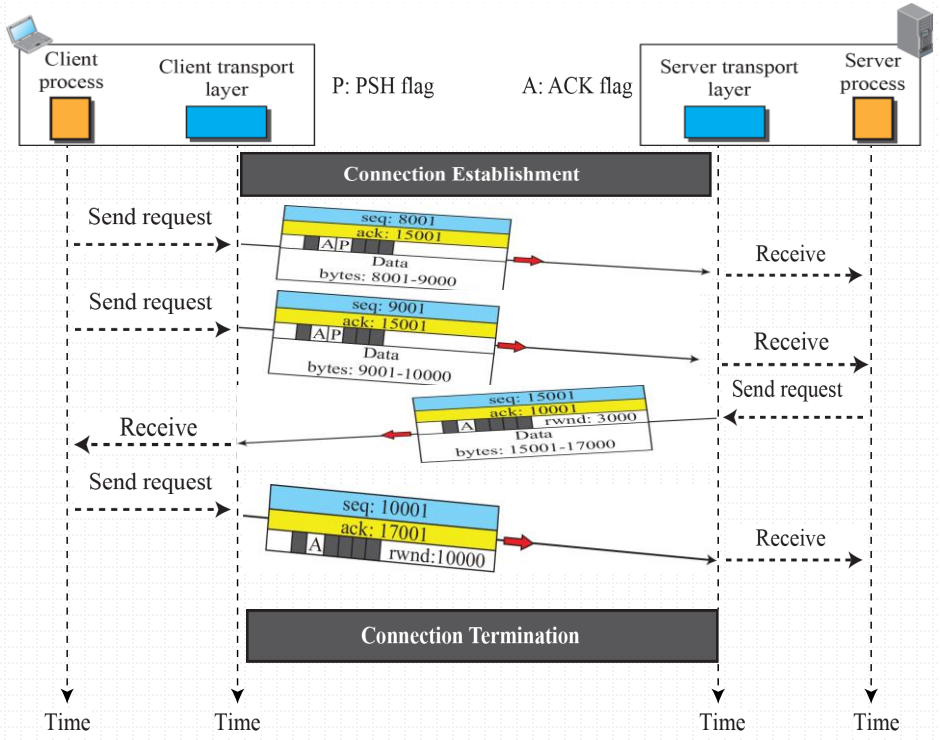
圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版



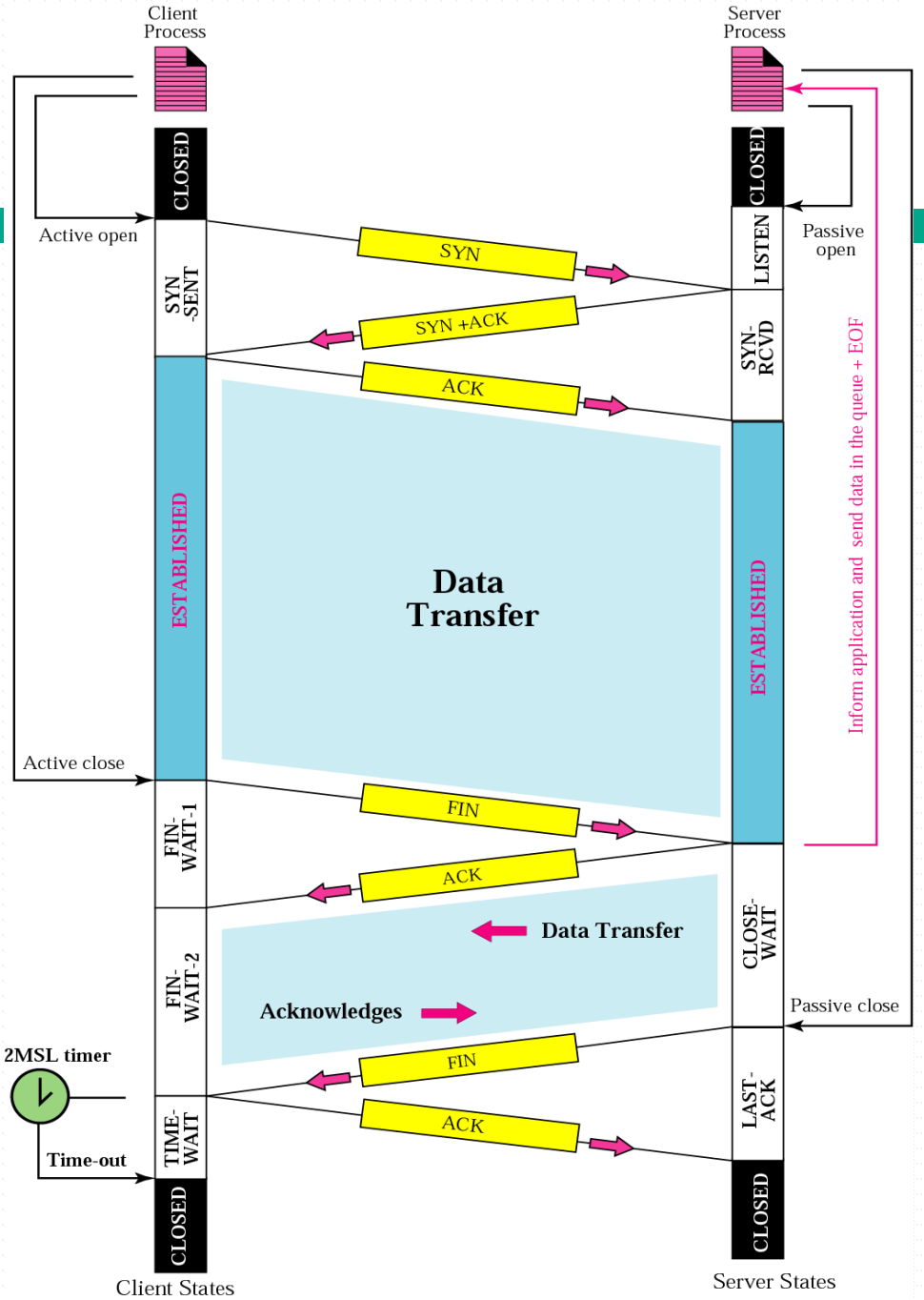
圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Connection Management

- TCP sender and receiver should establish a “connection” before exchanging data segments.



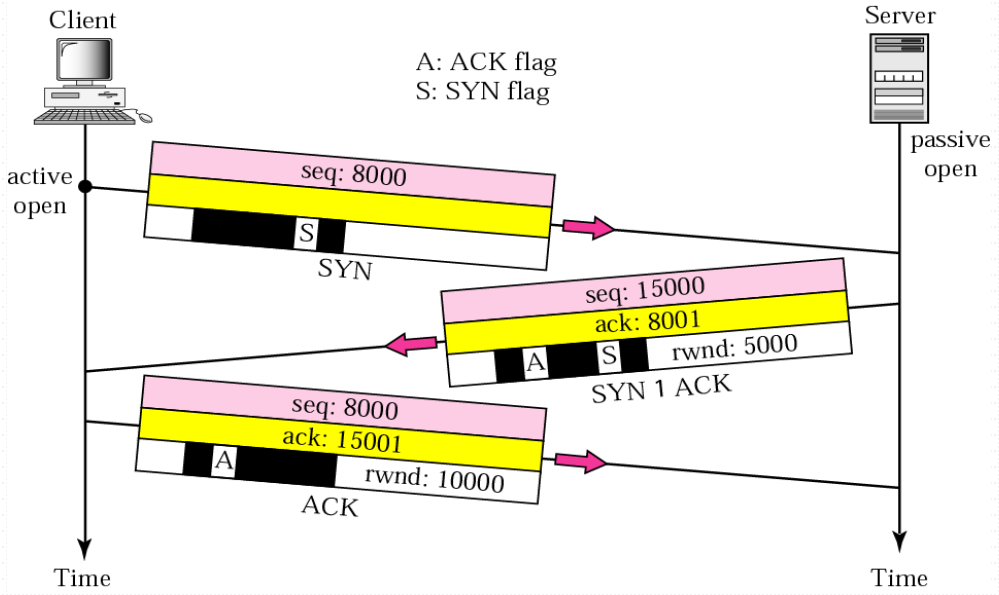
圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版



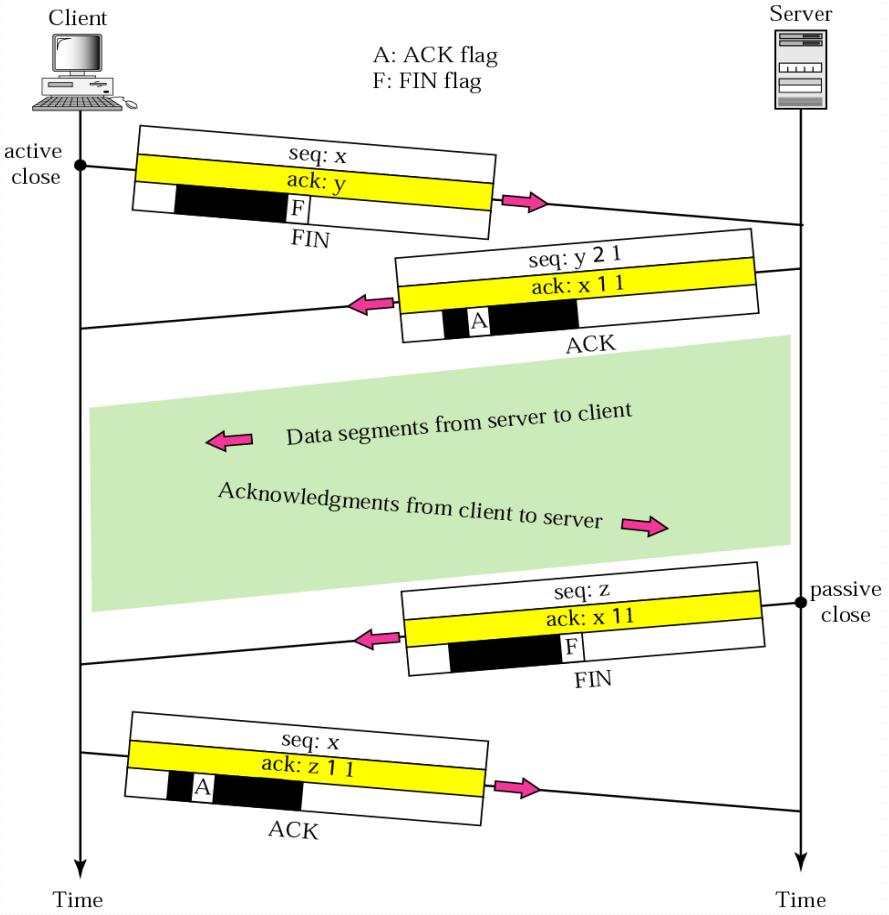
圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Connection Management

- Connection Establishment



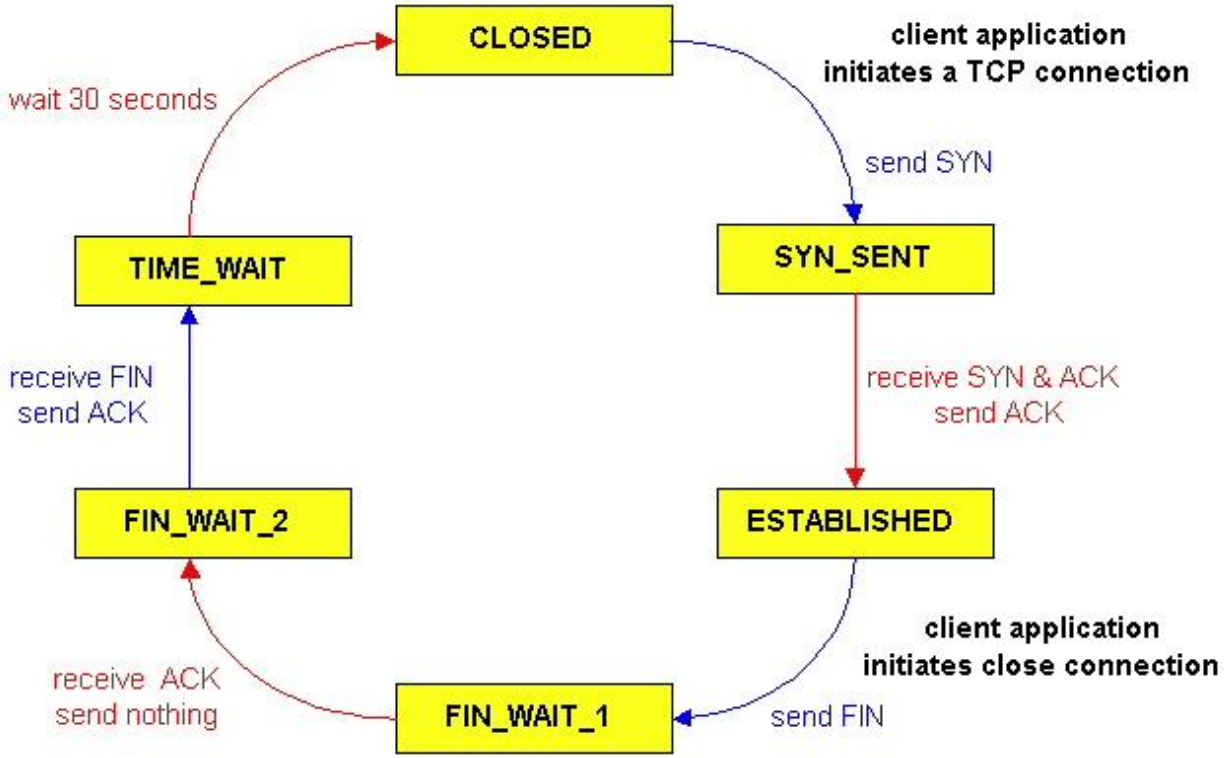
- Connection Termination



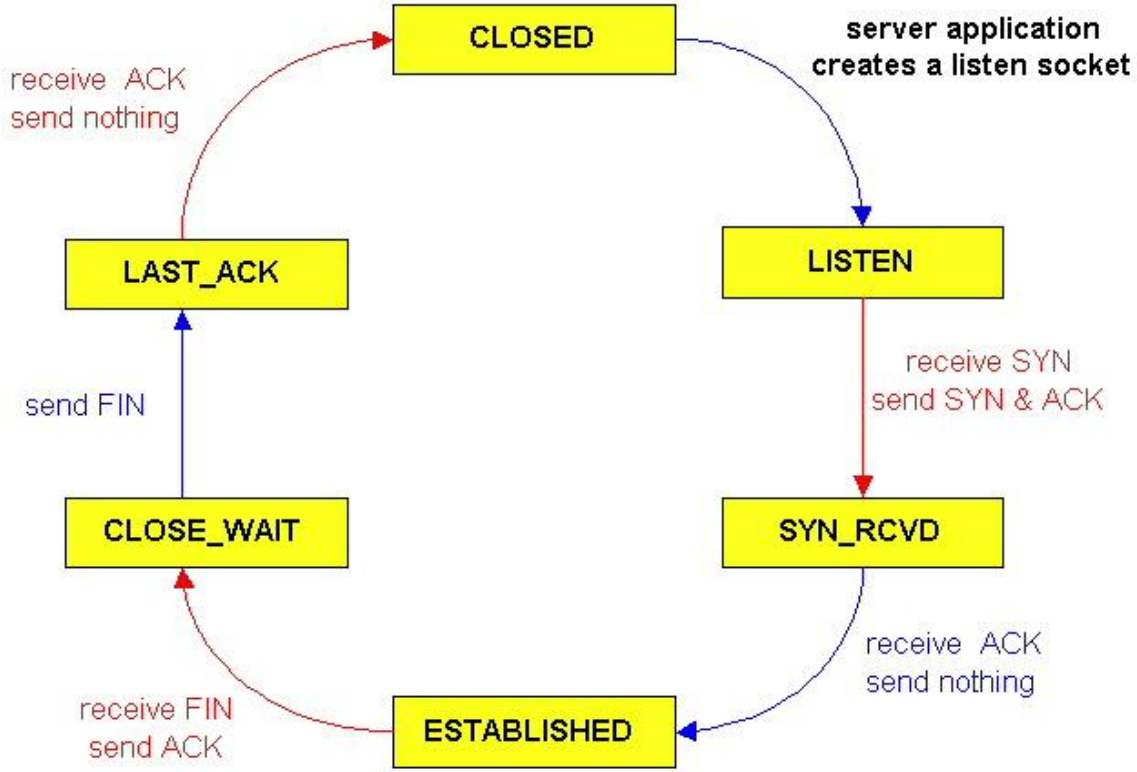
圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Connection Management

- Client lifecycle



- Server lifecycle



圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Reliable

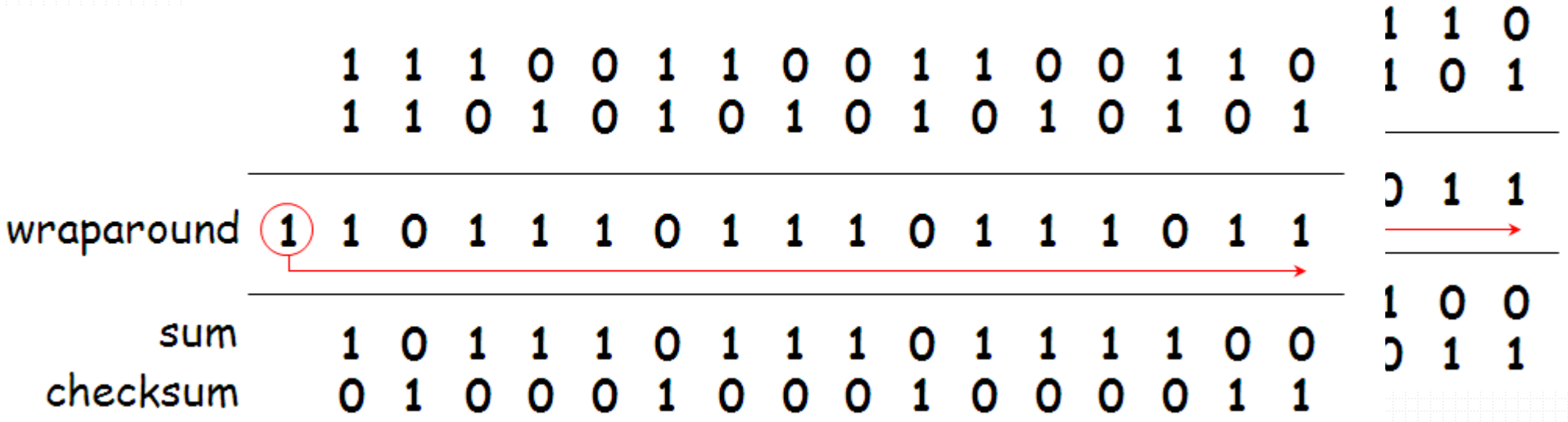
1. Checksum
   - To detect “errors” (e.g., flipped bits) in transmitted segment.
2. Automatic Repeat Request (ARQ)
   - To determine which bits in a packet may be corrupted.
3. Sequence Number
4. Countdown Timer
   - To determine whether a data packet or/and an ACK was lost or if the packet or ACK was delayed or not.
5. (Optional) Pipeline
   - Go-Back-N (GBN) and Selective Repeat (SR).

Checksum

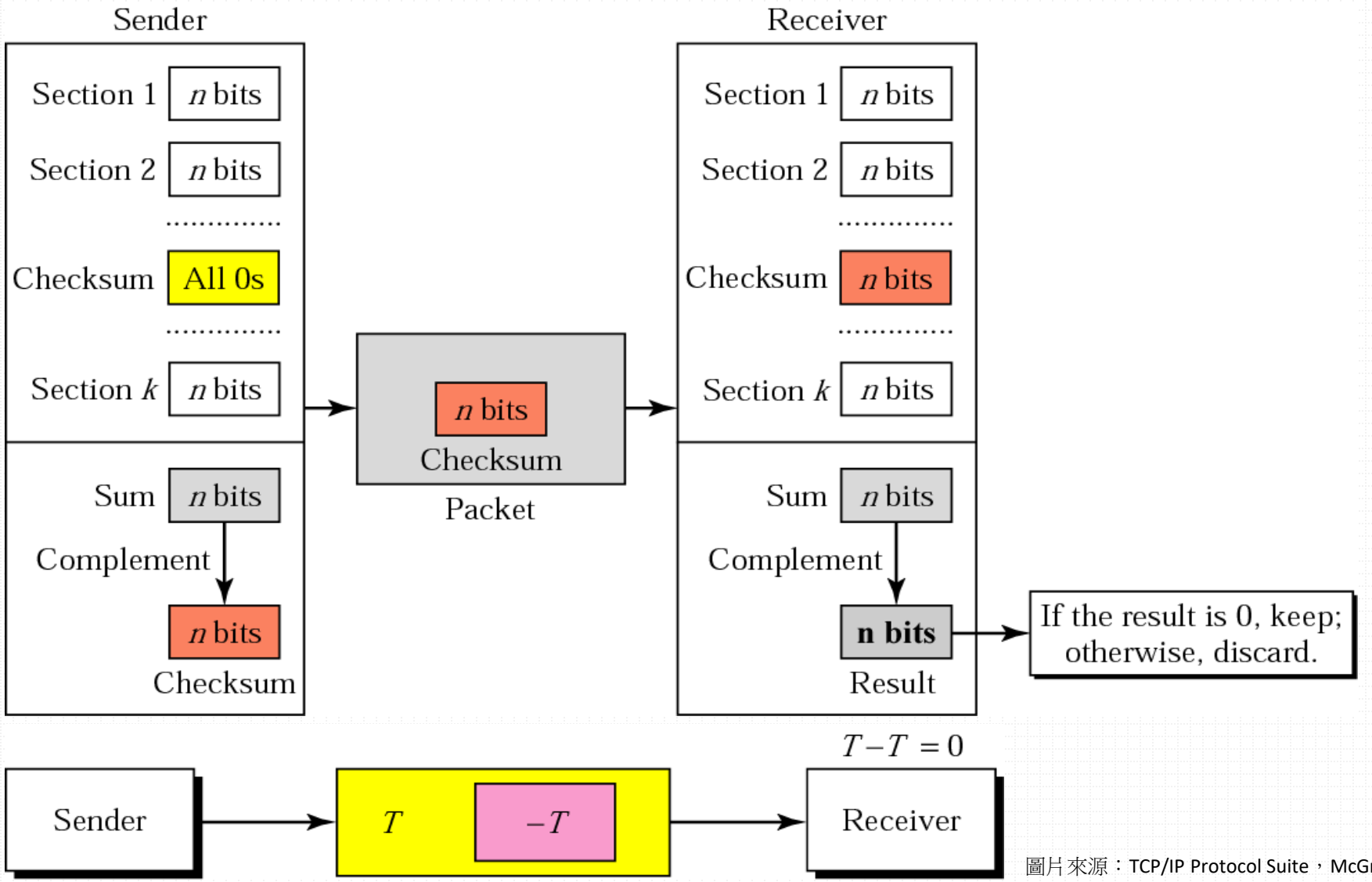
- To detect “errors” (e.g., flipped bits) in transmitted segment.
  - Sender
    - treat segment contents as sequence of 16-bit integers
    - addition (1’s complement sum) of segment contents
    - puts checksum value into header field.
  - Receiver
    - compute checksum of received segment
    - check if computed checksum equals to header field
      - YES - no error detected (adding including checksum will be all 1).
      - NO - error detected (one of the bits is a 0.)

Checksum

- Example：add two 16-bit integers



Checksum



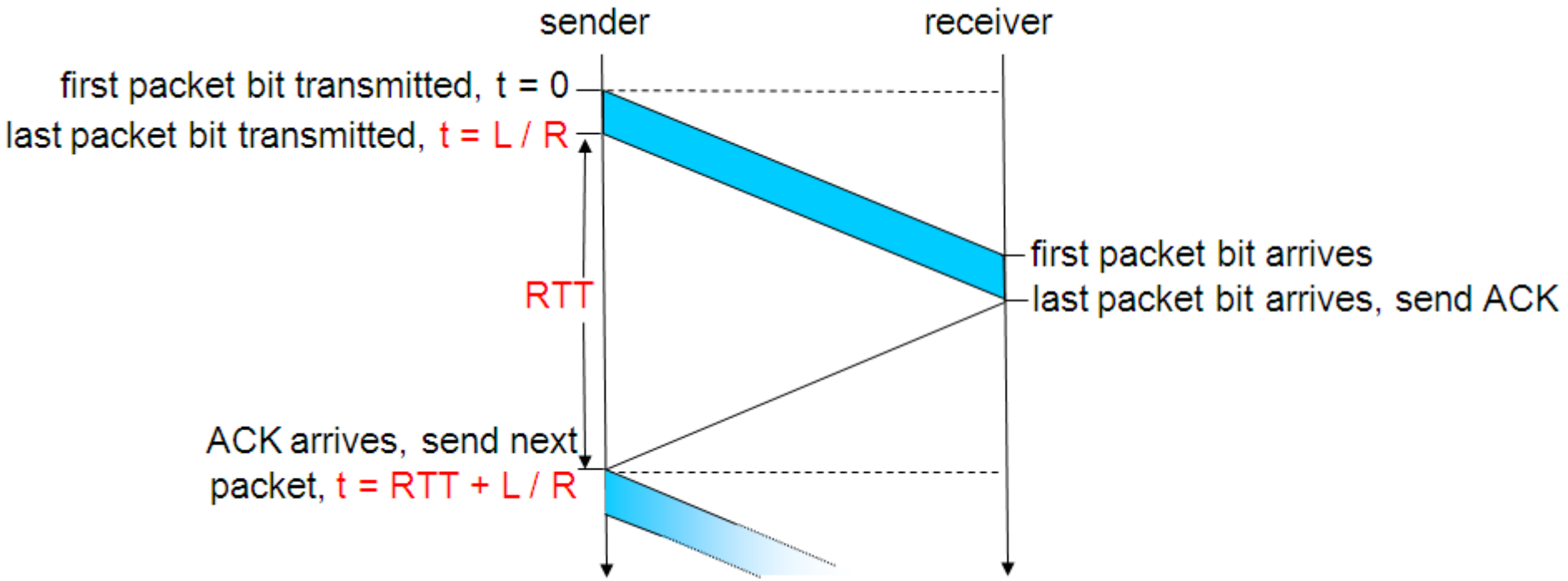
圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Automatic Repeat Request (ARQ)

- To determine which bits in a packet may be corrupted.
  1. Error Detection
     - Checksum
  2. Receiver Feedback
     - acknowledgements (ACKs)
       - receiver explicitly tells sender that the packet is OK.
     - negative acknowledgements (NAKs)
       - receiver explicitly tells sender that the packet has errors.
  3. Retransmission

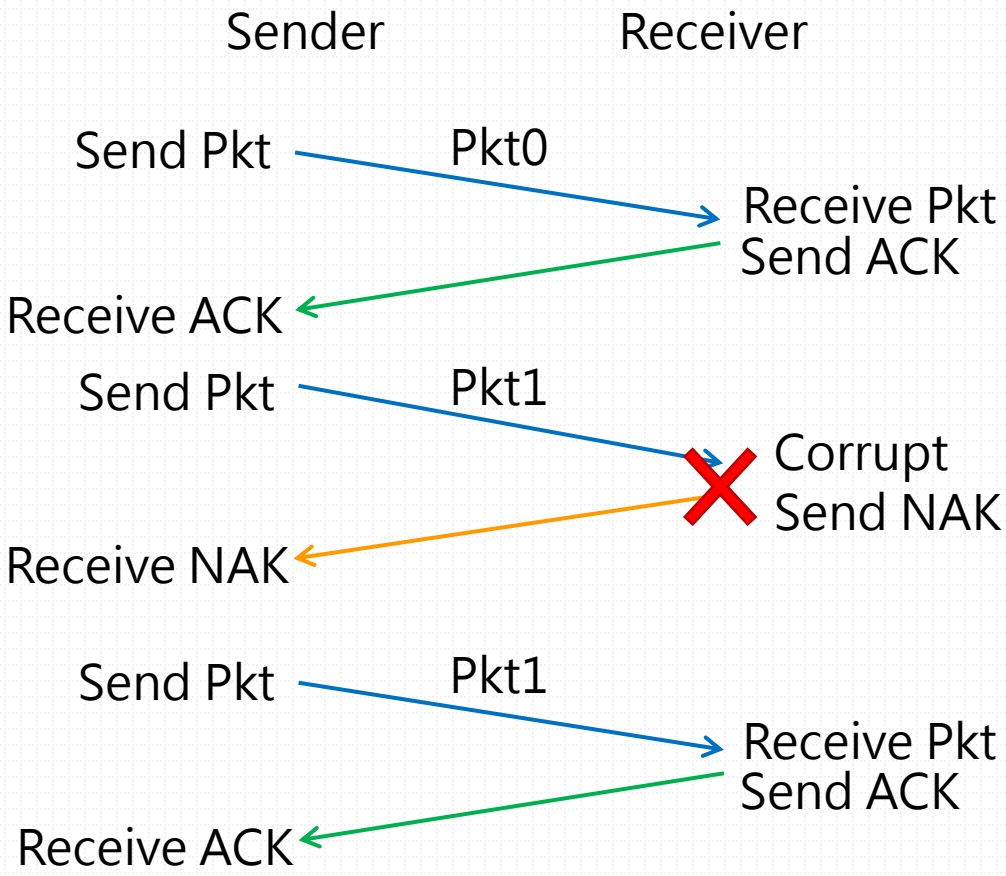
Automatic Repeat Request (ARQ)

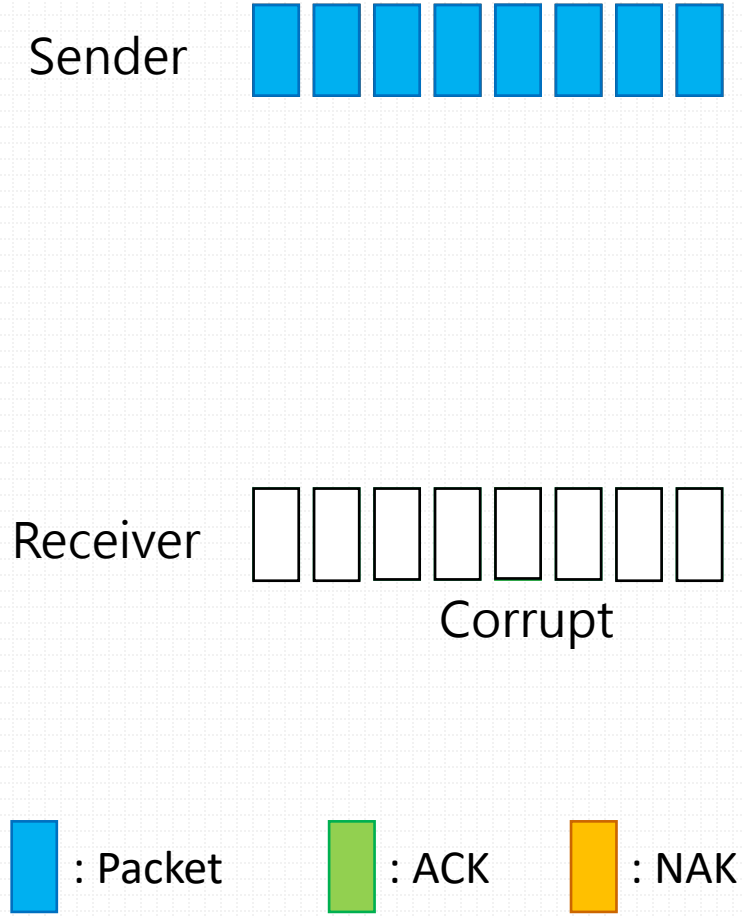
- Stop-and-wait
  - Sender sends one packet, then waits for receiver response.



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Example：NAK


Sender
Receiver
Send Pkt
Pkt0
Receive Pkt
Send ACK
Receive ACK
Send Pkt
Pkt1
Corrupt
Send NAK
Receive NAK
Send Pkt
Pkt1
Receive Pkt
Send ACK
Receive ACK


Sender
Receiver
Corrupt
: Packet
: ACK
: NAK

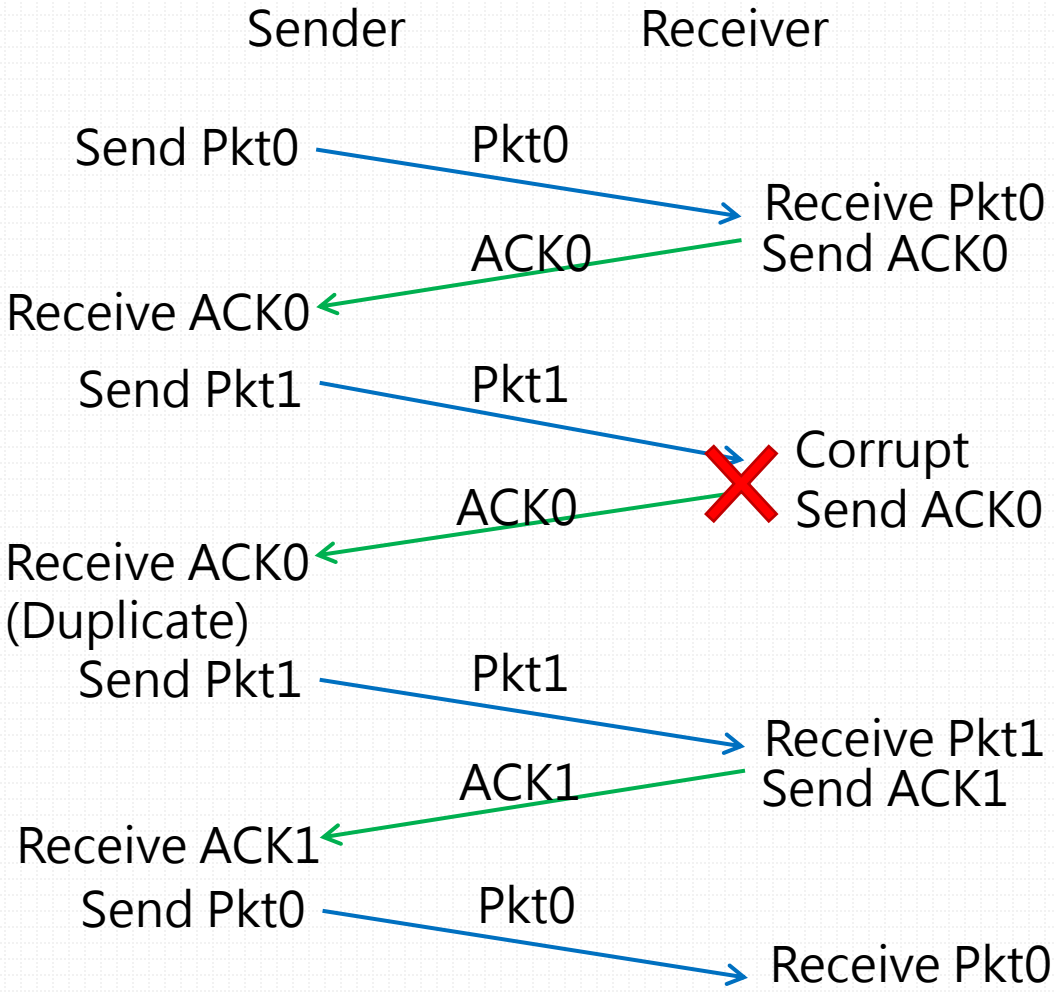
Sequence Number

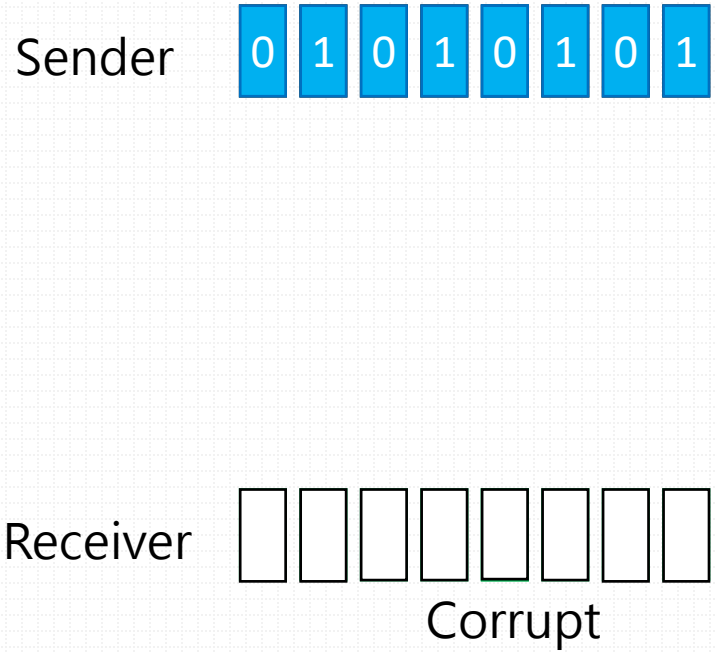
- To handle corrupted ACKs or NAKs.
  - sender doesn’t know what happened at receiver!
  - can’t just retransmit
    - possible duplicate
- How many bits for sequence number?

NAK-free

- Instead of NAK, receiver sends ACK for last packet received OK
  - Receiver must explicitly includes sequence number of packet being ACKed.
- Duplicate ACK at sender results in same action as NAK.
  - retransmit current packet

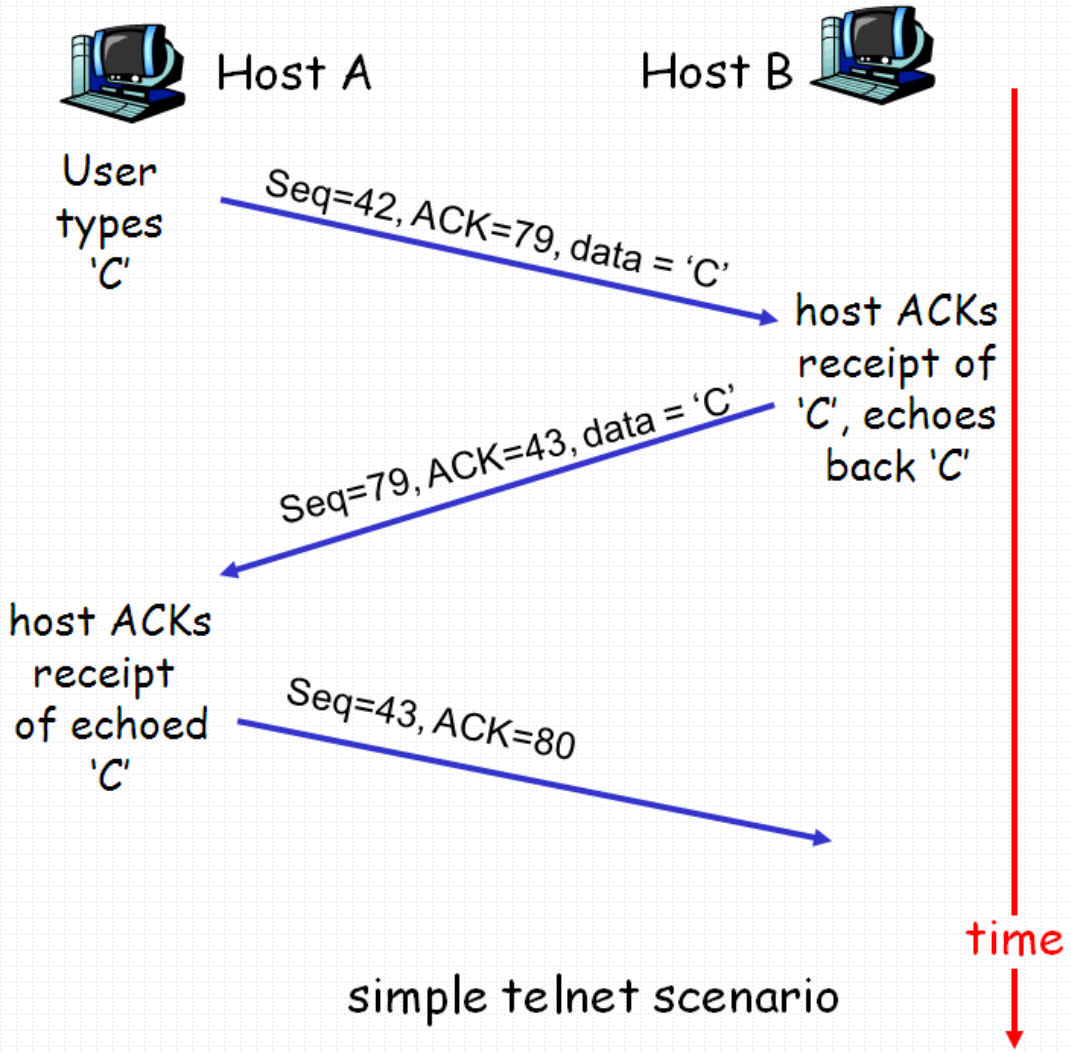
Example：NAK-free





Cumulated & Piggybacked ACK

- Cumulated
  - Sequence number of expected next byte.
- Piggybacked
  - ACK for client-to-server data is carried in a segment carrying server-to-client data.

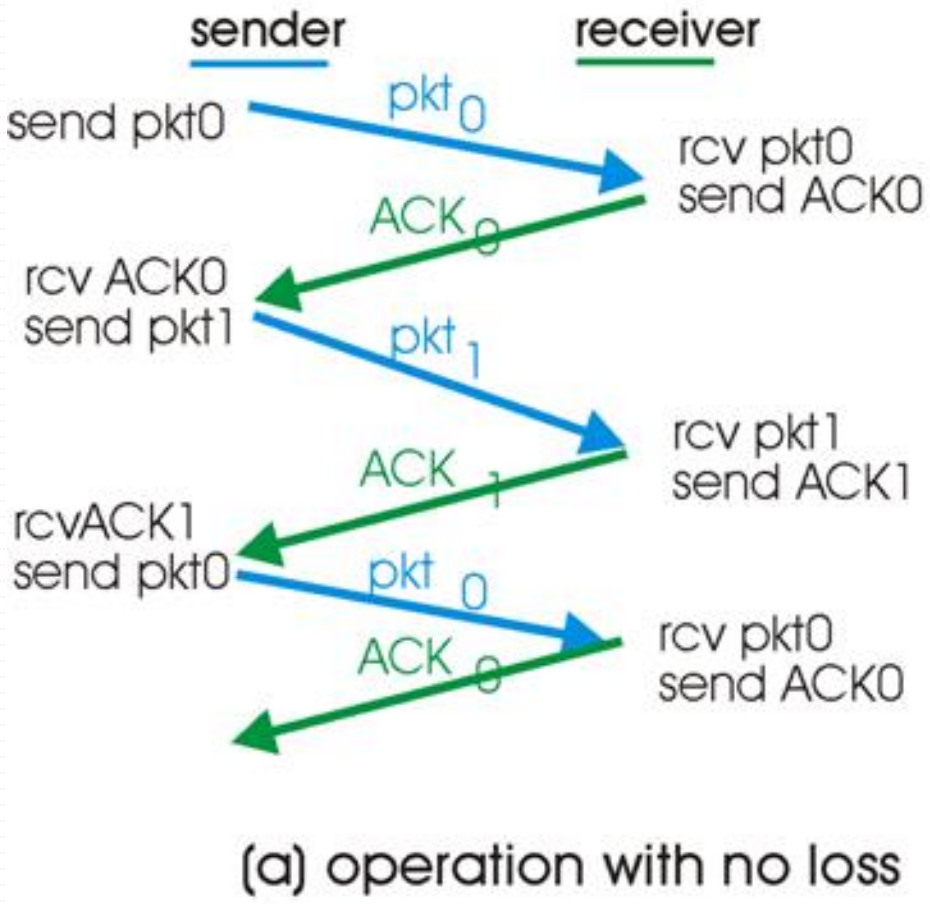


圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Countdown Timer

- To determine whether a data packet or/and an ACK was lost or if the packet or ACK was delayed or not.
- Time-based retransmission mechanism
  - Interrupt the sender after a given amount of time has expired.
  - The process is as follows.
    1. Start the timer each time a packet (a first-time packet or a retransmission) is sent.
    2. Respond to a timer interrupt
    3. Stop the timer

Example：Normal



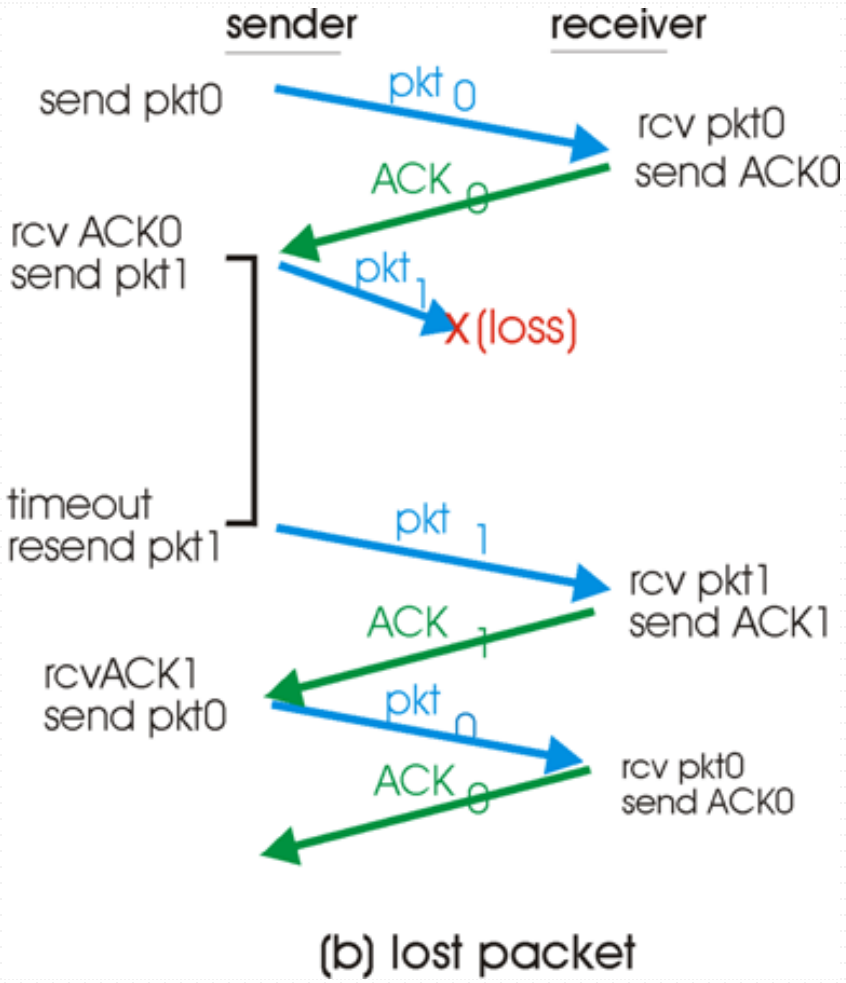
圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Sender 0 1 0 1 0 1 0 1

Receiver

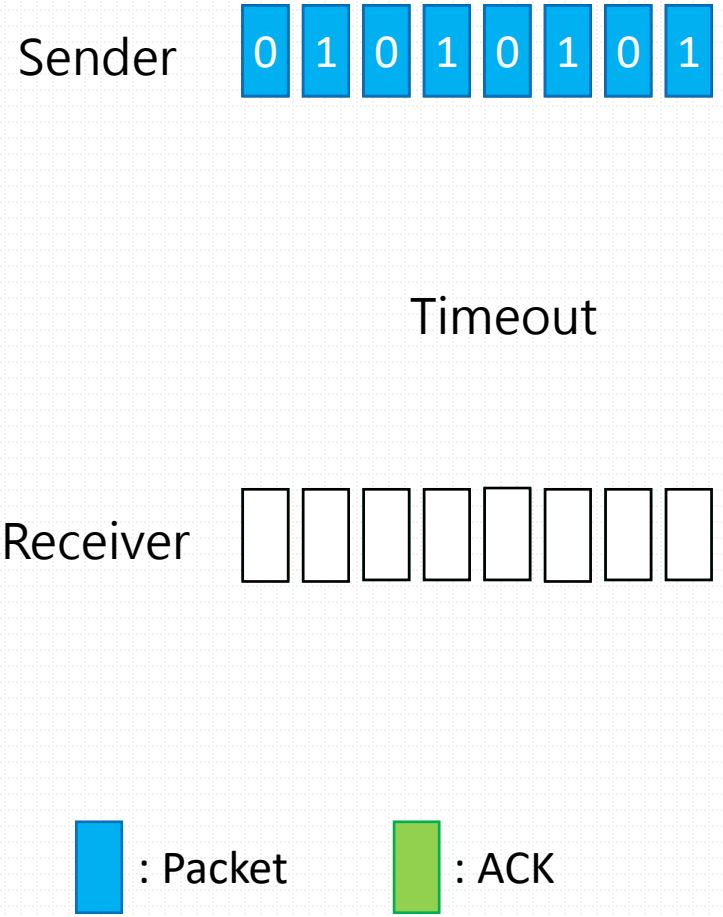
: Packet : ACK

Example：Packet Loss

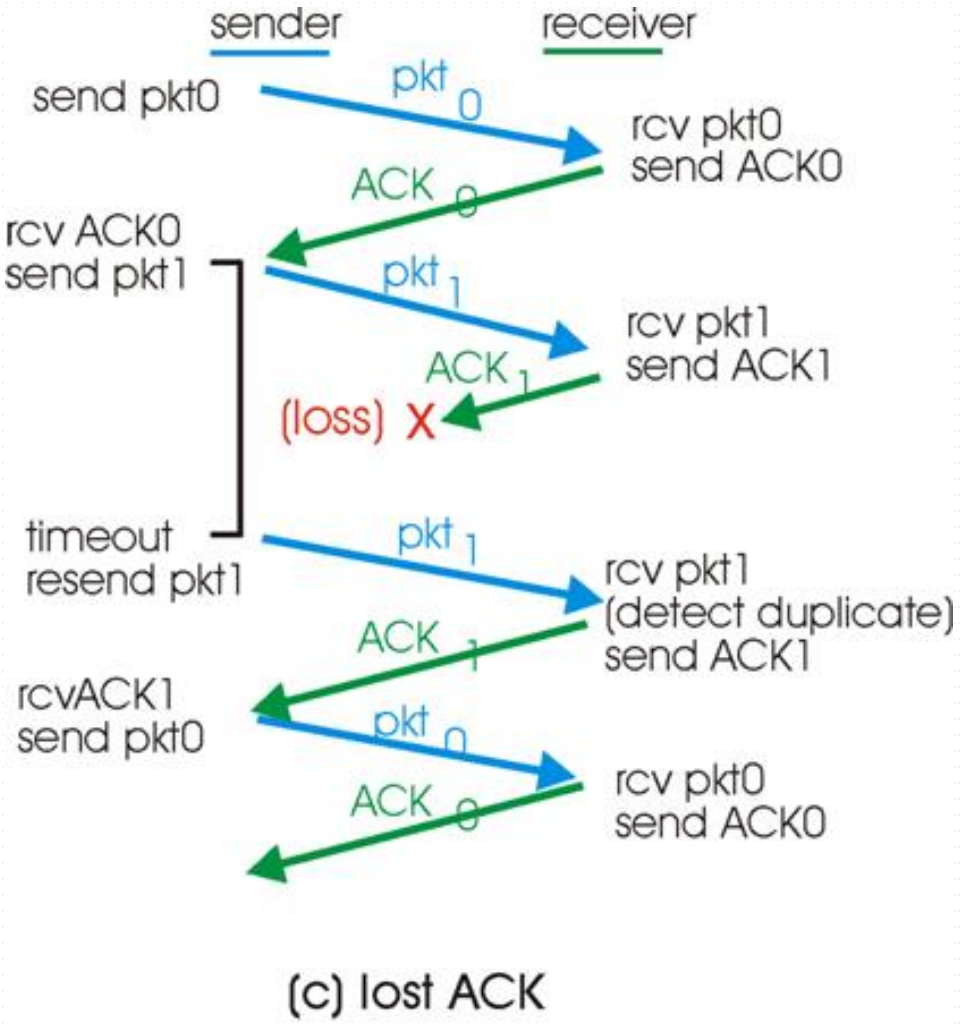


(b) lost packet

圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

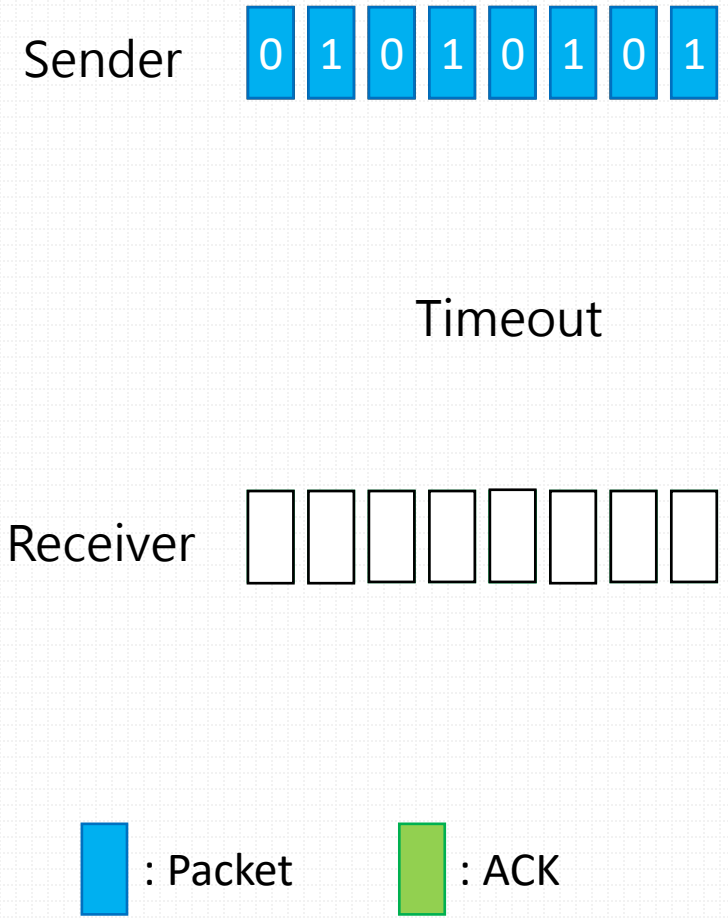


Example：ACK Loss

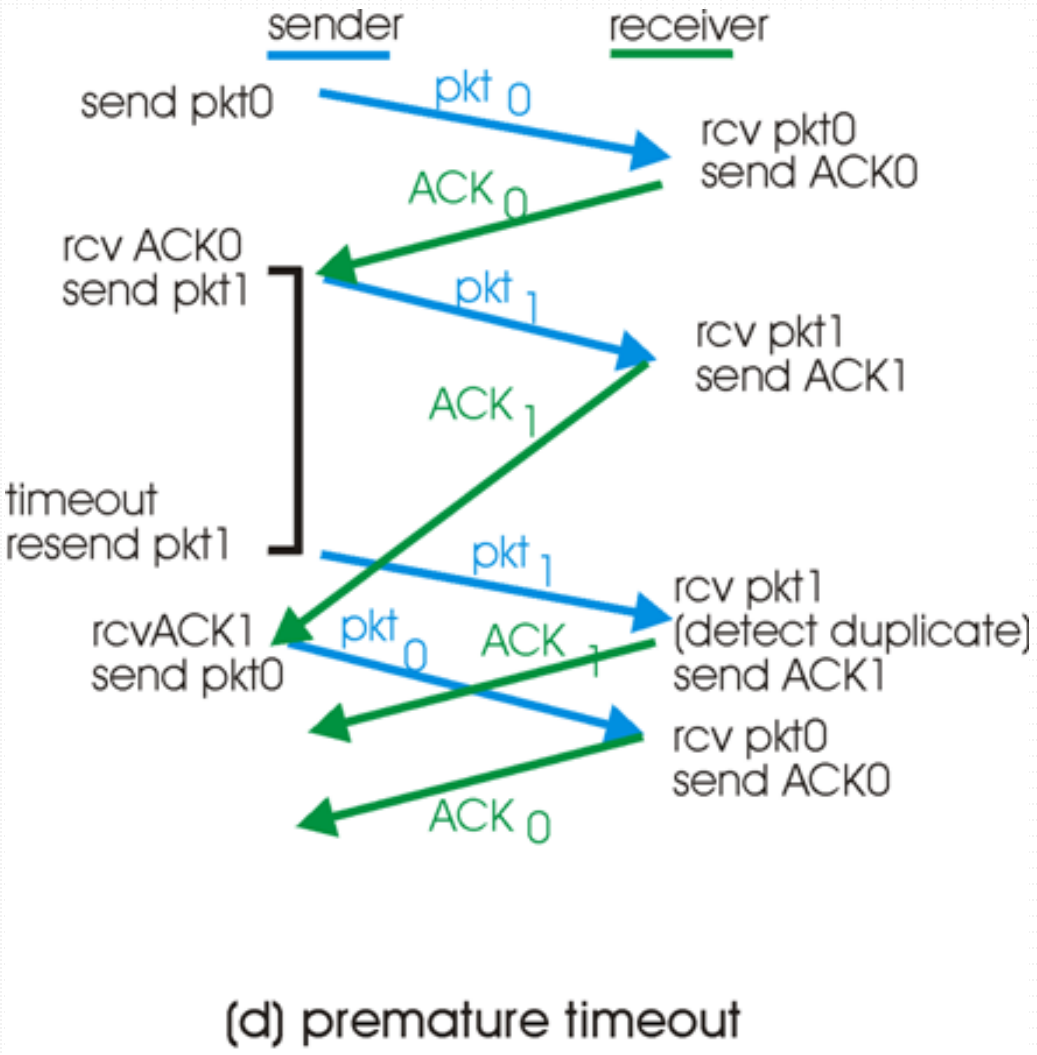


(c) lost ACK

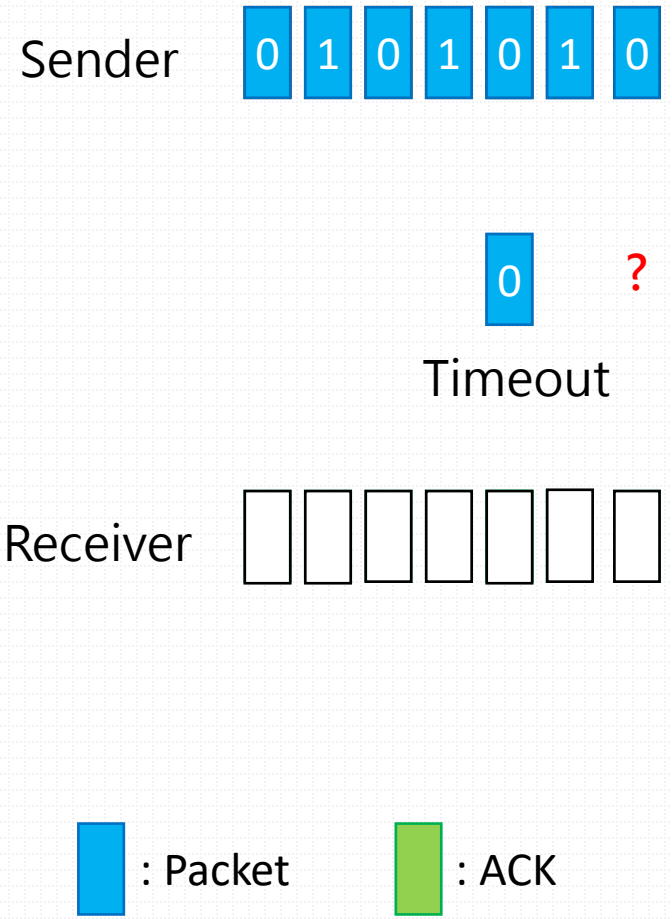
圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版



Example：Premature Timeout



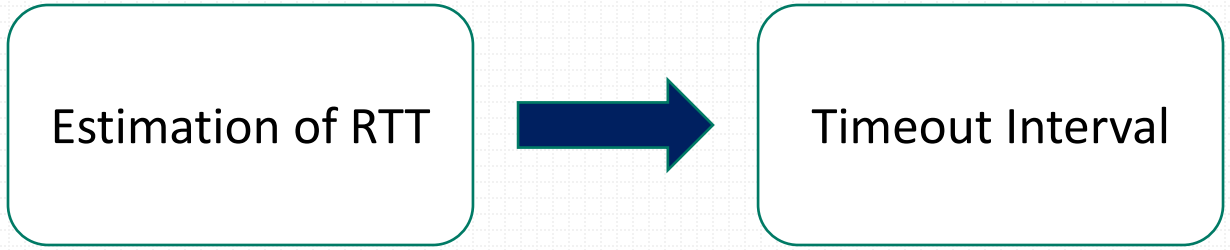
(d) premature timeout



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Countdown Timer

- How to set TCP timeout value?
  - Longer than RTT
    - too short
      - premature timeout
      - unnecessary retransmissions
    - too long
      - slow reaction to segment loss



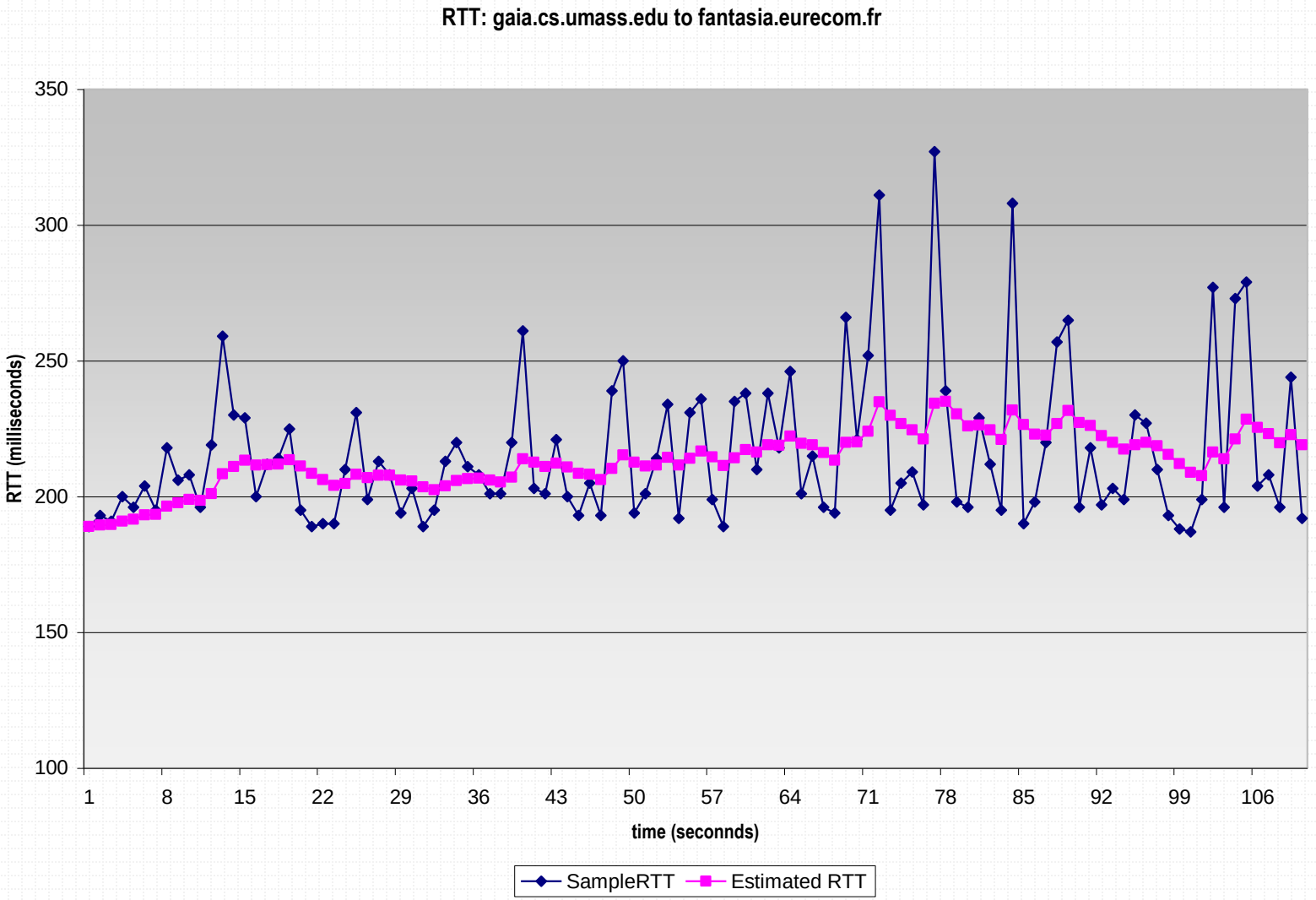
Countdown Timer

- How to estimate RTT?
  - SampleRTT
    - measured time from segment transmission until ACK receipt

      `EstimatedRTT = (1- α)*EstimatedRTT + α*SampleRTT`

    - Exponential Weighted Moving Average (EWMA)
    - Influence of past sample decreases exponentially fast
    - Typical value: $\alpha = 1/8 = 0.125$

Countdown Timer



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Countdown Timer

- In addition to having an estimation of the RTT, it is also valuable to have a measure of the variability of the RTT.
  - DevRTT: RTT variation
  - EstimtedRTT plus "safety margin"
    - Large variation in EstimatedRTT -> larger safety margin

      DevRTT = (1-β)*DevRTT + β*|SampleRTT - EstimatedRTT|

    - Typical value: β = 1/4 = 0.25
- Timeout Interval

  TimeoutInterval = EstimatedRTT + 4*DevRTT

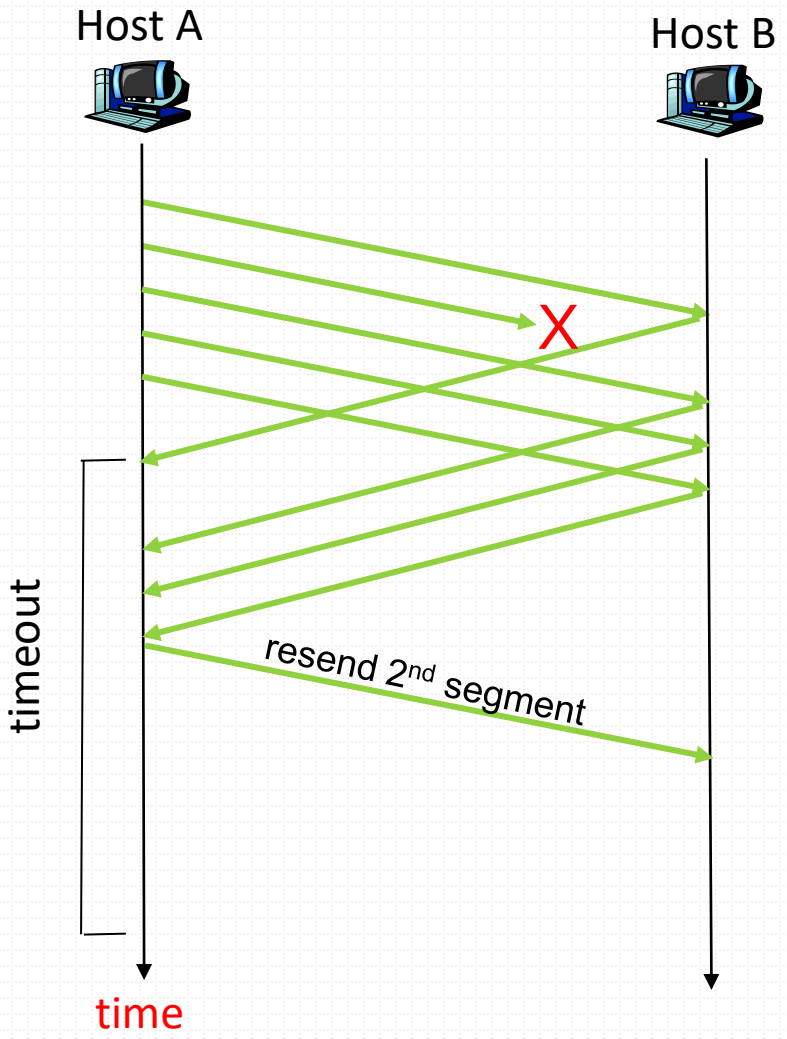
Countdown Timer

- Doubling the Timeout Interval
  - Set the next timeout interval to twice the previous value.
    - The intervals grow exponentially after each retransmission.
  - This modification provides a limited form of congestion control.
    - The timer expiration is most likely caused by congestion in the network.

Countdown Timer

- Re-send segment before timer expires.
  - Three duplicate ACKs

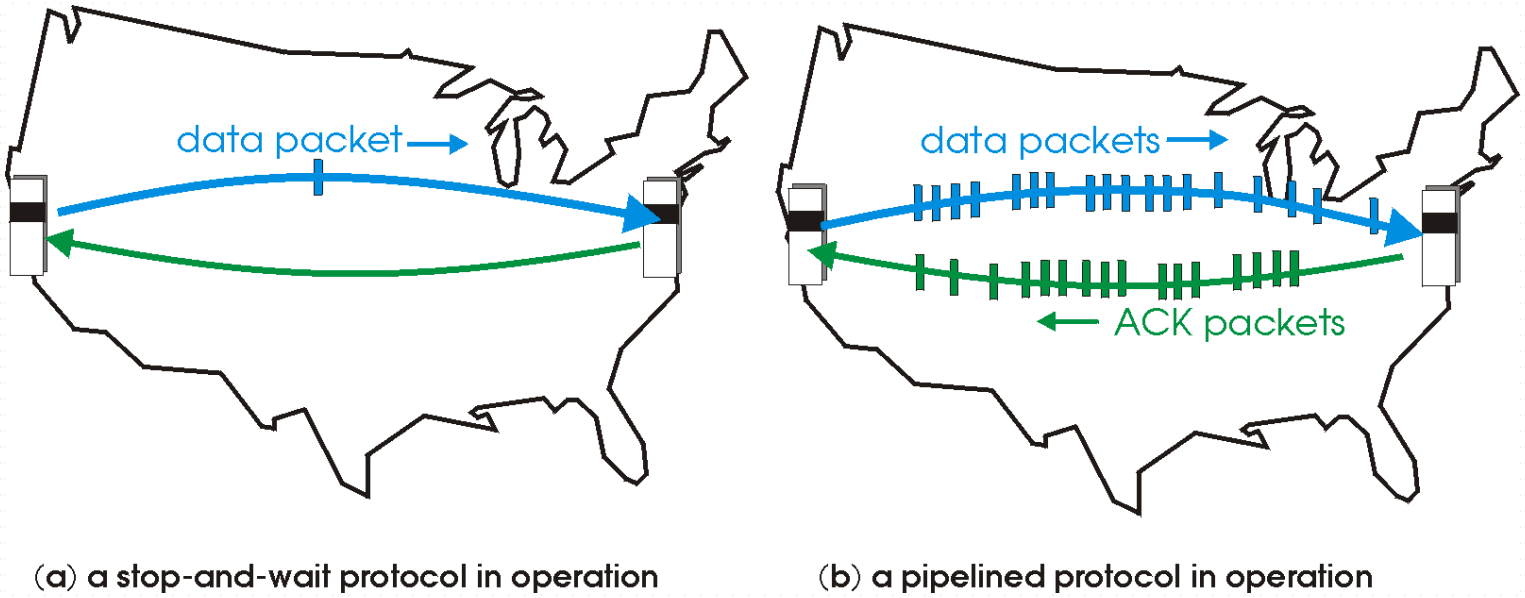
Event at Receiver	TCP Receiver action
Arrival of in-order segment with expected seq #. All data up to expected seq # already ACKed	Delayed ACK. Wait up to 500ms for next segment. If no next segment, send ACK
Arrival of in-order segment with expected seq #. One other segment has ACK pending	Immediately send single cumulative ACK, ACKing both in-order segments
Arrival of out-of-order segment higher-than-expect seq. # . Gap detected	Immediately send *duplicate ACK*, indicating seq. # of next expected byte
Arrival of segment that partially or completely fills gap	Immediate send ACK, provided that segment starts at lower end of gap



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Pipeline

- A sender allows multiple, “in-flight”, yet-to-be-acknowledged packets
  - range of sequence numbers must be increased
  - buffering at sender and/or receiver



(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Pipeline

- For example, 1 Gbps link, 8000 bit packet.
  - L: packet size
  - R: transmission rate

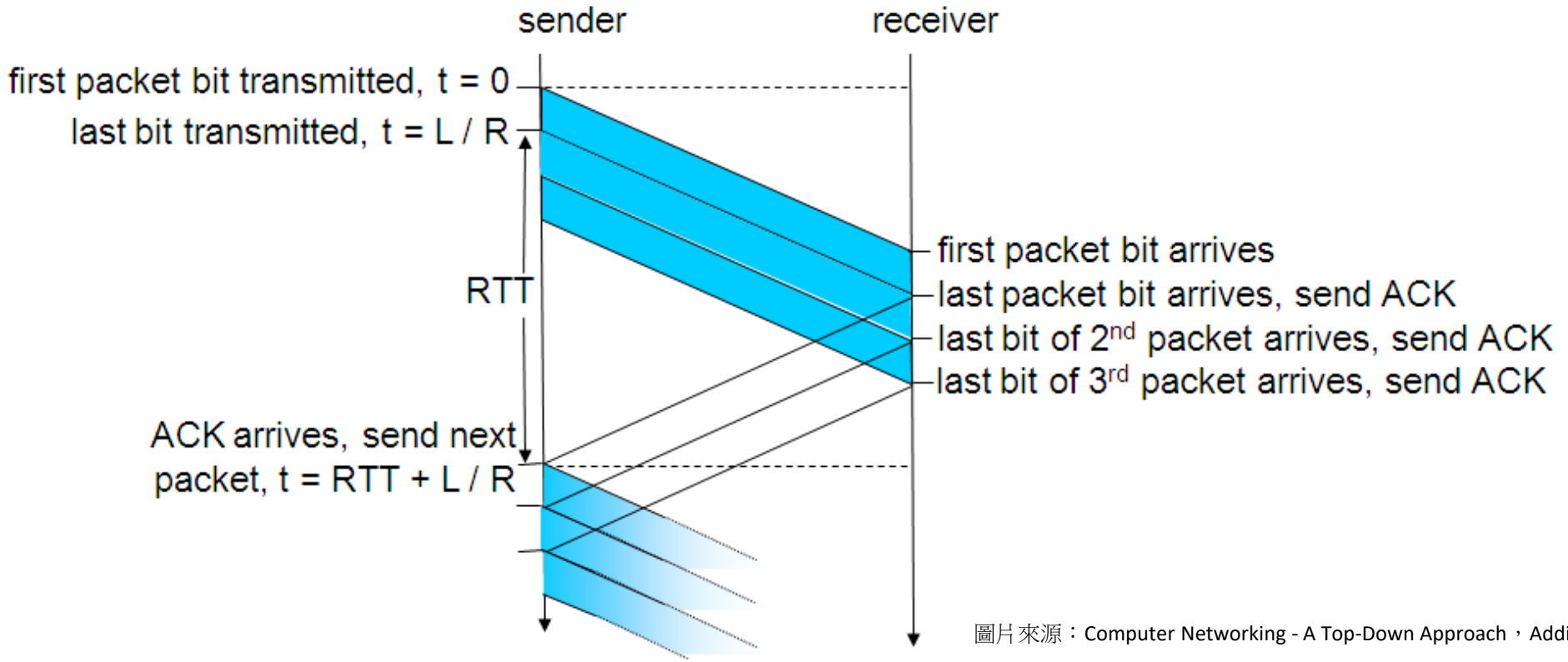
$$d_{trans} = \frac{L}{R} = \frac{8000\text{bits}}{10^9\,\text{bps}} = 8\,\text{microsecon ds}$$

- Utilization of stop-and-wait

$$U_{sender} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

Pipeline

- Increased utilization



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

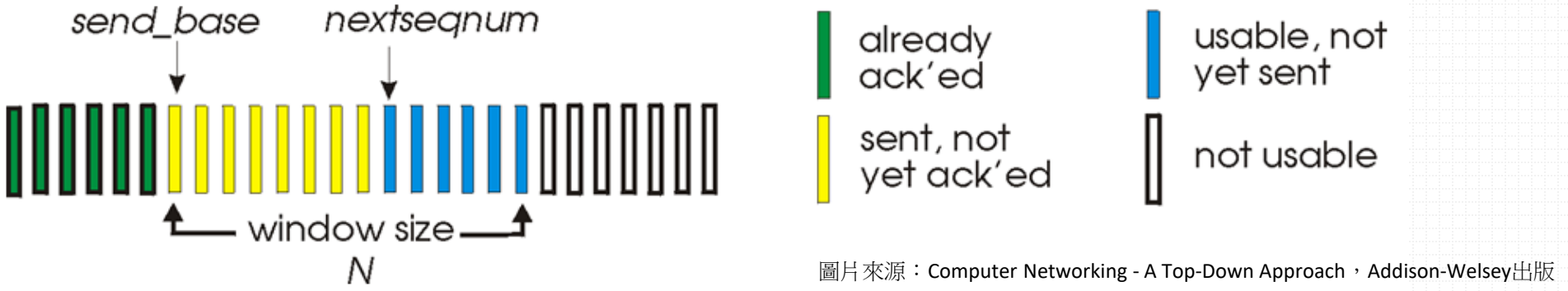
$$U_{sender} = \frac{3 * L / R}{RTT + L / R} = \frac{.024}{30.008} = 0.0008$$

Go-Back-N

- Always send ACK for the correctly-received packet with highest in-order sequence number
  - may generate duplicate ACKs
- Out-of-order packet
  - Discard (don't buffer) → No receiver buffering!
  - Re-ACK the packet with highest in-order sequence number

Go-Back-N

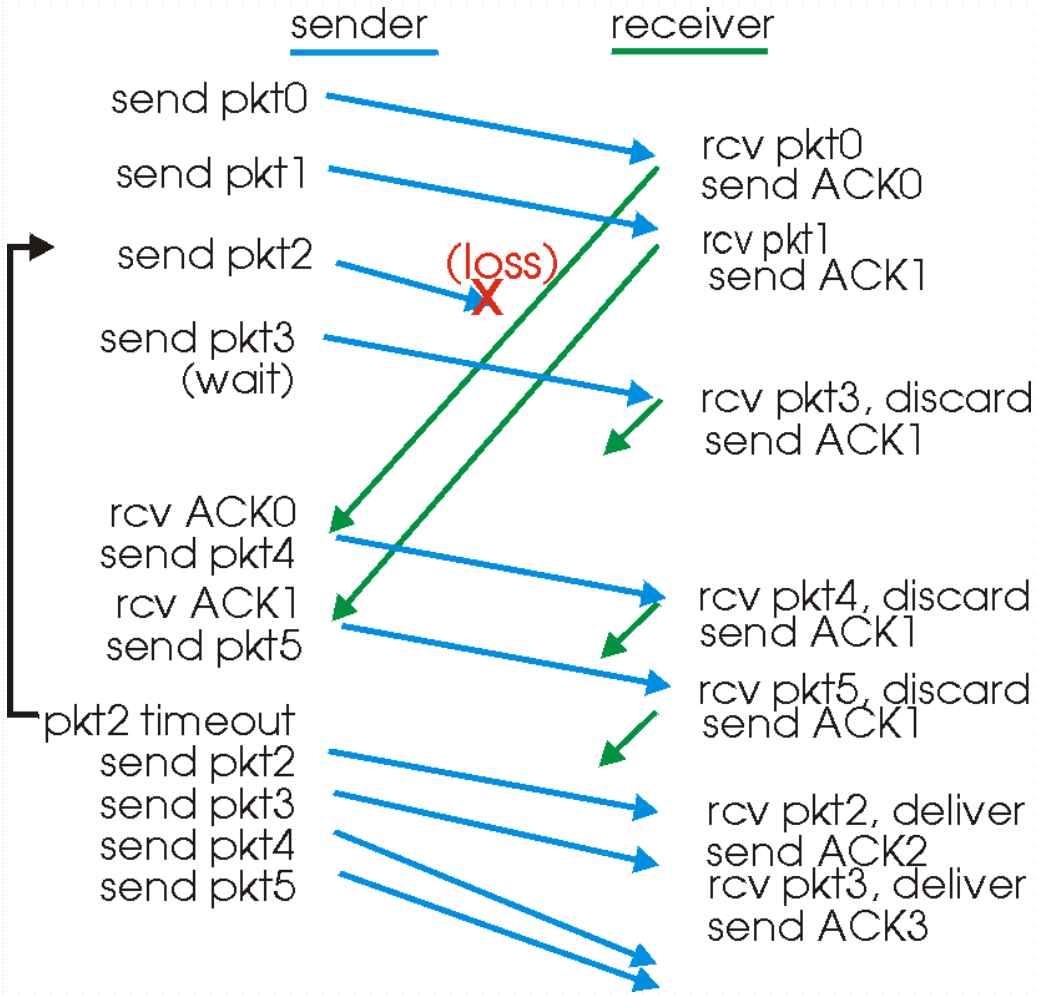
- Sliding-window protocol
  - k-bit sequence number in packet header
  - “window” of up to N, consecutive unacked packets allowed

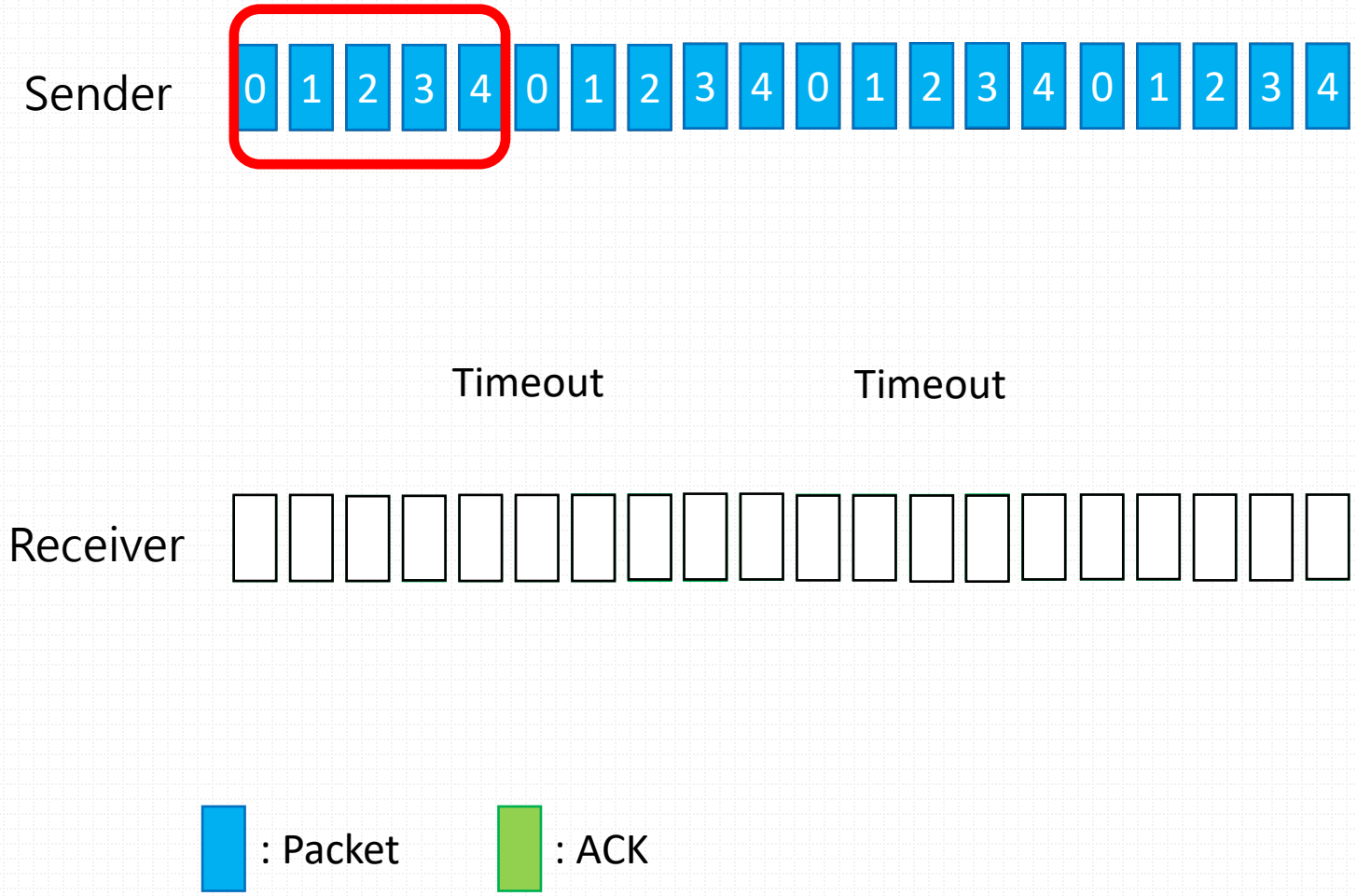


圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

  - Cumulative ACK
  - Timer for each in-flight packet

Go-Back-N



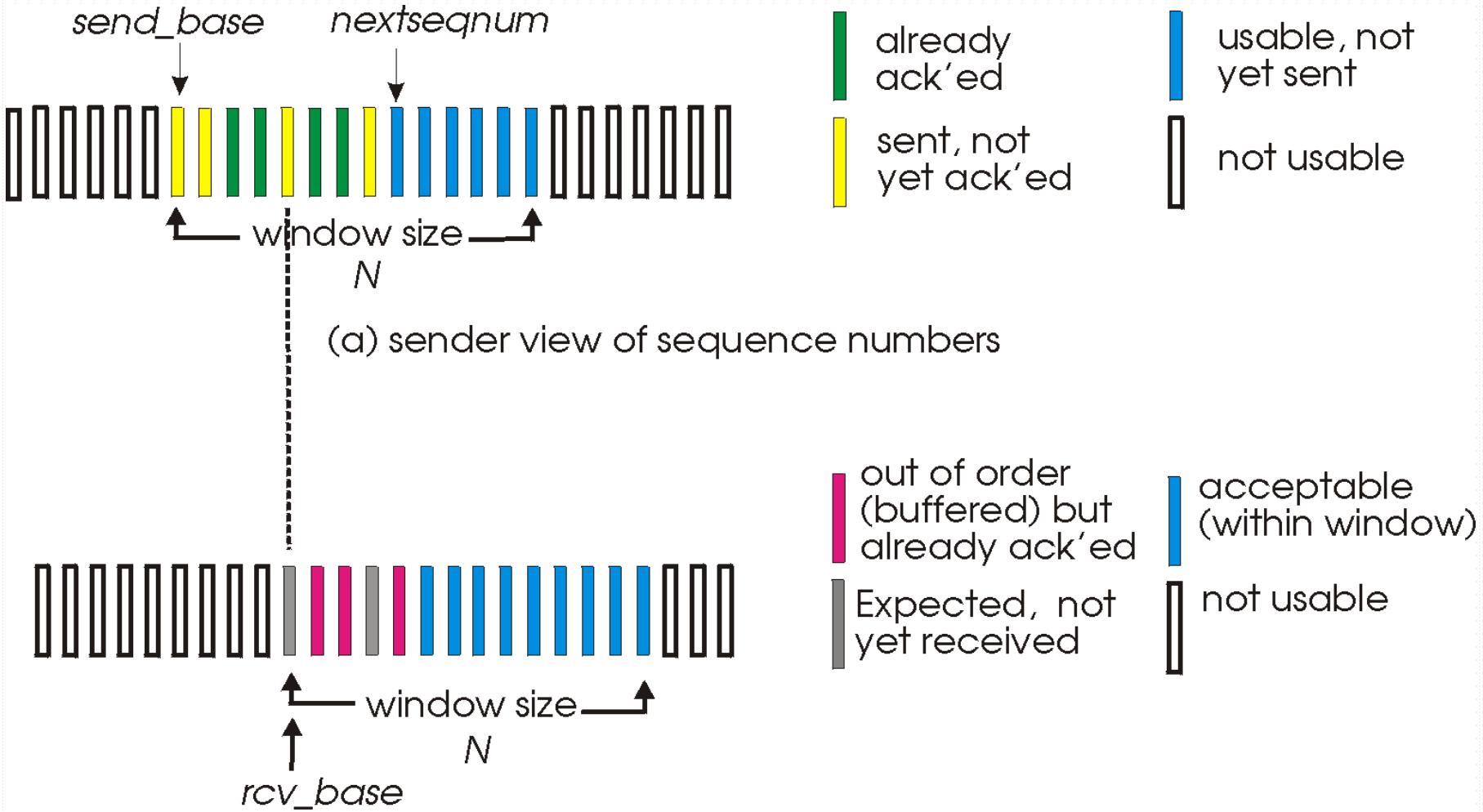


圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Selective Repeat

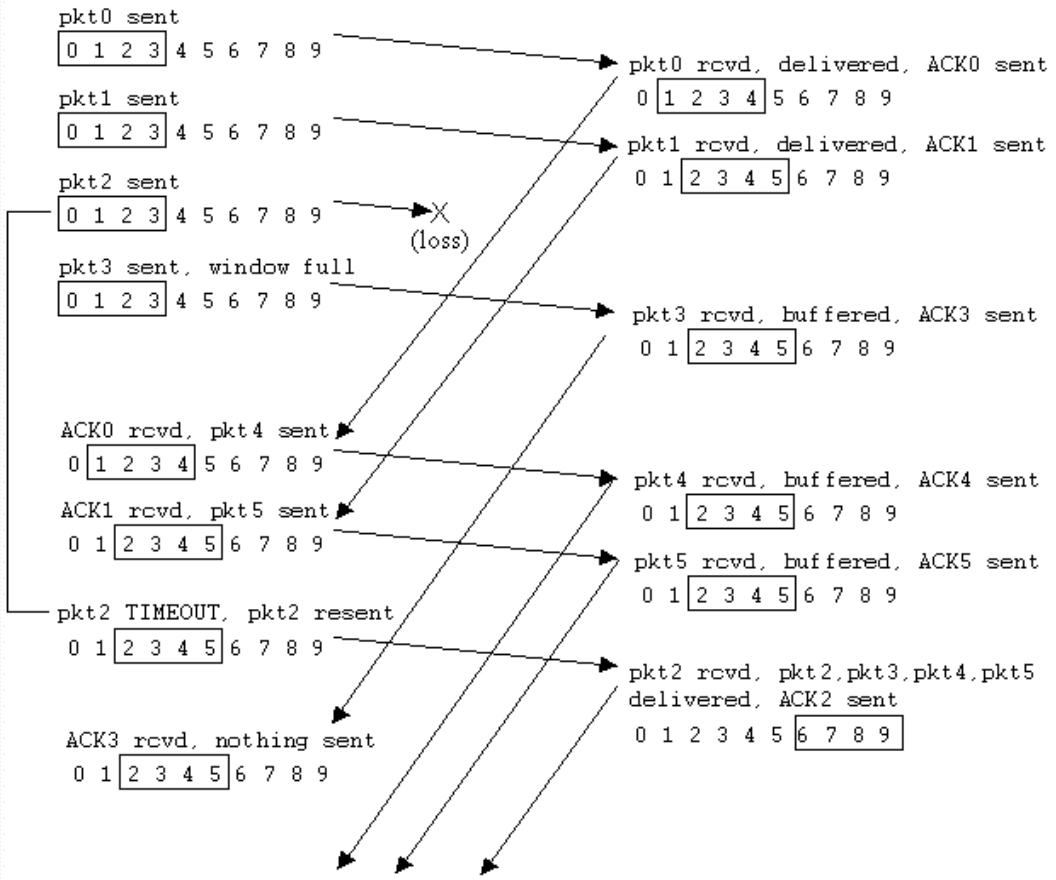
- Sender window
  - N consecutive sequence number
  - again limits sequence numbers of sent, unacked packets
  - Sender only re-sends packets for which ACK is not received
    - Sender timer for each unacked packet
- Receiver individually acknowledges all correctly received packets
  - buffers packets, as needed, for eventual in-order delivery to upper layer

Selective Repeat

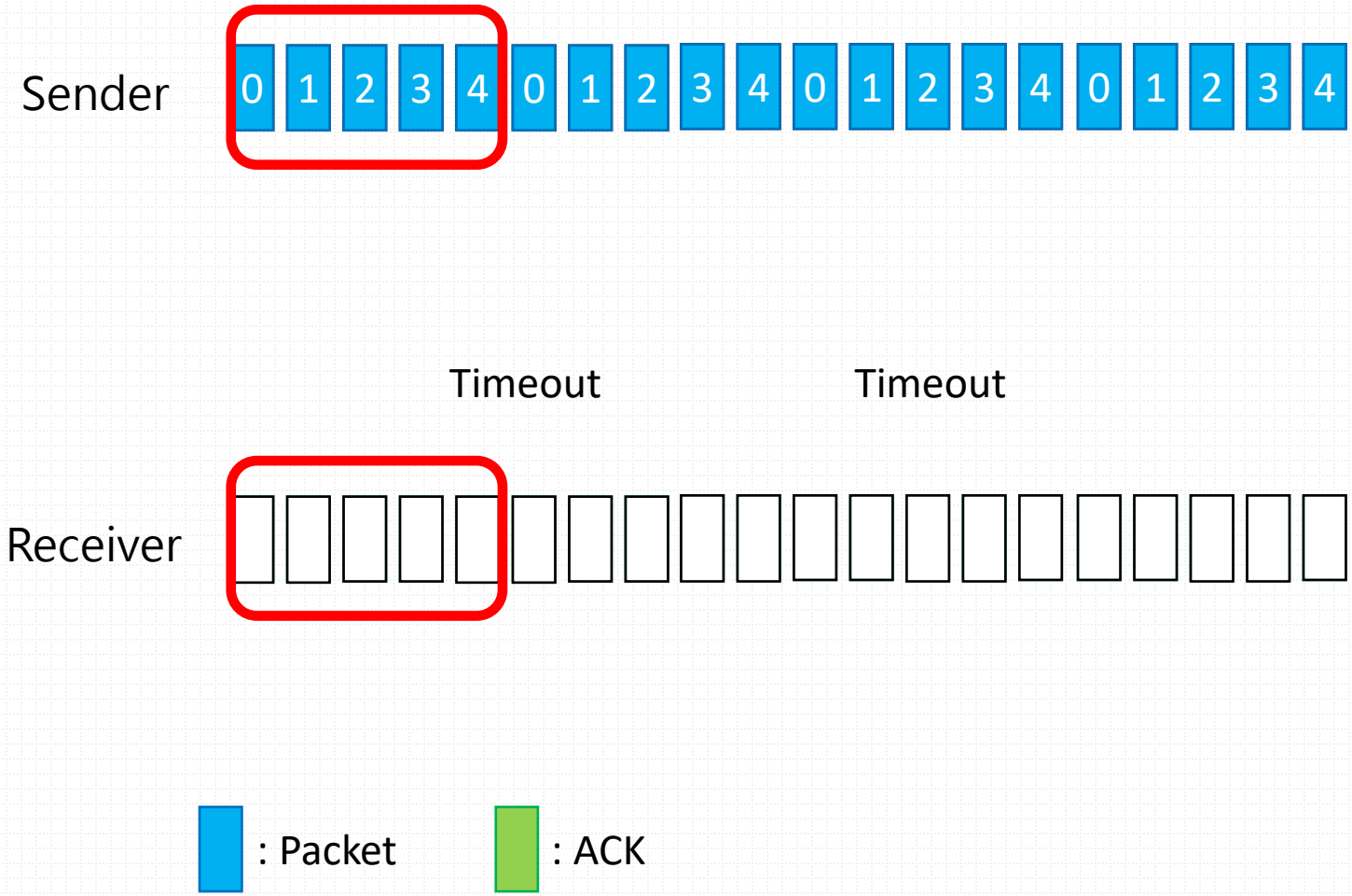


圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Selective Repeat

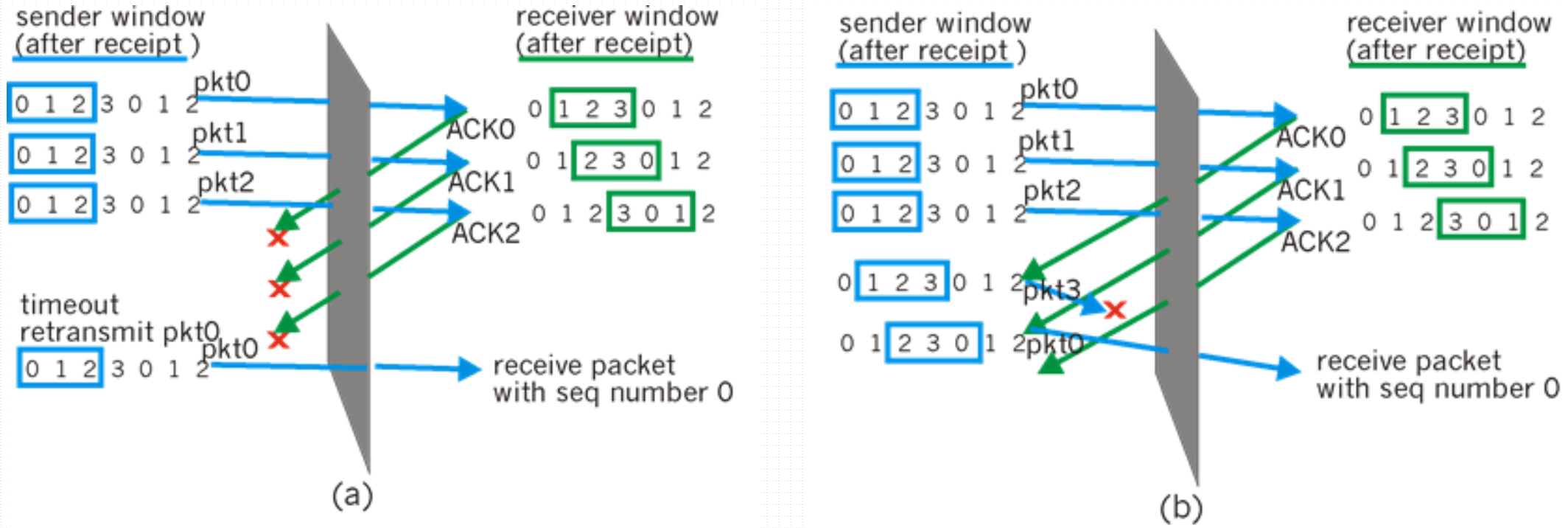


圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版



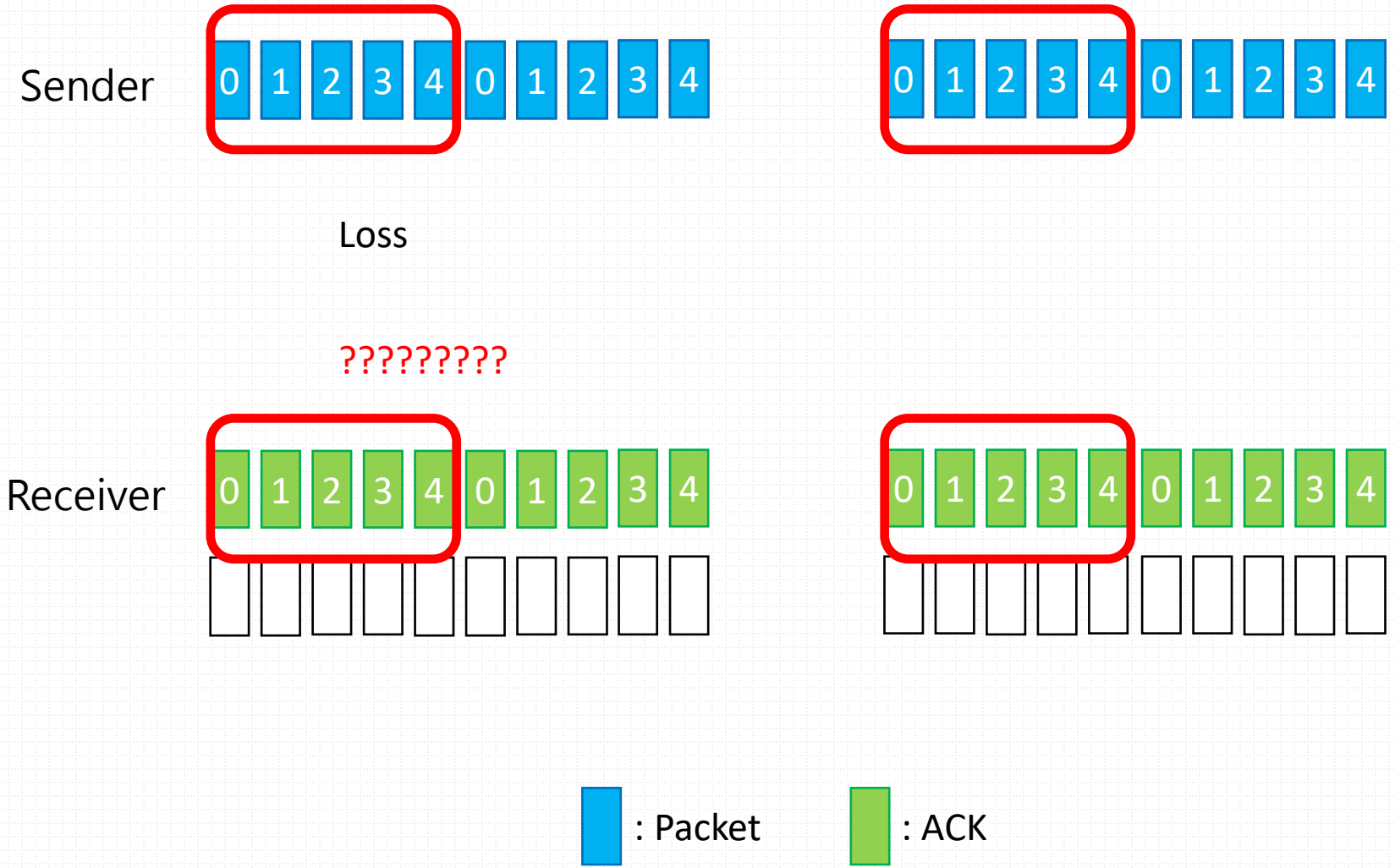
Dilemma of Selective Repeat

- Dilemma
  - Receiver sees no difference in two scenarios.
    - What relationship between sequence number size and window size?



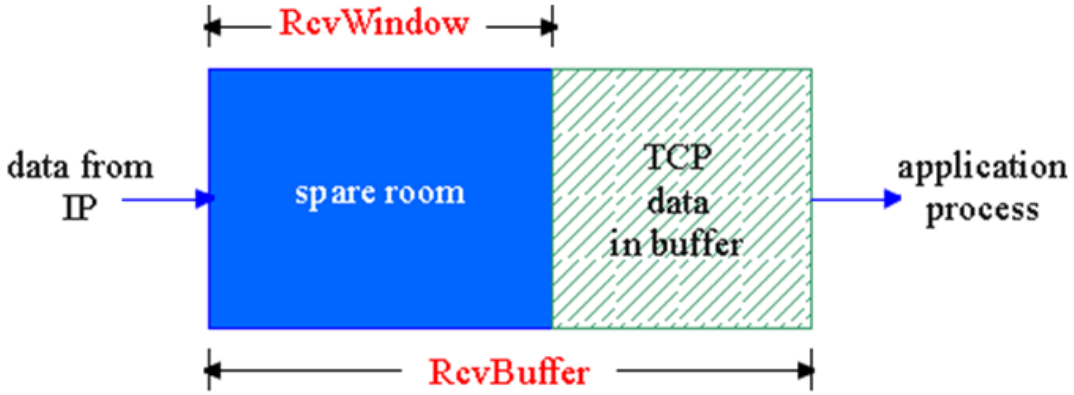
圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Example：Dilemma of Selective Repeat



Flow Control

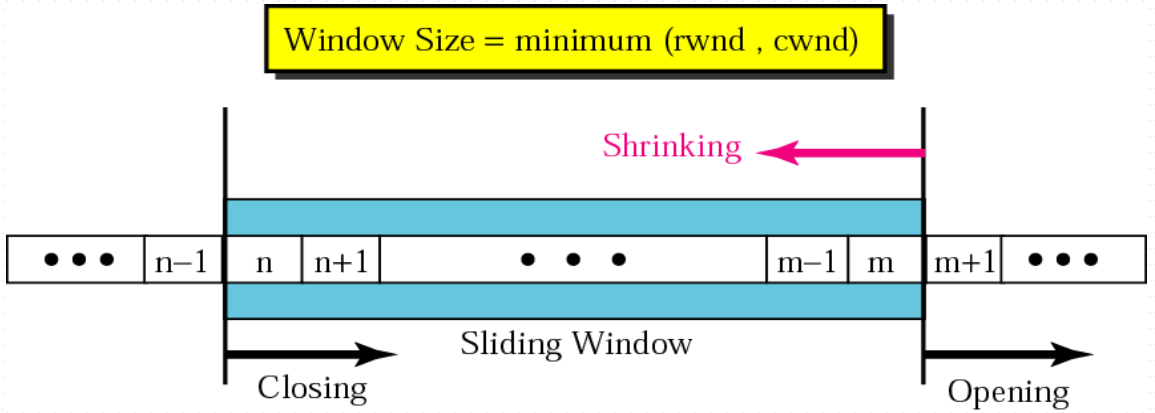
- Speed-matching service
  - Sender won't overflow receiver's buffer by transmitting too much or too fast.
- Receive Window (RcvWindow)
  - Give the sender an idea of how much free buffer space is available at the receiver.

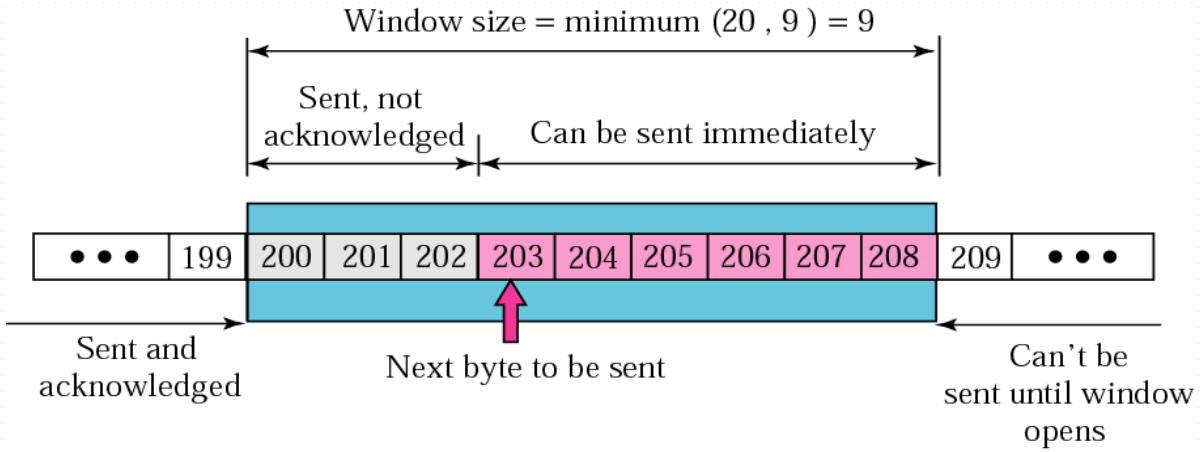


圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Flow Control

- Sender limits unacked data to RcvWindow
  - It guarantees receive buffer doesn't overflow.





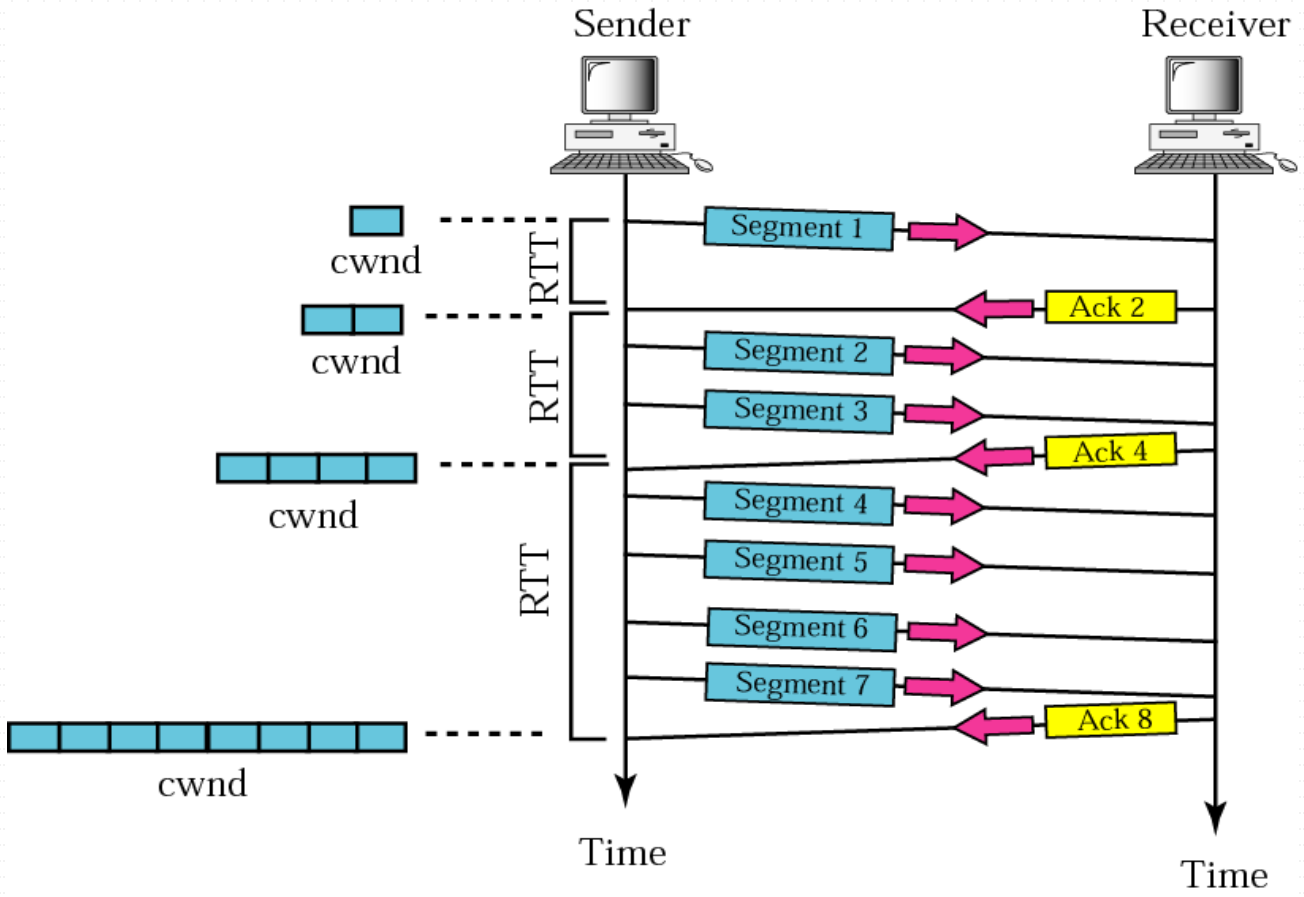
圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Congestion Control

- Congestion window (CongWin)
  - The rate at which a TCP sender can send traffic into the network
- How does a sender perceive congestion?
  - Loss event = timeout or 3 duplicate acks.
  - TCP sender reduces rate (CongWin) after loss event.
- Three stages for TCP congestion control
  - Slow start (SS)
  - Congestion Avoidance (CA)
  - Fast Recovery

Slow Start (SS)

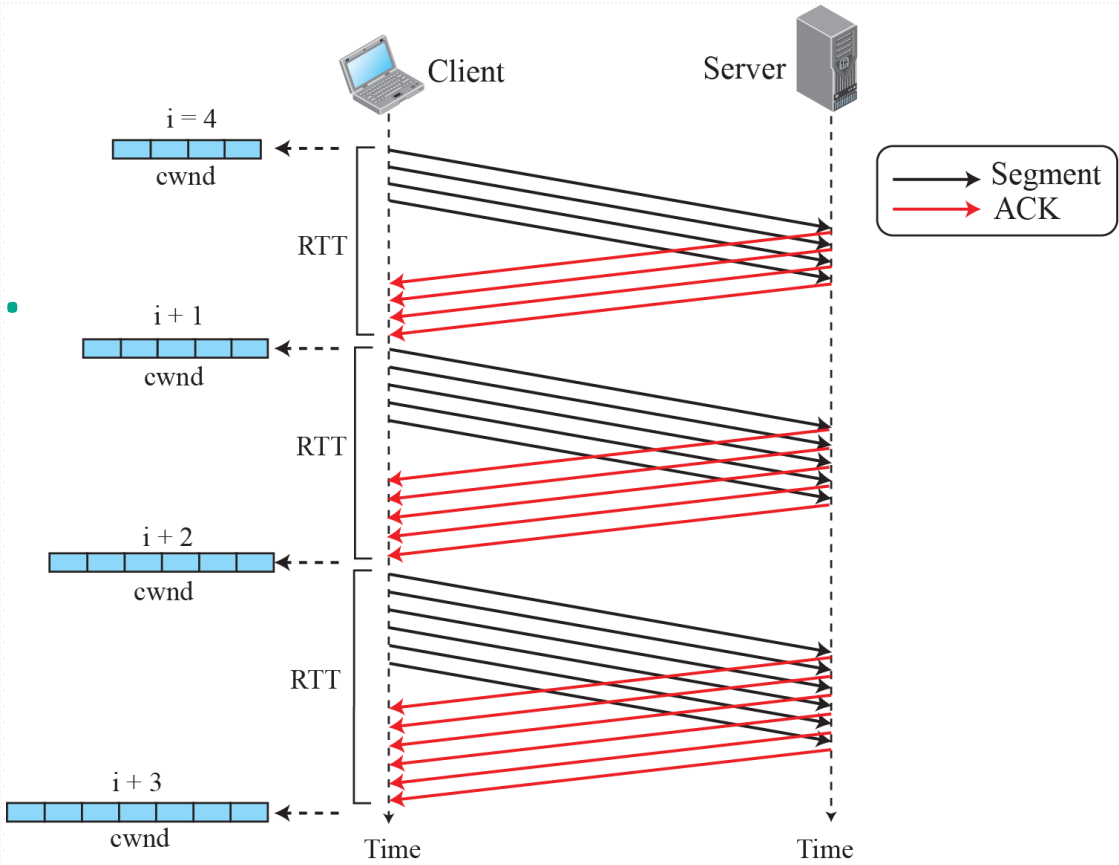
- When one connection begins, increase rate exponentially until the first loss event.
  - CongWin = 1 MSS
  - Double CongWin every RTT
  - If a loss event happens,
    - CongWin = 1 MSS
    - Slow Start Threshold (ssthresh) = CongWin / 2
    - Restart SS



圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Congestion Avoidance (CA)

- When CongWin ≧ ssthresh, window grows linearly, i.e., increase CongWin by 1 MSS every RTT.
- If timeout, re-start SS (just like SS).
  - CongWin = 1 MSS
  - ssthresh = CongWin / 2
- If 3 duplicate ACKs were received, enter fast recovery.

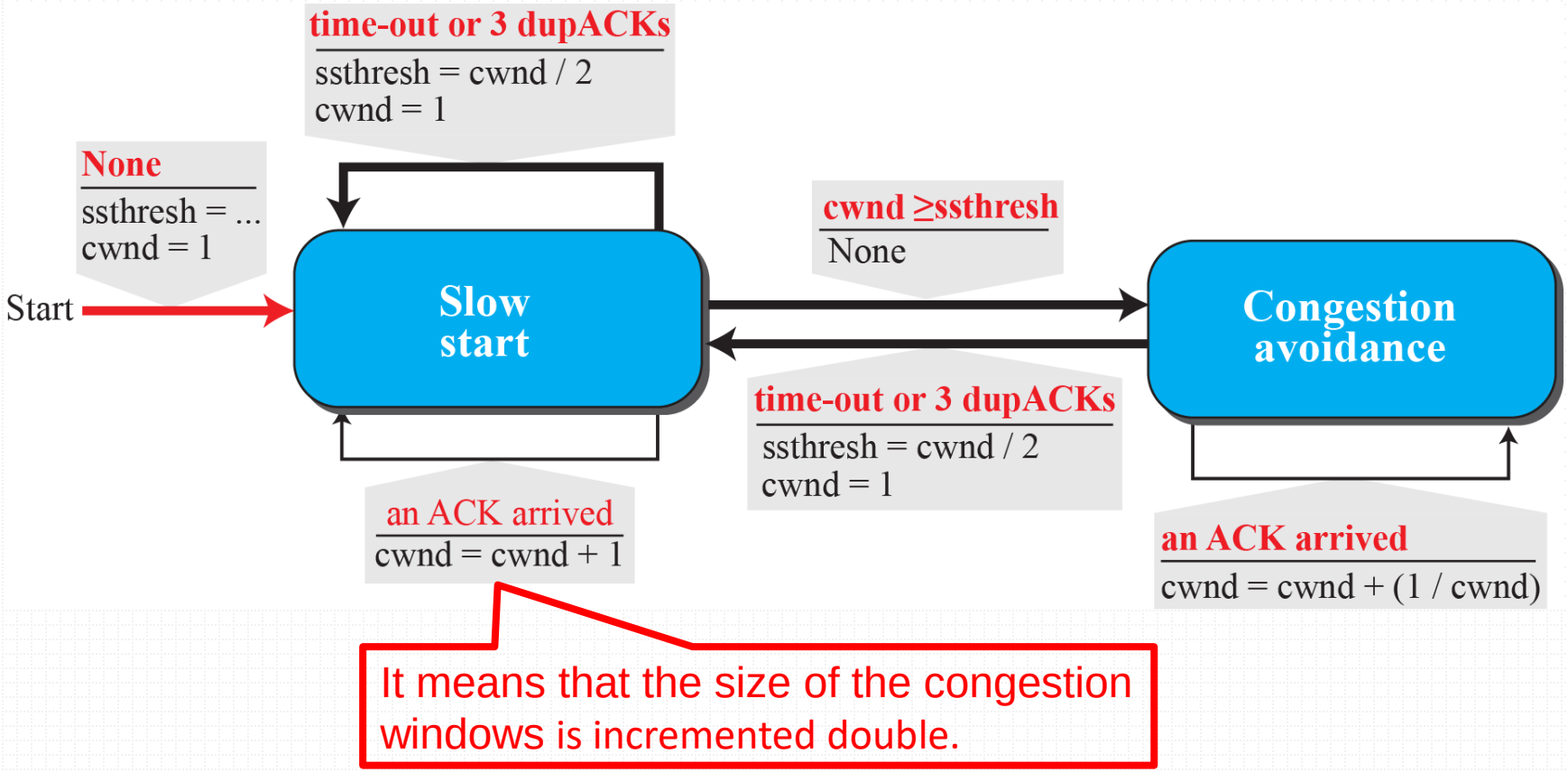


圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Fast Recovery

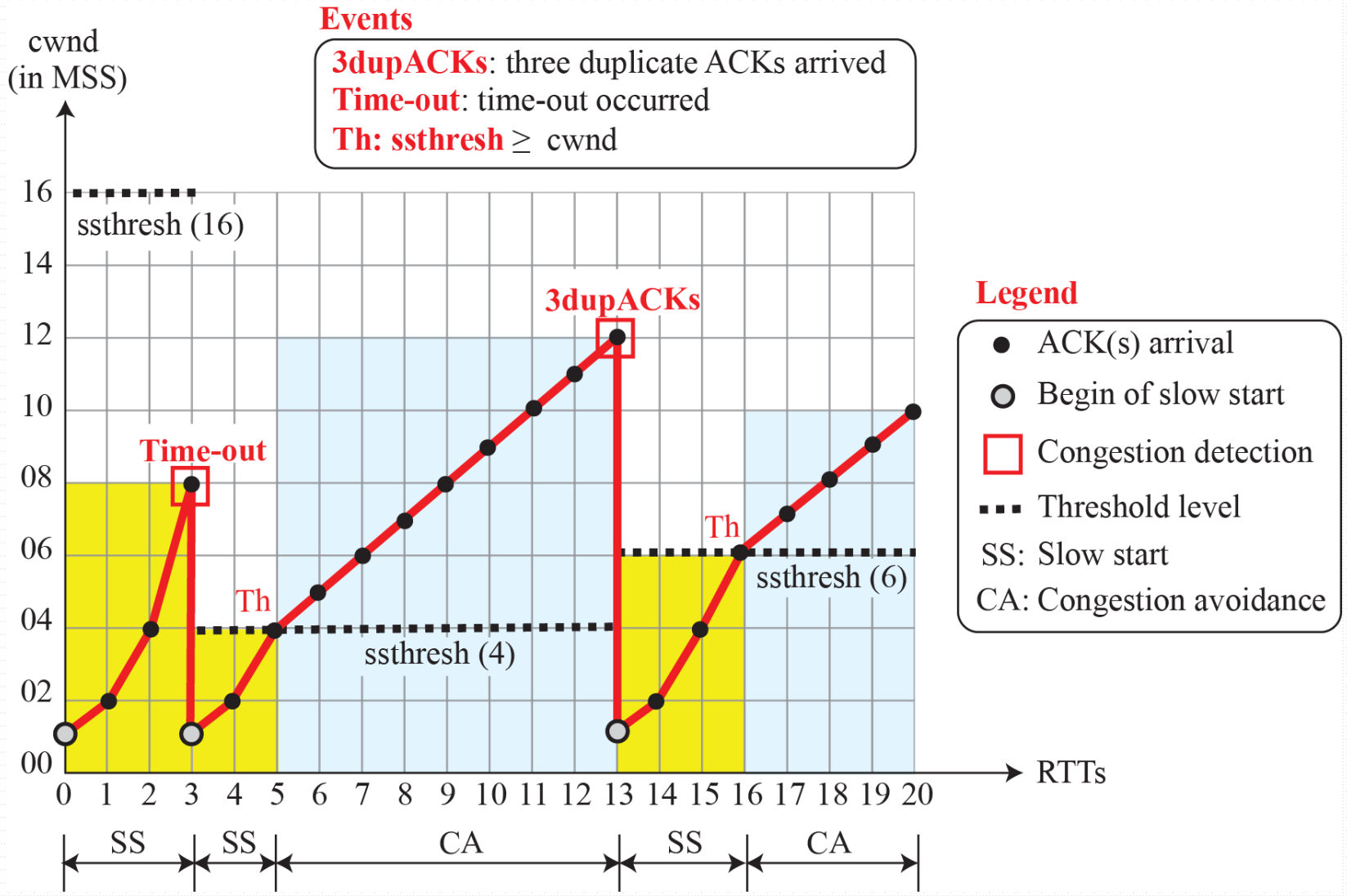
- TCP Tahoe
  - Unconditionally cuts its congestion window to 1 MSS and enters the slow-start phase after either type of loss event.
- TCP Reno
  - Cancel the slow-start phase after a triple duplicate ACK.
    - Three duplicate ACKs indicates that some segments have been received at the sender.
  - Fast recovery

TCP Tahoe



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

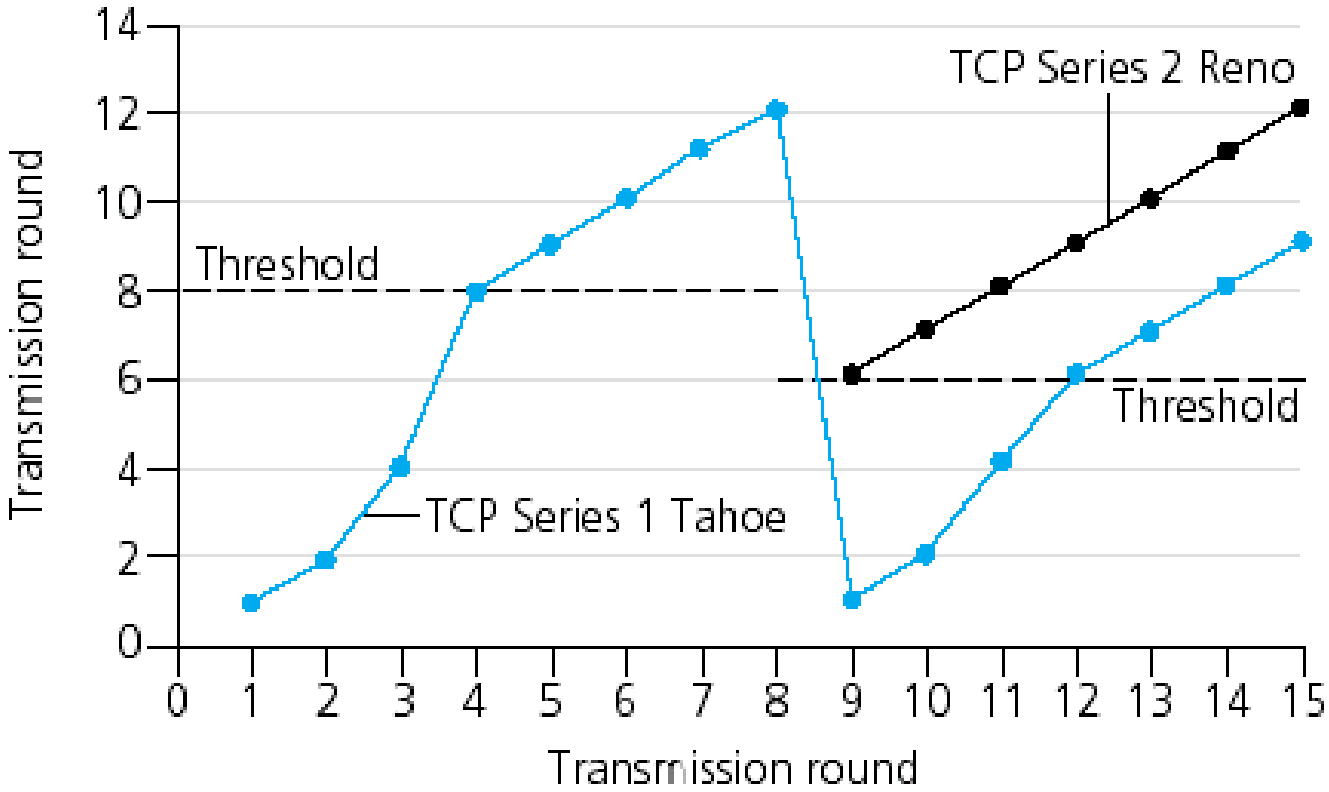
TCP Tahoe



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

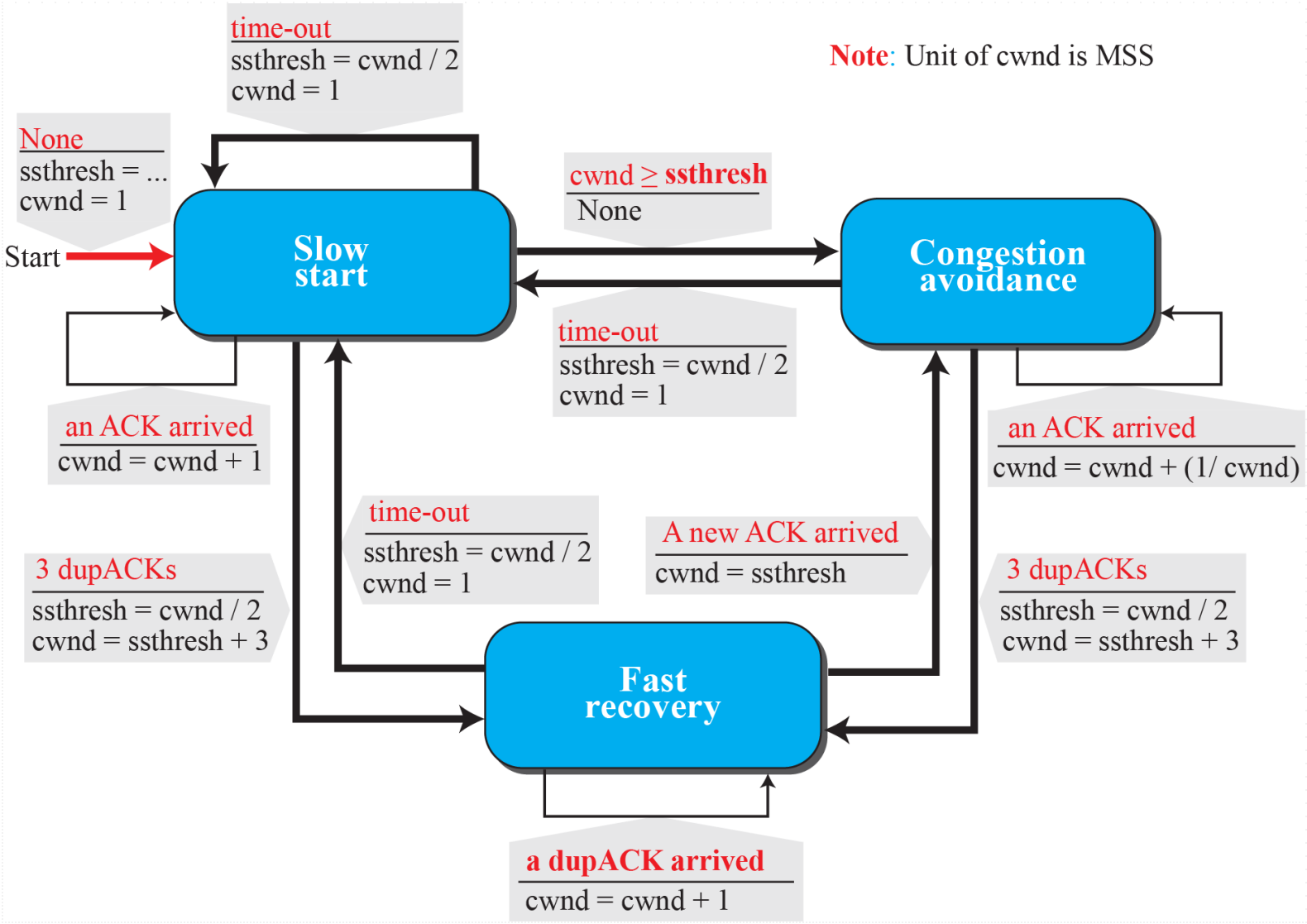
Fast Recovery

- Increase CongWin by 1 MSS every RTT.
- ssthresh = CongWin / 2
- CongWin = ssthresh
- If timeout, re-start SS (just like SS).



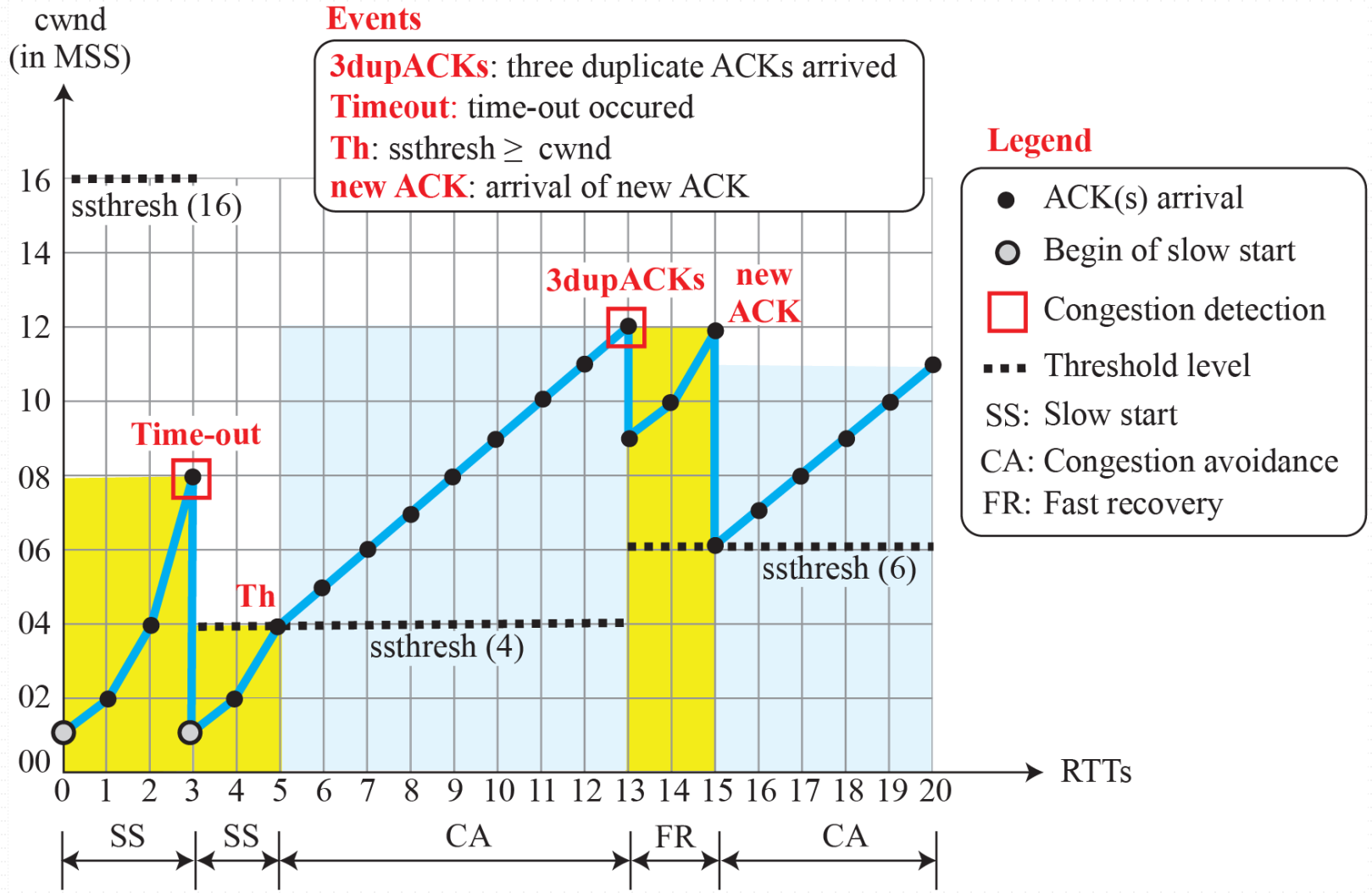
圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

TCP Reno



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

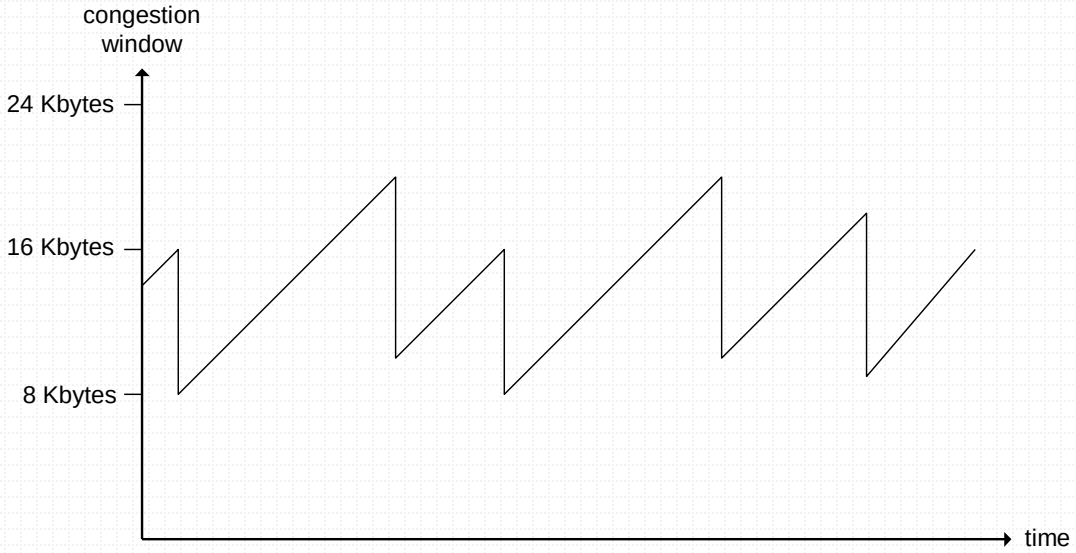
TCP Reno



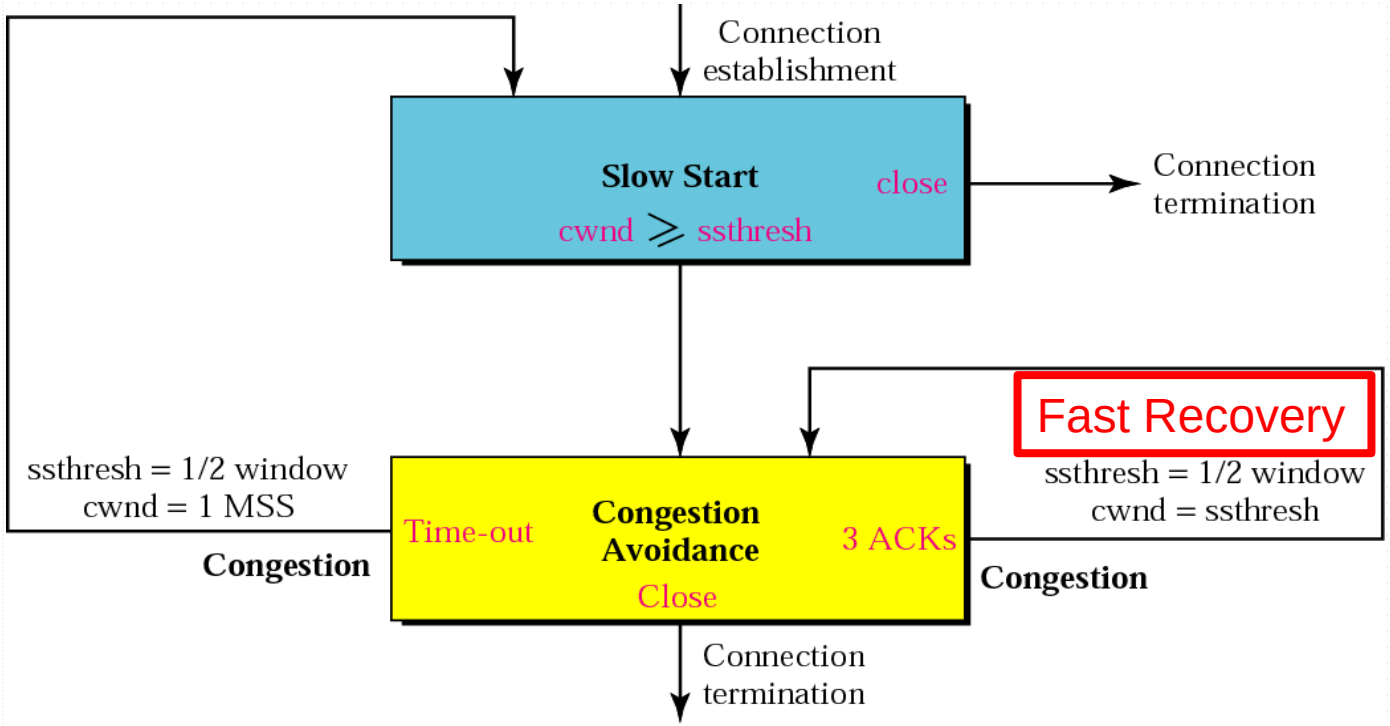
圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

Congestion Control

- Additive-Increase, Multiplicative-Decrease (AIMD)
  - probing for usable bandwidth, until loss occurs



圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版



圖片來源：TCP/IP Protocol Suite，McGraw-Hill出版

Congestion Control

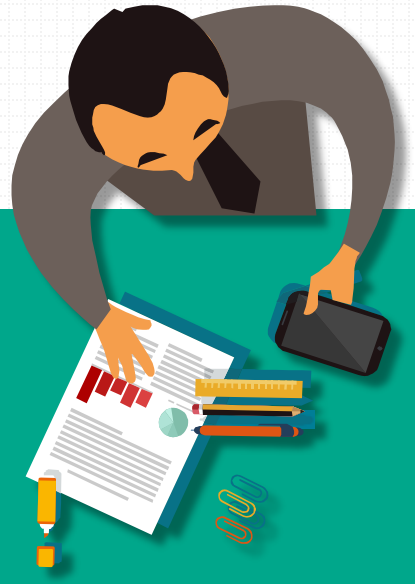
State	Event	TCP Sender Action	Commentary
Slow Start (SS)	ACK receipt for previously unacked data	CongWin = CongWin + MSS, If (CongWin > Threshold) set state to "Congestion Avoidance"	Resulting in a doubling of CongWin every RTT
Congestion Avoidance (CA)	ACK receipt for previously unacked data	CongWin = CongWin+MSS * (MSS/CongWin)	Additive increase, resulting in increase of CongWin by 1 MSS every RTT
SS or CA	Loss event detected by triple duplicate ACK	Threshold = CongWin/2, CongWin = Threshold, Set state to "Congestion Avoidance"	Fast recovery, implementing multiplicative decrease. CongWin will not drop below 1 MSS.
SS or CA	Timeout	Threshold = CongWin/2, CongWin = 1 MSS, Set state to "Slow Start"	Enter slow start
SS or CA	Duplicate ACK	Increment duplicate ACK count for segment being acked	CongWin and Threshold not changed

圖片來源：Computer Networking - A Top-Down Approach，Addison-Welsey出版

NTUT NESL

總結

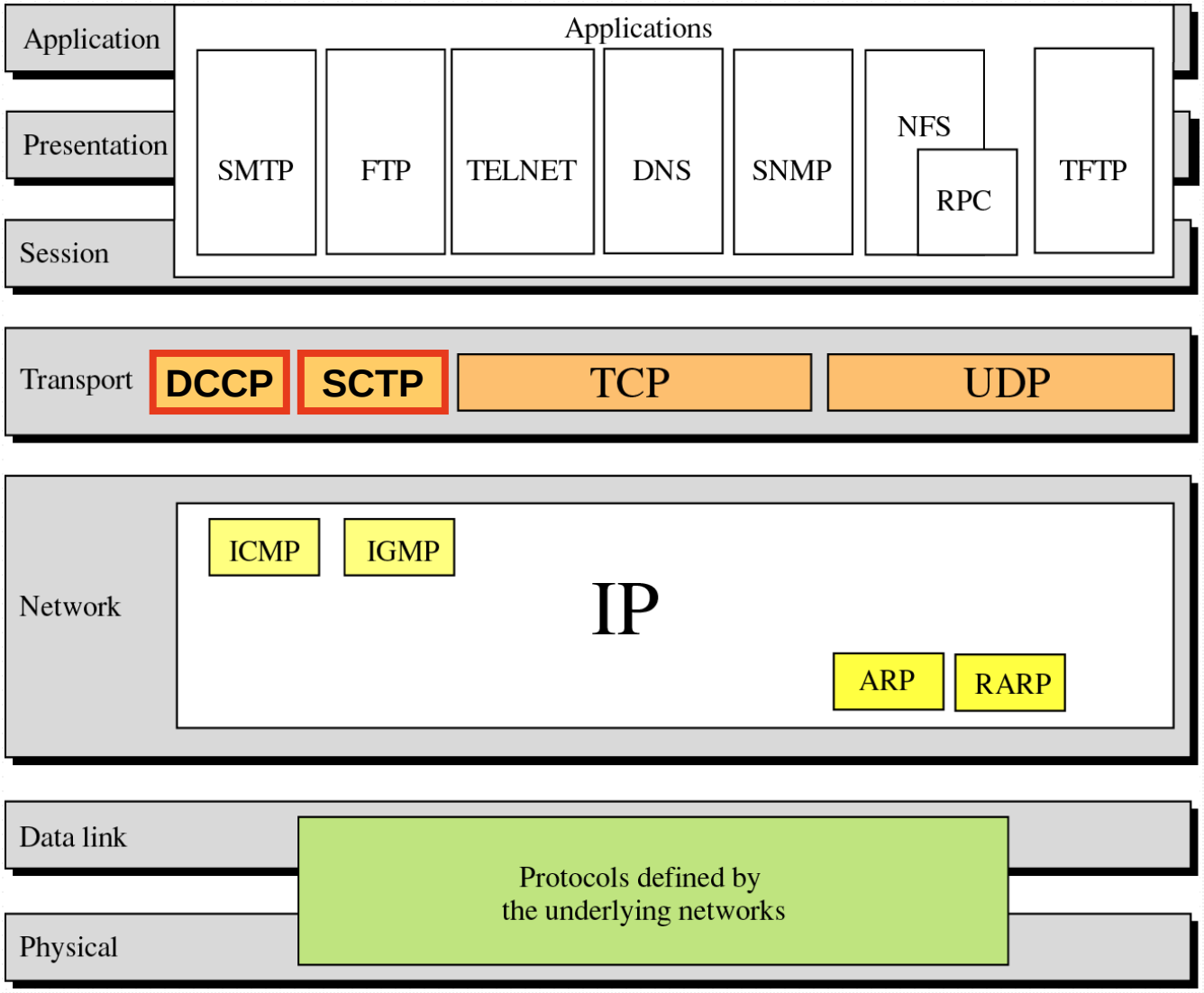
- The key characteristics of TCP
  - Reliable
    - Checksum, ARQ, Sequence Number, Countdown Timer, Pipeline
  - Connection Management
  - Flow Control
  - Congestion Control
    - Slow Start (SS), Congestion Avoidance (CA) and Fast Recovery



學習目標 Outline

- Datagram Congestion Control Protocol (DCCP)
- Stream Control Transmission Protocol (SCTP)

New Transport-layer Protocols



Datagram Congestion Control Protocol (DCCP)

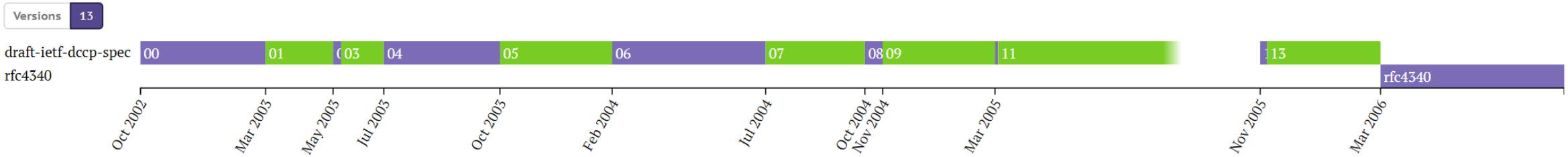
NTUT NESL

IETF dccp / WG

- Datagram Congestion Control Protocol (dccp) **Concluded**
  - This WG is maintaining the Datagram Congestion Control Protocol (DCCP).
  - https://datatracker.ietf.org/wg/dccp/about/

Datagram Congestion Control Protocol (DCCP)

- RFC 4340



  - Updated by RFC 5595, RFC 5596, RFC 6773, RFC 6335

Datagram Congestion Control Protocol (DCCP)

- TCP- Transport Control Protocol
  - Connection Oriented-provide reliable data transfer, flow control, congestion control
  - Guarantee data delivery
  - Delay tolerance
  - Files transfer, Web browsing, email apps
- UDP- User Datagram Protocol
  - Connectionless
  - Guarantees nothing
  - No Congestion Control
  - DNS, SNMP, streaming media, telephony, on-line games

Datagram Congestion Control Protocol (DCCP)

- What streaming media needs?
  - Timeliness of data
- What streaming media doesn't need?
  - Retransmissions of lost/expired packets
  - Annoying "rebuffering…" – HOL blocking
- For example,
  - Internet Telephony
    - Constant-packet-rate sources
    - Change data rate by adjusting packet size
    - Extremely sensitive to delay
    - Demands a slower congestion response
  - Interactive games
    - Can quickly make use of available bandwidth
    - Prefers TCP-like sawtooth congestion response

Datagram Congestion Control Protocol (DCCP)

DCCP = UDP + congestion control

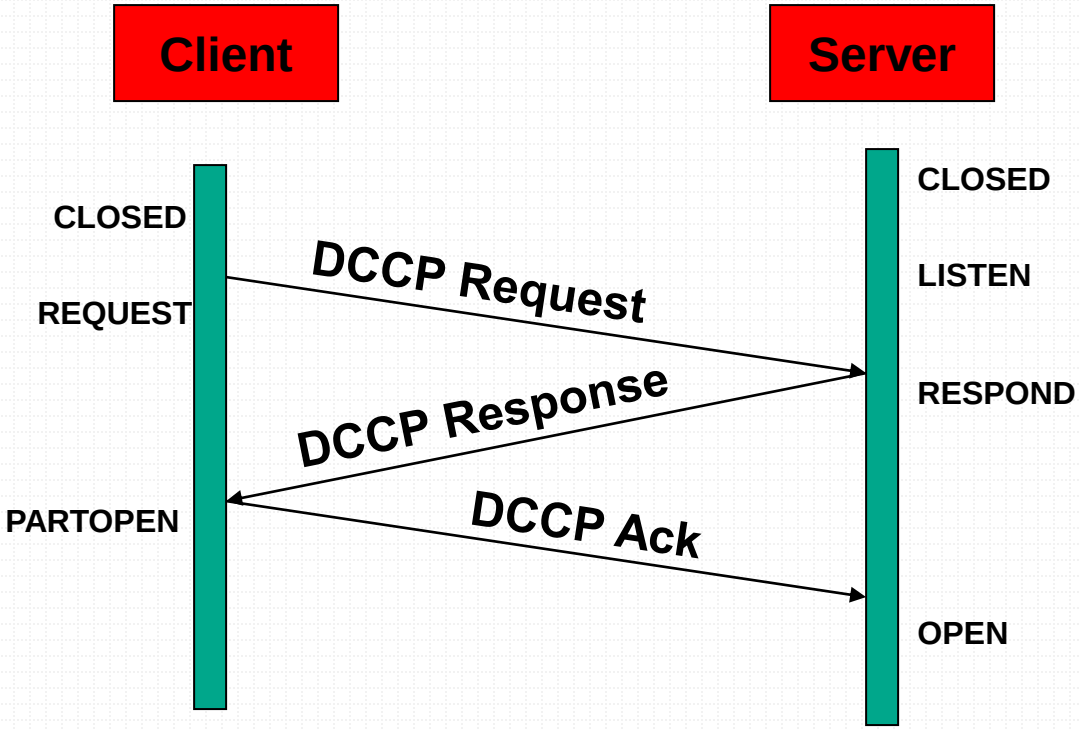
or

DCCP = TCP – bytestream semantics – full reliability

- DCCP provides unreliable flow of datagrams
- DCCP provides congestion control using
  - Acknowledgment
  - Sequence number
  - Connection oriented
- DCCP does not provide
  - Full reliability: no-loss & no-error & in-order & no-duplicate
  - flow control
  - streaming

Datagram Congestion Control Protocol (DCCP)

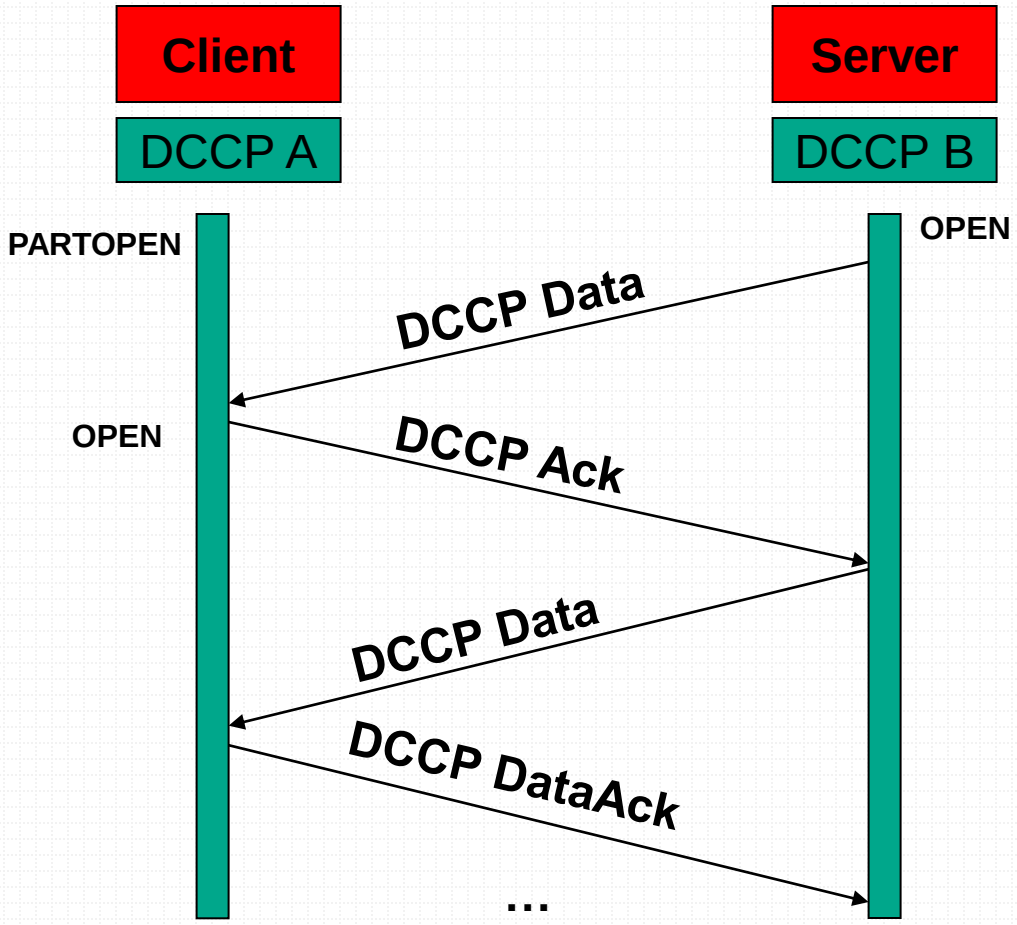
- Full-duplex bi-directional connection
  - Two logical half connections
  - A-to-B half connection:
    - Application data sent from A to B
    - Corresponding acks from B to A
  - In practice overlapped: DataAck
- Each half connection can have independent features negotiated during connection initiation, e.g., different congestion control mechanism



DCCP uses a three-way handshake to establish a connection.

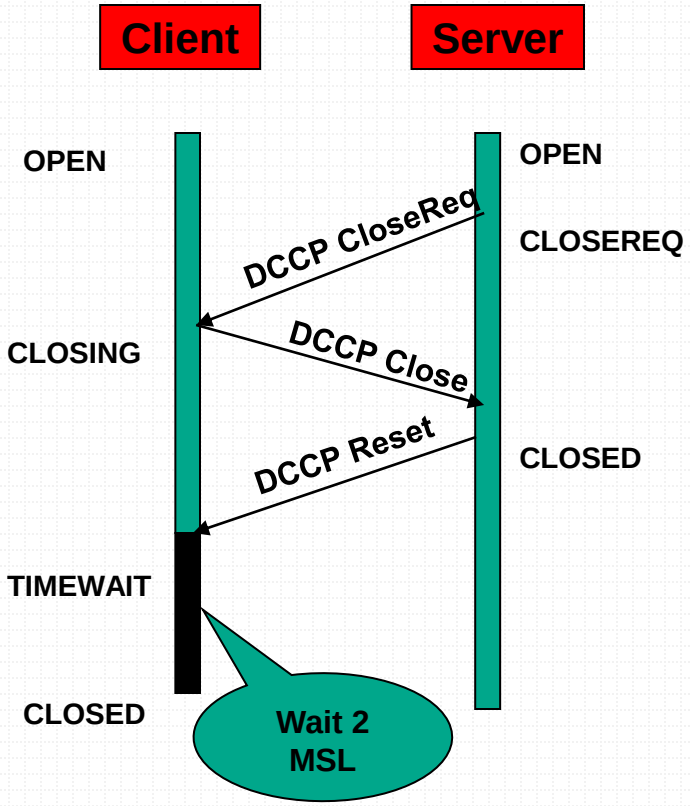
Datagram Congestion Control Protocol (DCCP)

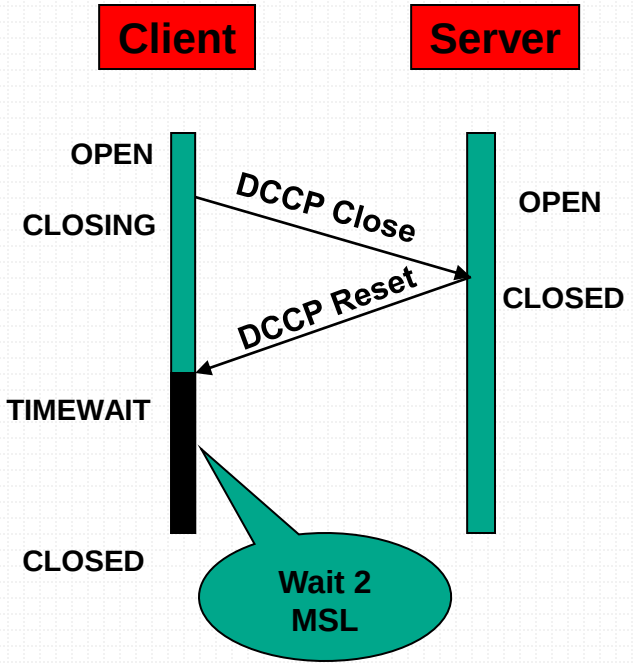
- The data packet exchange between the client and server is unreliable but restricted to congestion control.
- During the transfer process, DCCP endpoints might use DCCP-Sync and DCCP-SyncAck packets for synchronization.



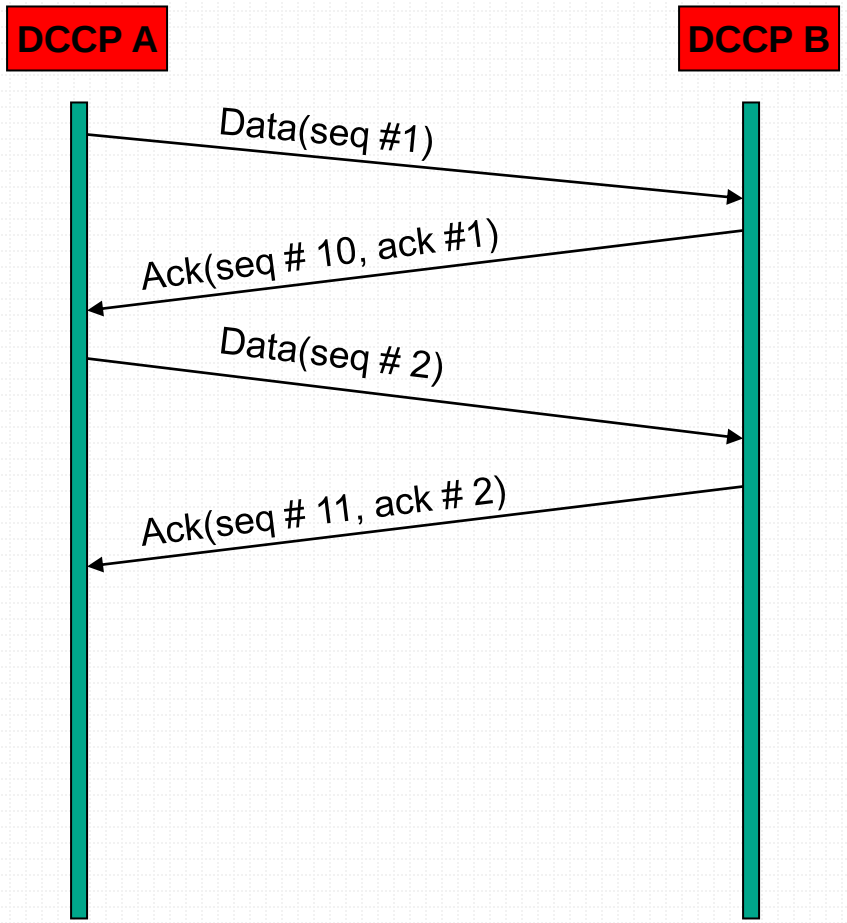
Datagram Congestion Control Protocol (DCCP)

- DCCP uses two- or three-way handshakes to terminate a connection.



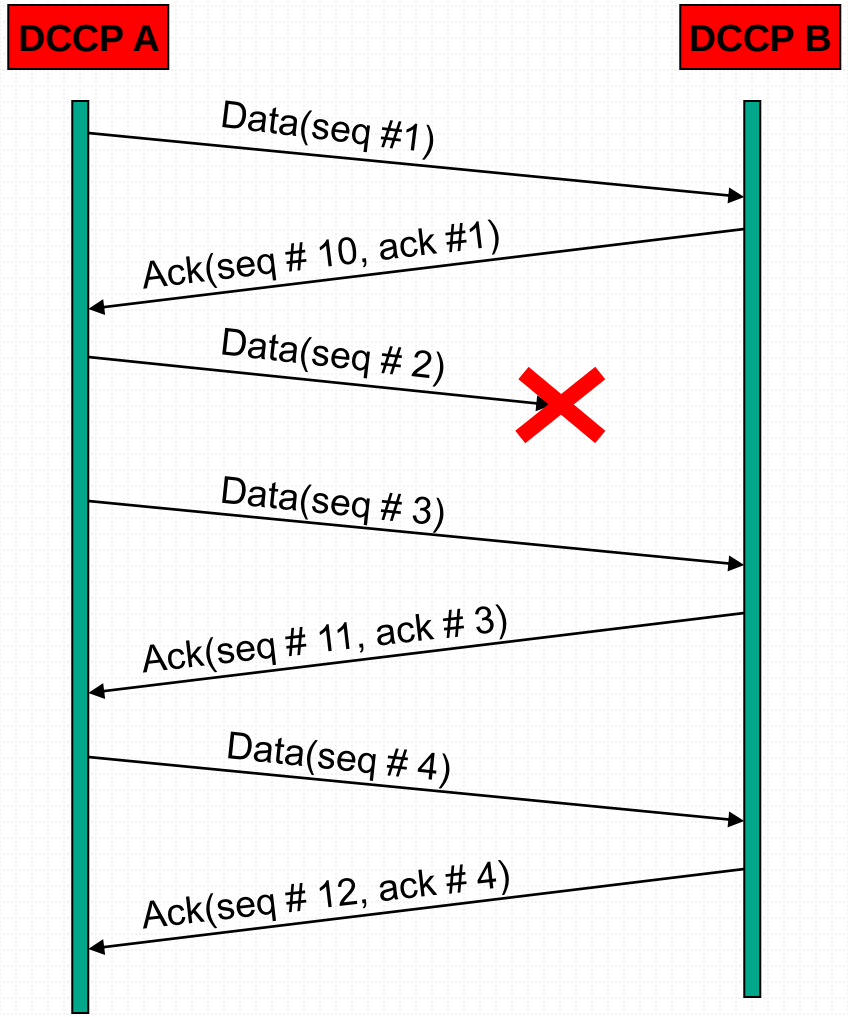


Datagram Congestion Control Protocol (DCCP)



- Example：without loss
  - Seq # on DCCP-PDU, not byte
  - Each PDU carries a Seq #
  - Seq # increases per PDU
    - detect loss – congestion control
    - network duplicate – ignored
  - Pure acks also consume Seq #
    - possible to detect ack loss
  - No cum ack – ack # is the Greatest Seq # Received (GSR) normally

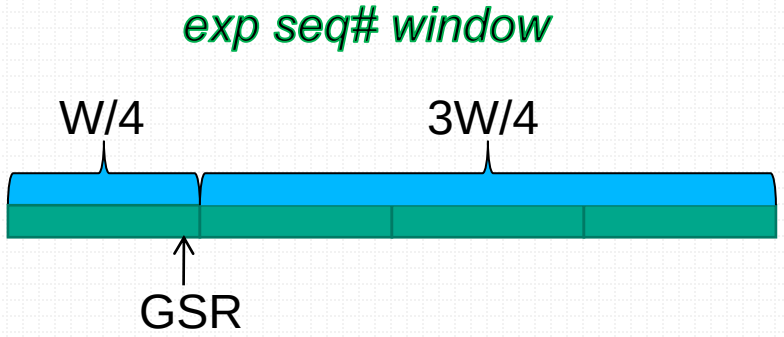
Datagram Congestion Control Protocol (DCCP)

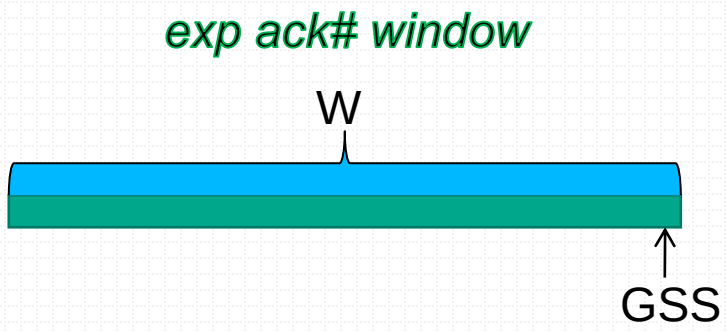


- Example：non-large burst of loss
  - Maybe loss: no data retransmissions
  - Separate options indicate PDU loss or ECN info: Ack Vector (SACK-like)
  - Ack of Ack: clear receiver's state
  - A PDU is ackable – its header has been successfully processed (e.g., valid header checksum and seq #)
  - Acked PDUs may be dropped -- no guarantee of data delivery
    - Due to receiver buffer overflow or corruption -- endpoint loss
    - Data dropped option

Datagram Congestion Control Protocol (DCCP)

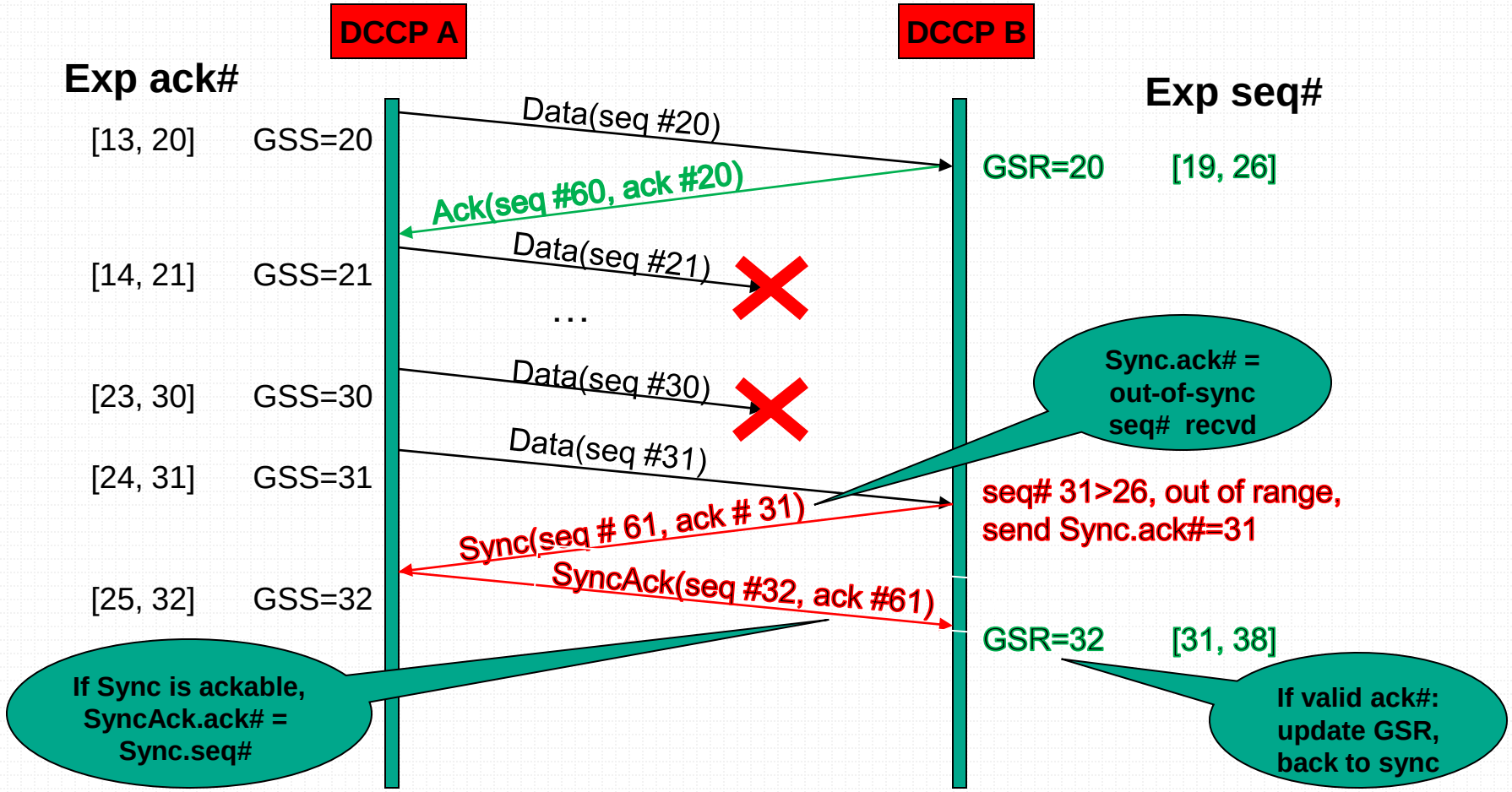
- Both endpoints keep expected seq#/ack# -- in sync
  - To detect seq# attacks, significant reordering, or one endpoint crash
  - Out of sync after large burst of loss
  - No cum ack, use separate DCCP-Sync/SyncAck PDU to recover
- Sequence number variables
  - Maintained at each endpoint for each connection
  - GSR – Greatest Seq# Received
  - GSS – Greatest Seq# Sent
- Sequence validity windows
  - Window width W: Seq Win feature
  - Expected seq# [SWL, SWH], SWH=GSR+3W/4
  - Expected ack# [AWL, AWH], AWH=GSS
  - Seq#/ack# out of range – seq invalid PDU, ignore and send Sync PDU





Datagram Congestion Control Protocol (DCCP)

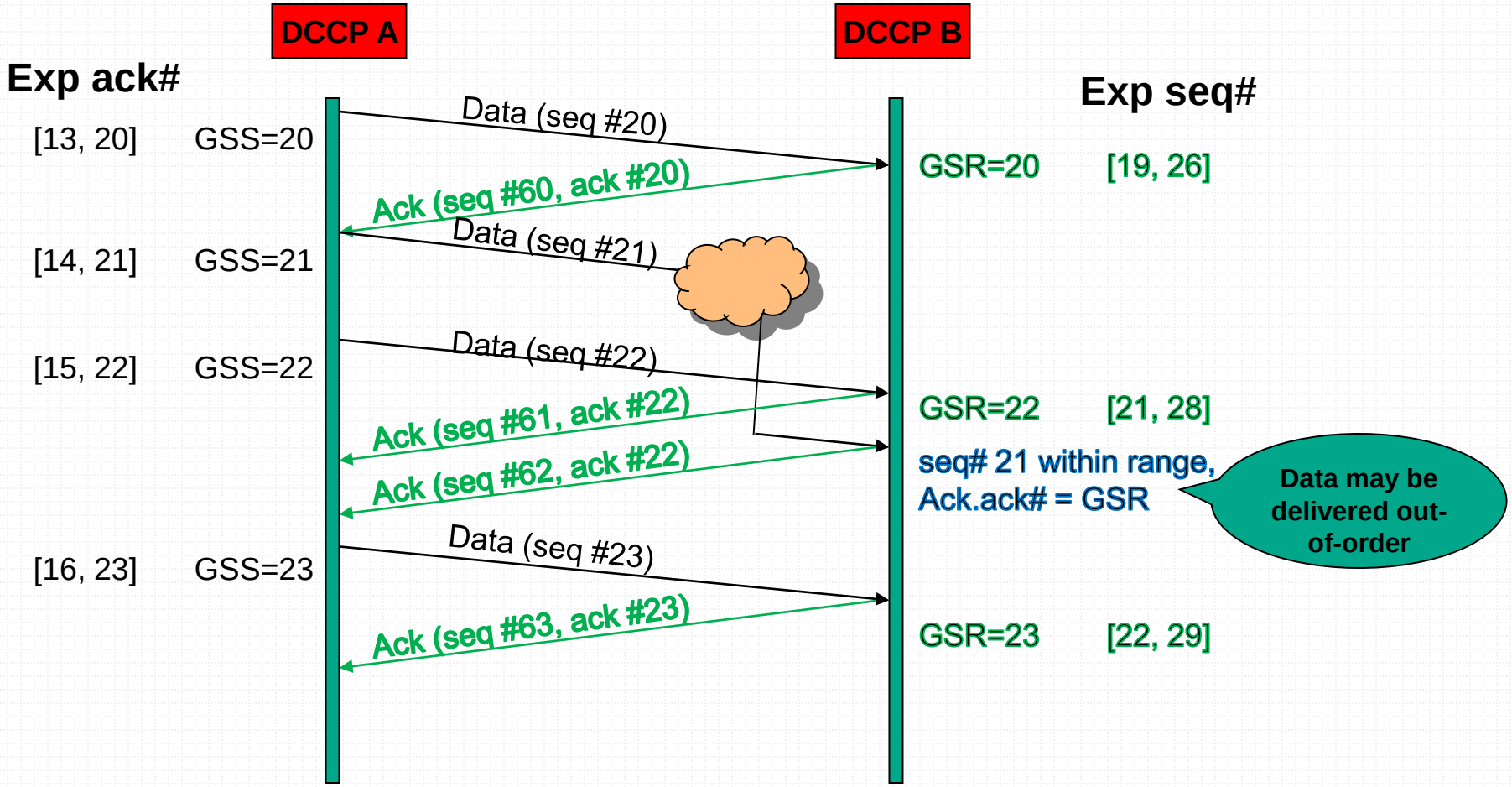
- Example：large burst of loss



- **GSR – Greatest Sequence number Received**
- **GSS – Greatest Sequence number Sent**
- **Window Size = 8**

Datagram Congestion Control Protocol (DCCP)

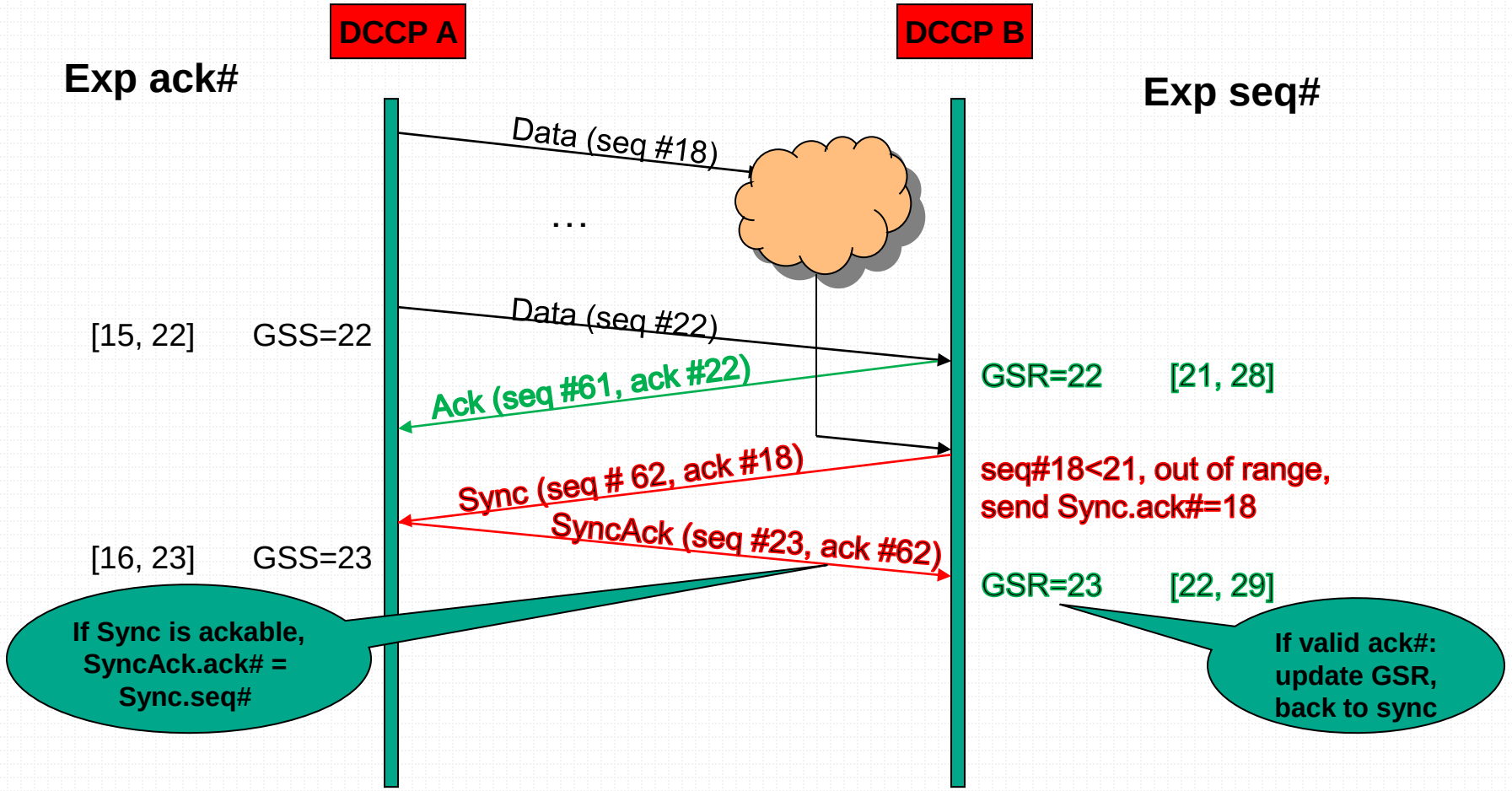
- Example：slight reordering



- **GSR – Greatest Sequence number Received**
- **GSS – Greatest Sequence number Sent**
- **Window Size = 8**

Datagram Congestion Control Protocol (DCCP)

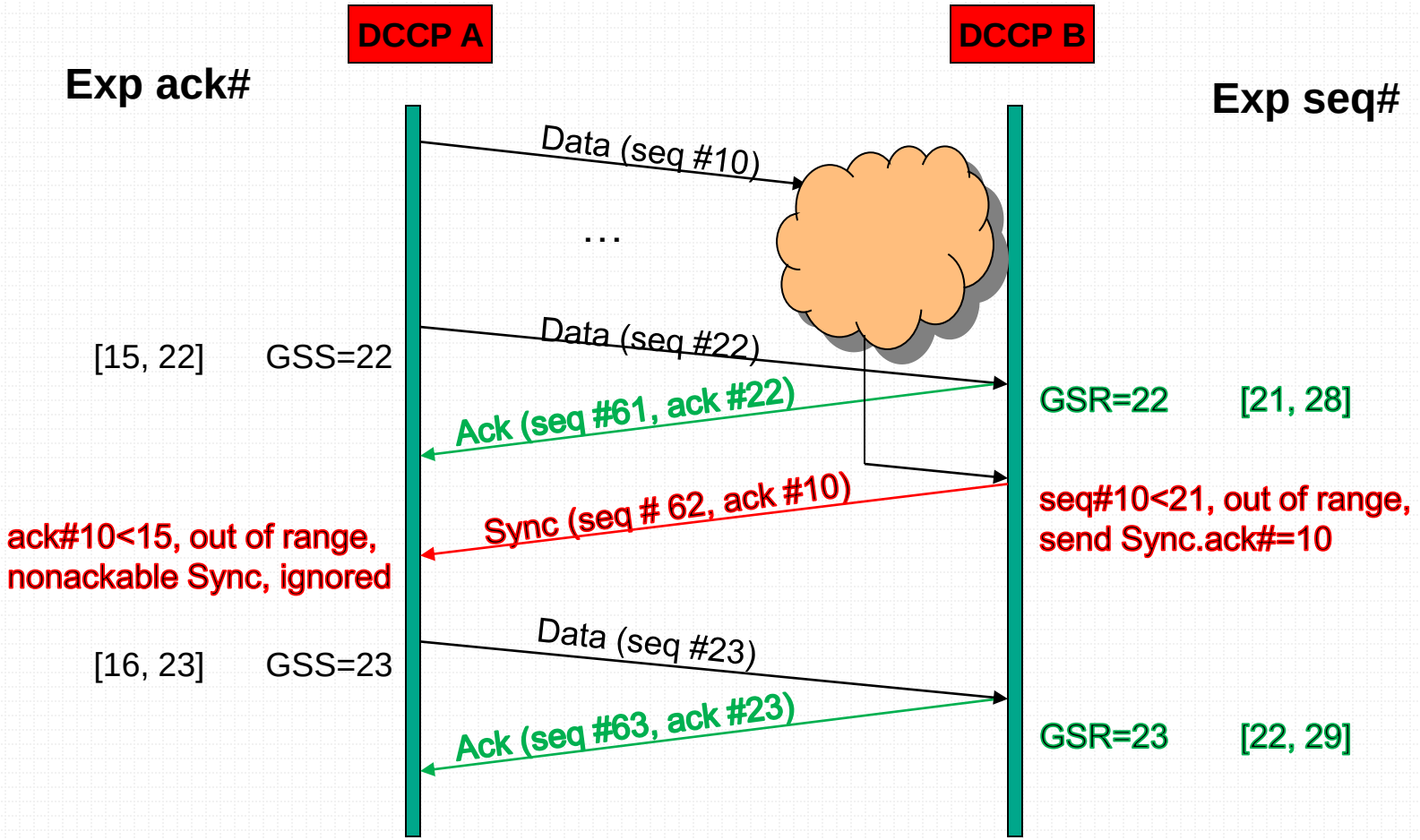
- Example：medium reordering



- **GSR – Greatest Sequence number Received**
- **GSS – Greatest Sequence number Sent**
- **Window Size = 8**

Datagram Congestion Control Protocol (DCCP)

- Example：significant reordering (or blind attack)



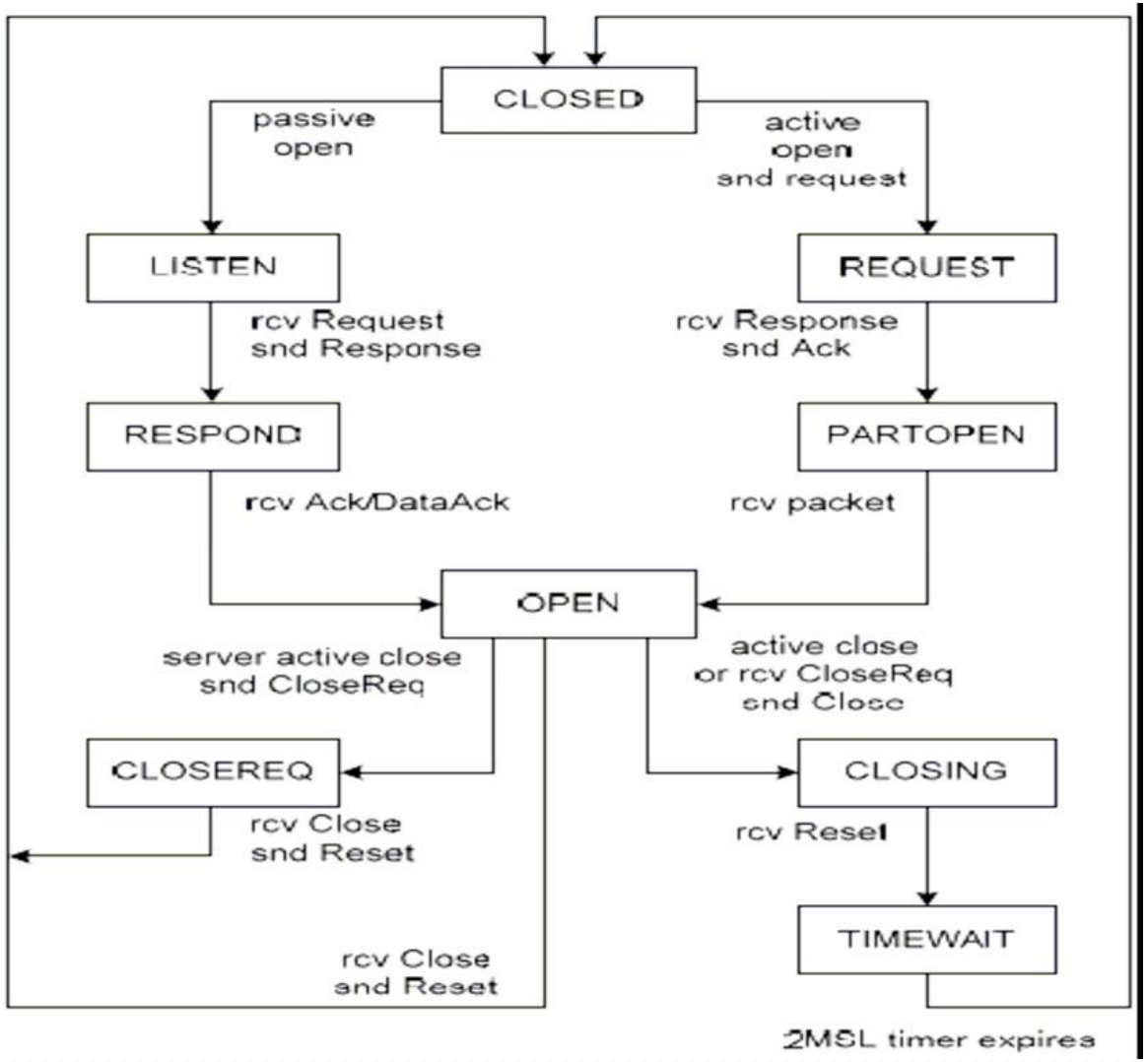
- **GSR – Greatest Sequence number Received**
- **GSS – Greatest Sequence number Sent**
- **Window Size = 8**

Datagram Congestion Control Protocol (DCCP)

- Modular Congestion Control
  - Each congestion control mechanism supported by DCCP is assigned a 1-byte congestion control identifier, or CCID
    - A number from 0 to 255.
    - CCID 2
      - additive increase/ multiplicative decrease (AIMD) congestion-control mechanism whereby the sender adopts a congestion window to control the transmission rate.
    - CCID 3: TCP-Friendly Rate Control (TFRC)
      - The sender uses the loss event ratio the receiver measures, calculates the transmission rate T via the following formula, and accordingly adjusts its sending rate.

Datagram Congestion Control Protocol (DCCP)

- DCCP has defined 9 states that signify these phases' processes
  - CLOSED
  - LISTEN
  - REQUEST
  - RESPOND
  - PARTOPEN
  - OPEN
  - CLOSEREQ
  - CLOSING
  - TIMEWAIT



NTUT NESL

總結

- DCCP
  - Transport layer protocol
  - Unreliable datagrams
  - Modular congestion control
  - Negotiable features

Stream Control Transmission Protocol (SCTP)

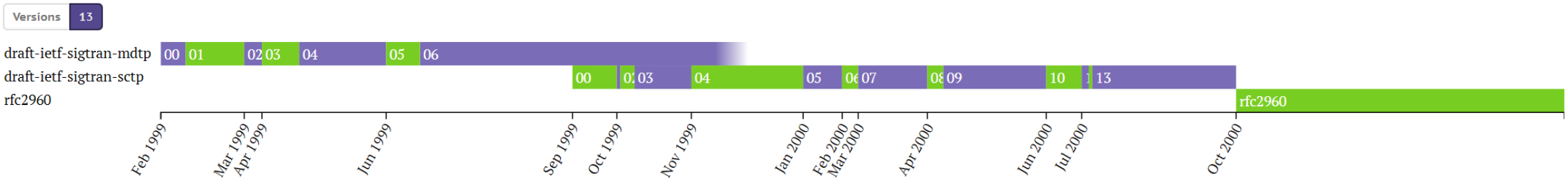
NTUT NESL

IETF sigtran / tsvwg WG

- Signaling Transport (sigtran) **Concluded**
  - This WG produces a standards track proposal or proposals defining transport of signaling protocols using SCTP, based on the requirements identified above.
  - https://datatracker.ietf.org/wg/sigtran/about/
- Transport Area Working Group (tsvwg) **Active**
  - This WG Maintenance of the Stream Control Transmission Protocol (SCTP), which involves bug fixes to the SCTP specifications and their progression along the standards track.
  - https://datatracker.ietf.org/wg/tsvwg/about/

Stream Control Transmission Protocol (SCTP)

- RFC 2960



- RFC 4960

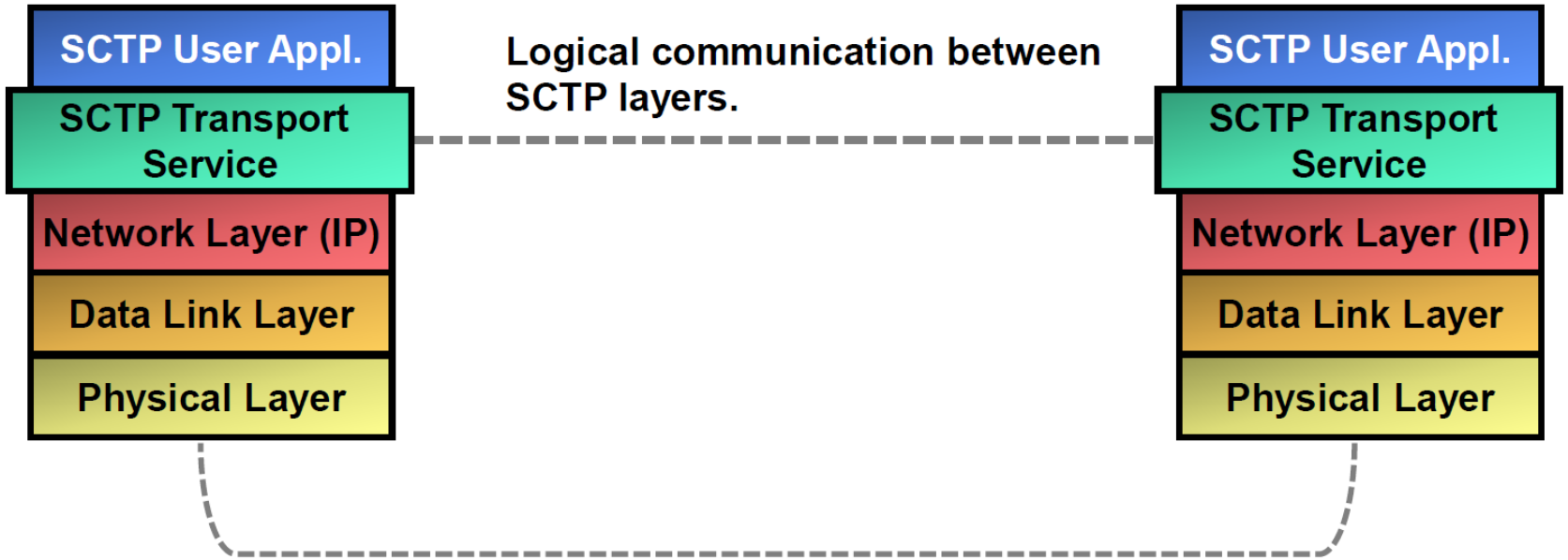
  - Obsoletes RFC 2960, RFC 3309
  - Updated by RFC 7053, RFC 6096, RFC 6335

Stream Control Transmission Protocol (SCTP)

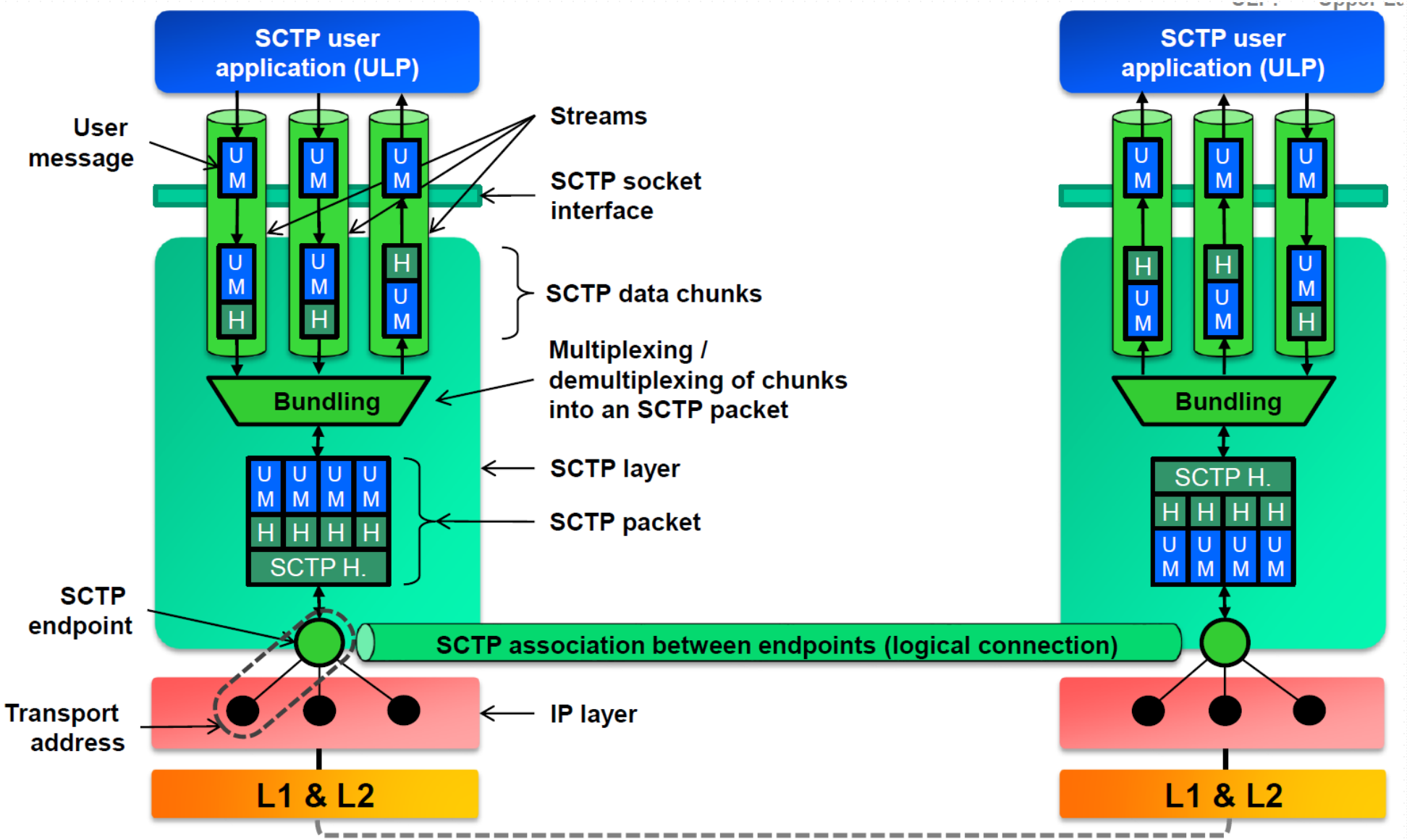
- SCTP evolved from work on IP telephony signaling
  - Like TCP, it provides reliable, full-duplex connections
    - Congestion control similar; some optional mechanisms mandatory
    - Two basic types of enhancements:
      - performance
      - robustness
  - Unlike TCP and UDP, it offers new delivery options that are particularly desirable for telephony signaling and multimedia applications

Stream Control Transmission Protocol (SCTP)

- On top of a connectionless packet network
- Perform better in the presence of losses
  - no strictly ordered delivery
- Designed for new Internet applications

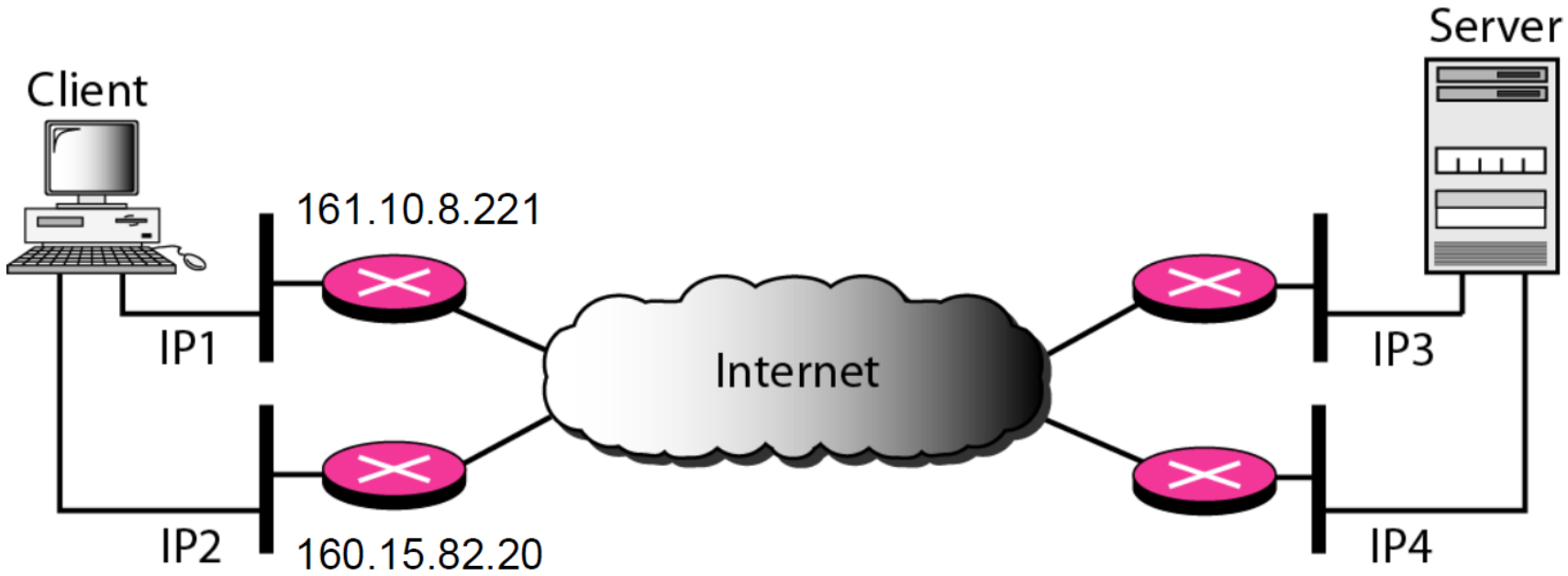


Stream Control Transmission Protocol (SCTP)



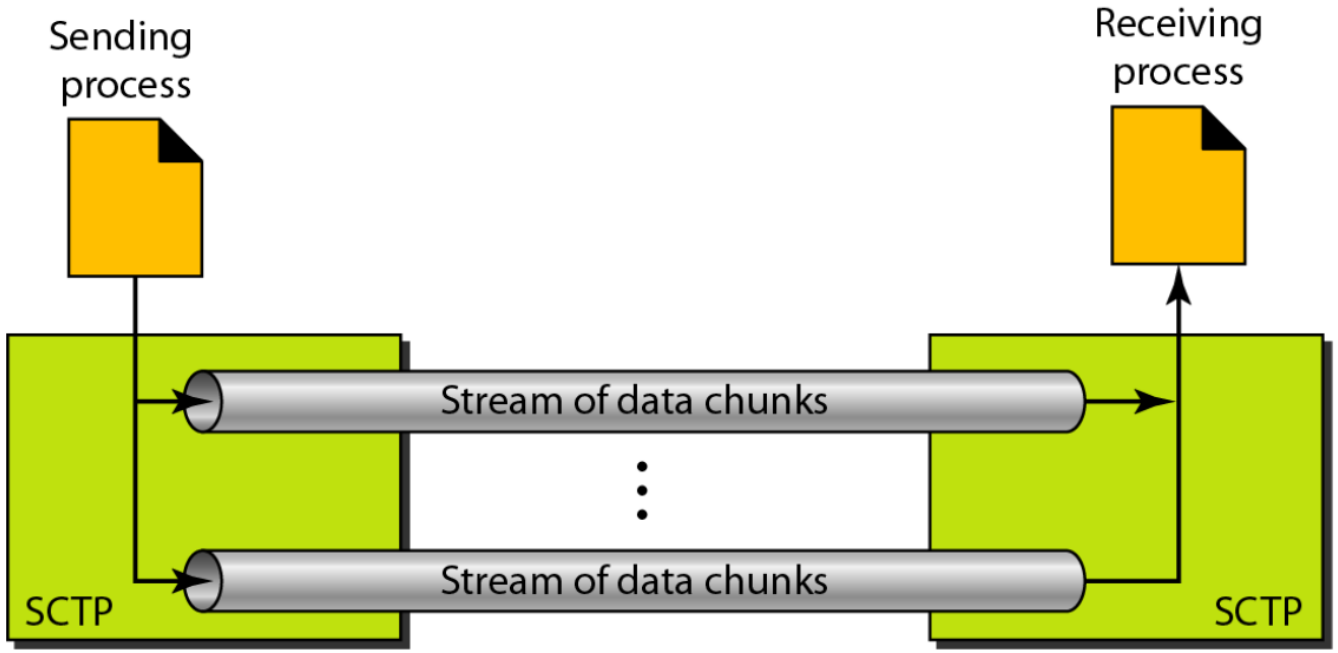
Stream Control Transmission Protocol (SCTP)

- Multi-homing
  - SCTP association allows multiple IP addresses for each end point.
  - This can be represented as a pair of SCTP endpoints.
    - assoc = { [10.1.61.11:2233], [161.10.8.221, 120.1.1.5:80]}



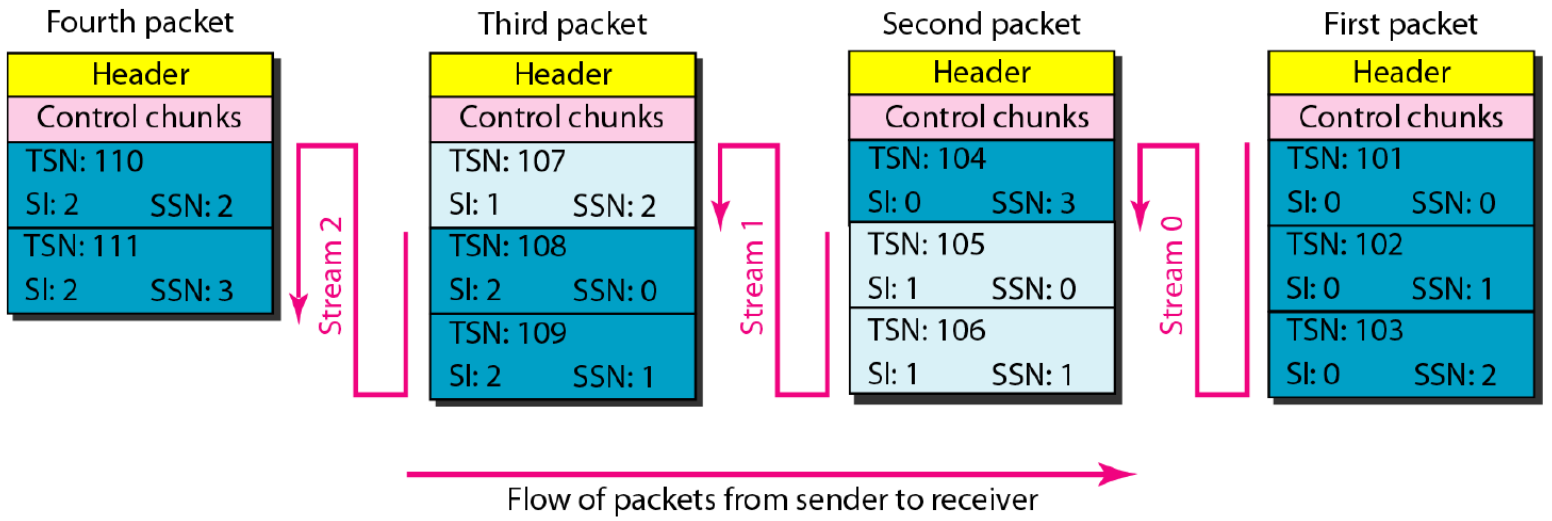
Stream Control Transmission Protocol (SCTP)

- Multi-streams
  - SCTP association can involve multiple streams.
  - Reduces unnecessary head-of-line (HOL) blocking

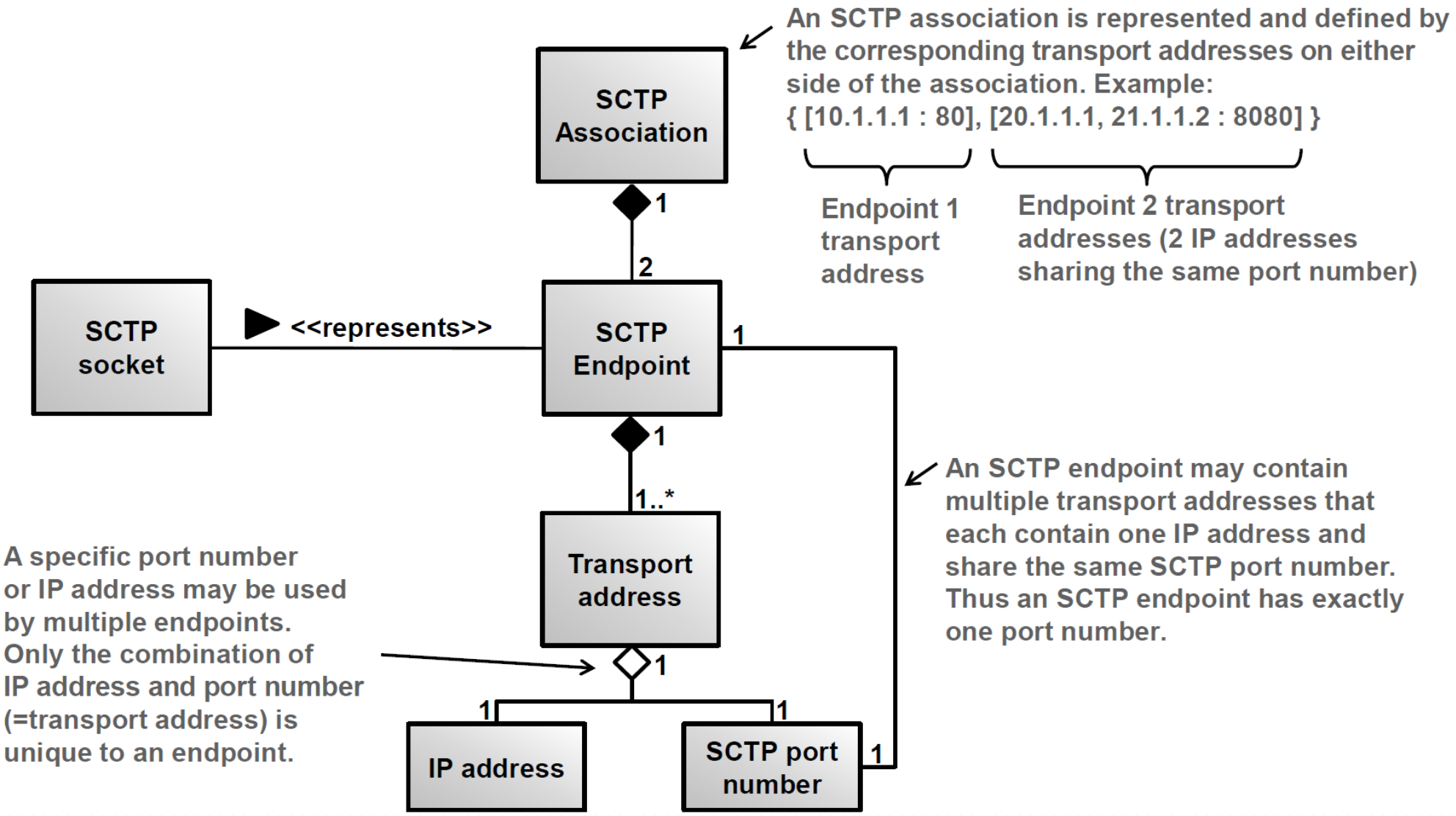


Stream Control Transmission Protocol (SCTP)

- Data chunks are identified by 3 items.
  - TSN：cumulative number defines the association
  - SI：defines the stream
  - SSN：defines the chunk in a stream
- Control information & data information are carried n separate chunks.

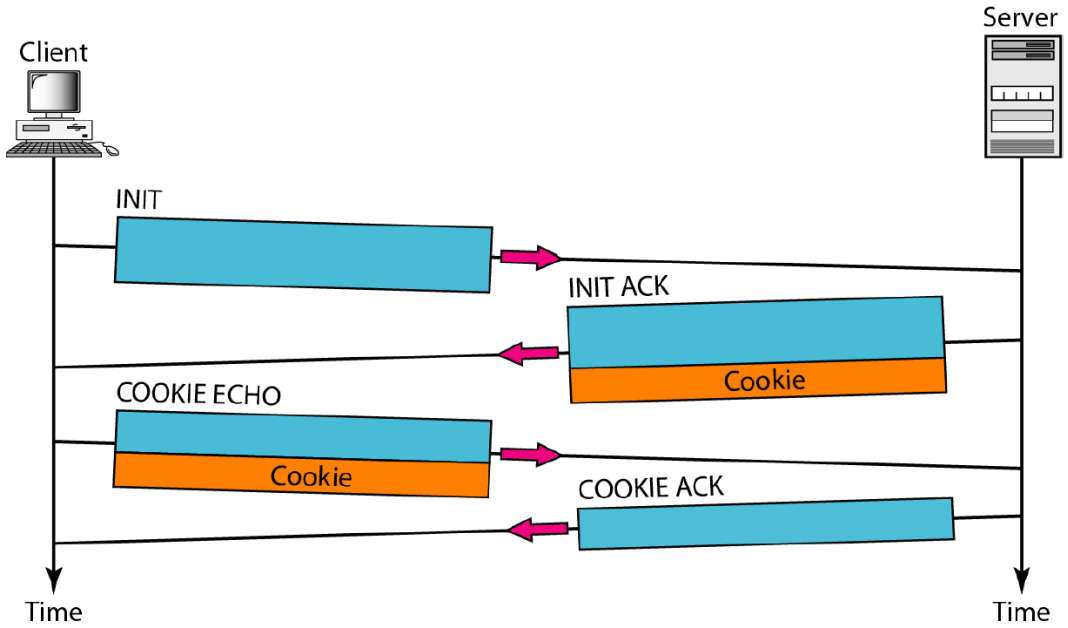


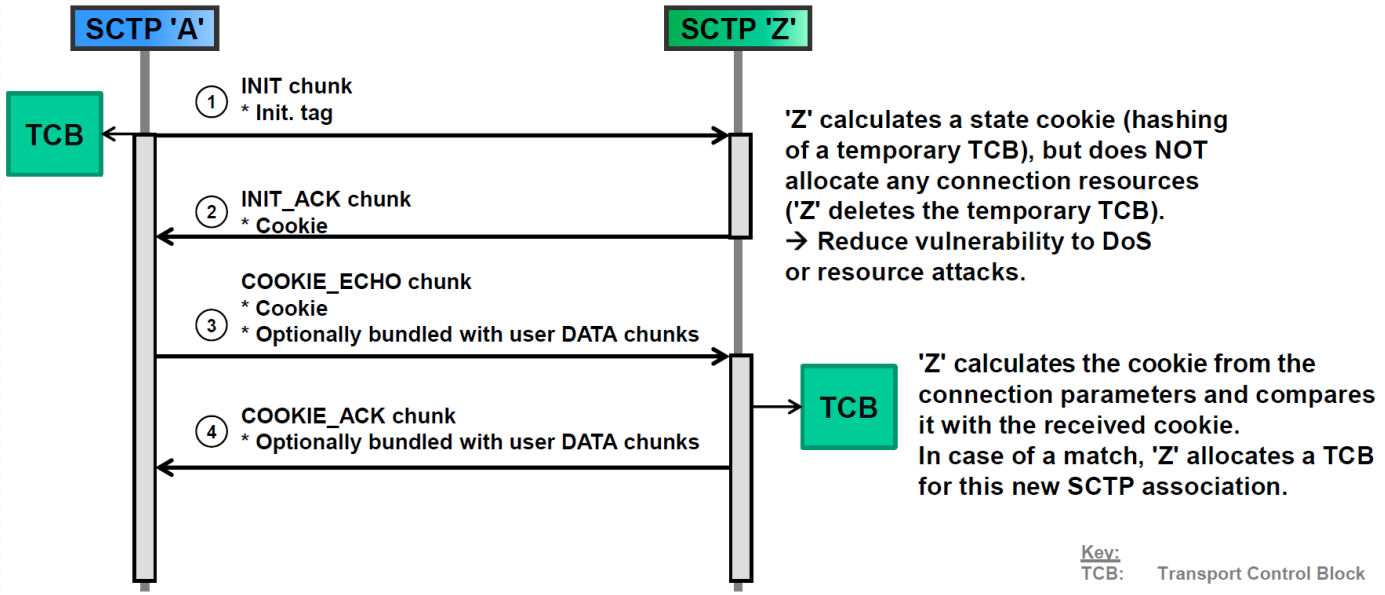
Stream Control Transmission Protocol (SCTP)



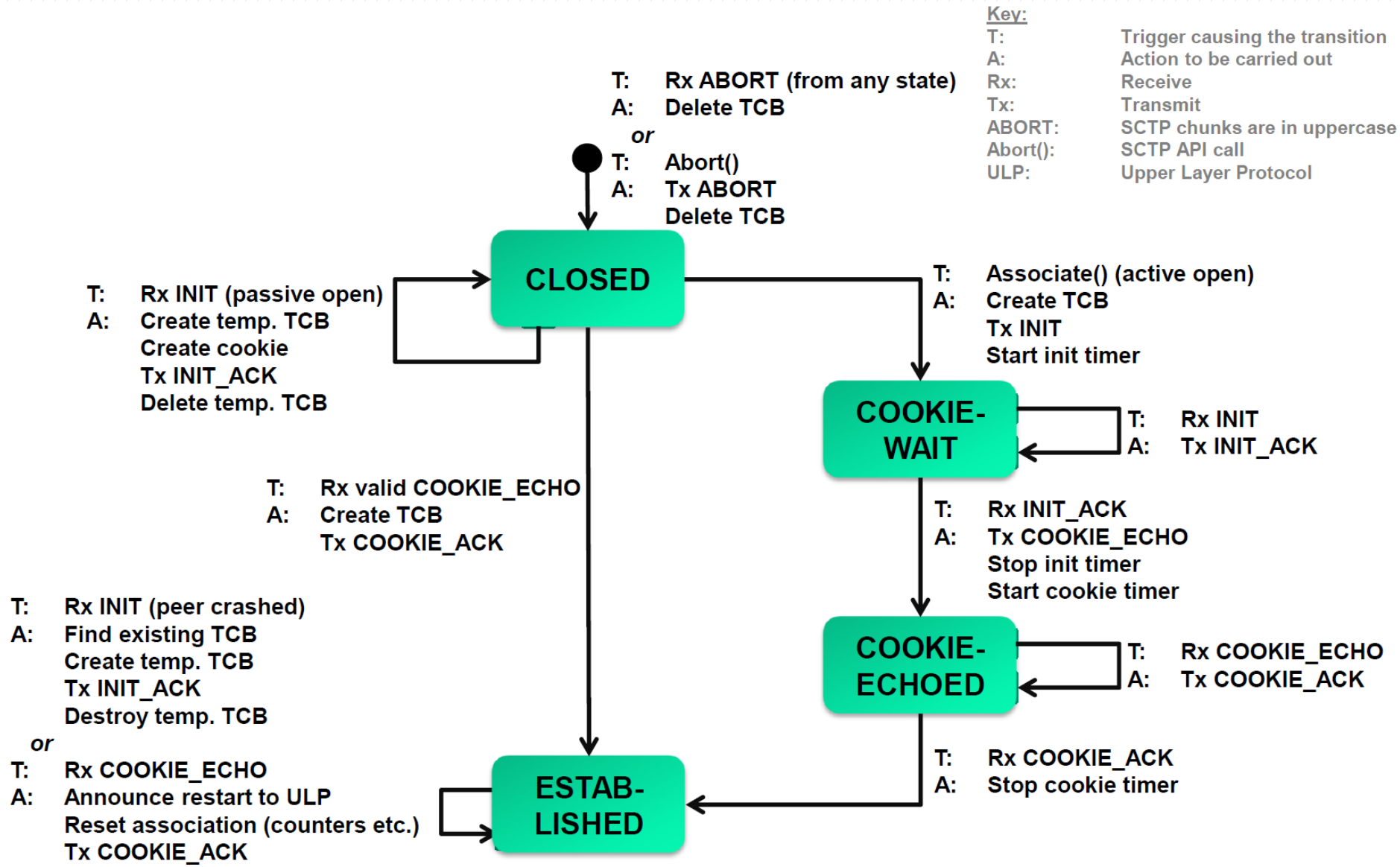
Stream Control Transmission Protocol (SCTP)

- SCTP uses a four-way handshake to set up an association.
  - Cookie to defense against TCP SYN attack.



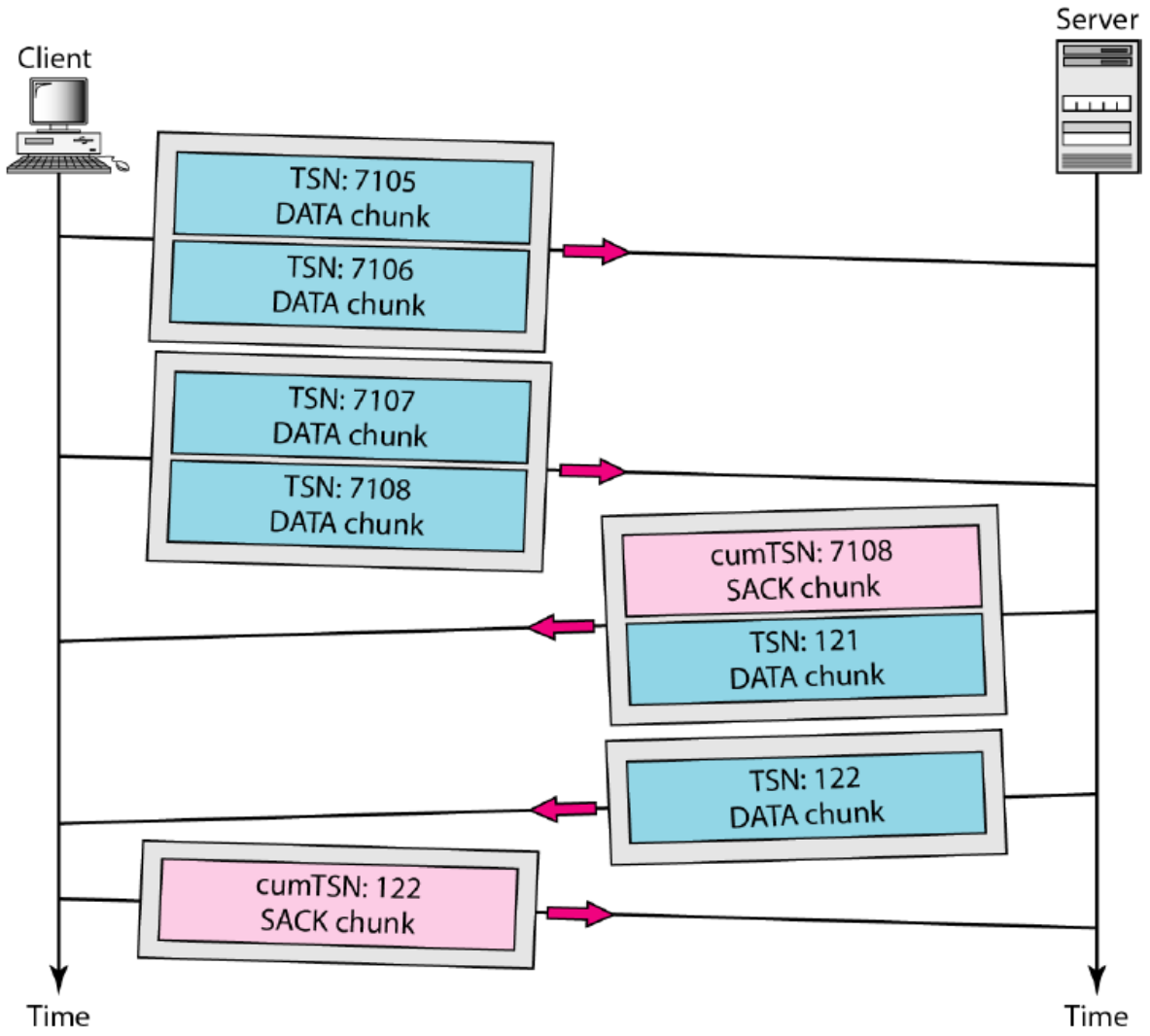


Stream Control Transmission Protocol (SCTP)



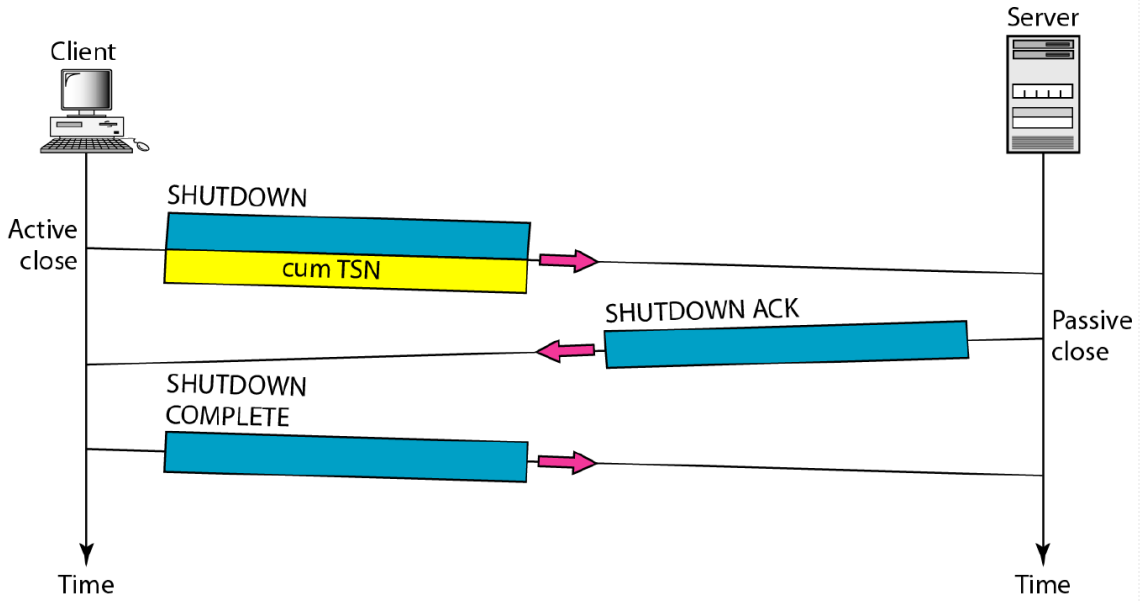
Stream Control Transmission Protocol (SCTP)

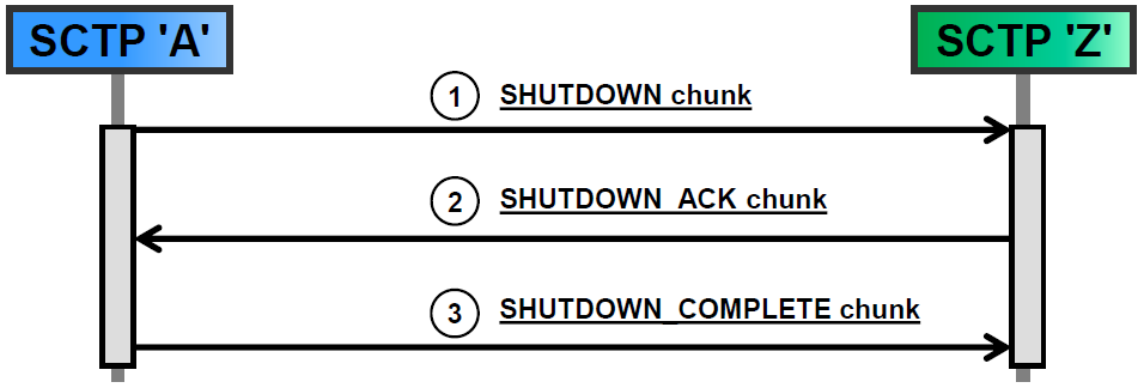
- Only data chunks consume TSNs.
- Data chunks are the only chunks that are acknowledged.



Stream Control Transmission Protocol (SCTP)

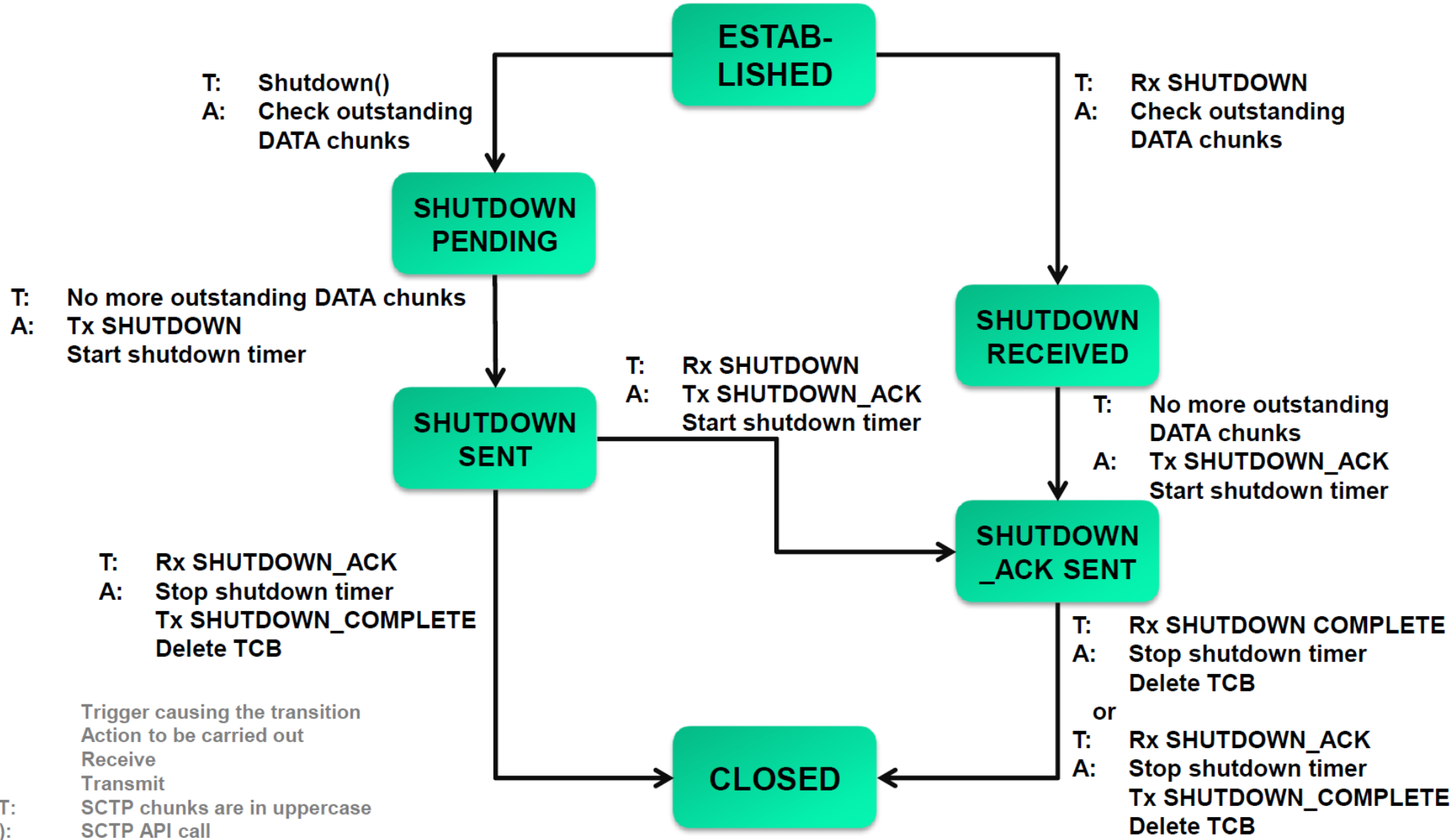
- Acknowledgement defines the cumulative TSN
- The TSN of the last data chunk receive in order.





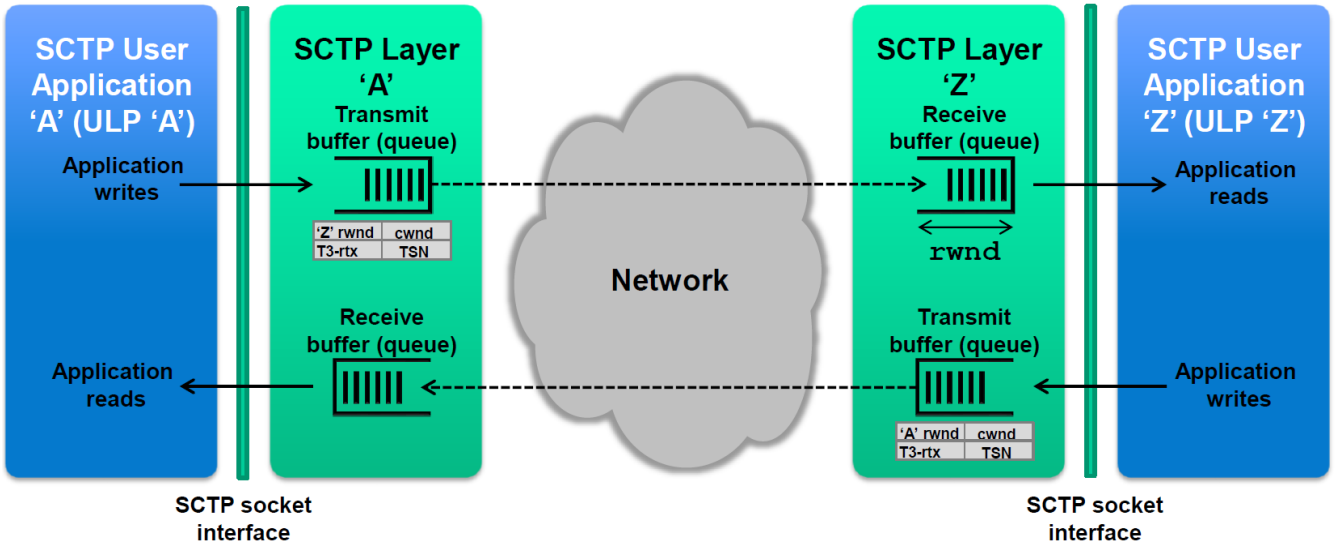
Normal shutdown procedure where either side initiates the shutdown by sending a SHUTDOWN chunk.

Stream Control Transmission Protocol (SCTP)



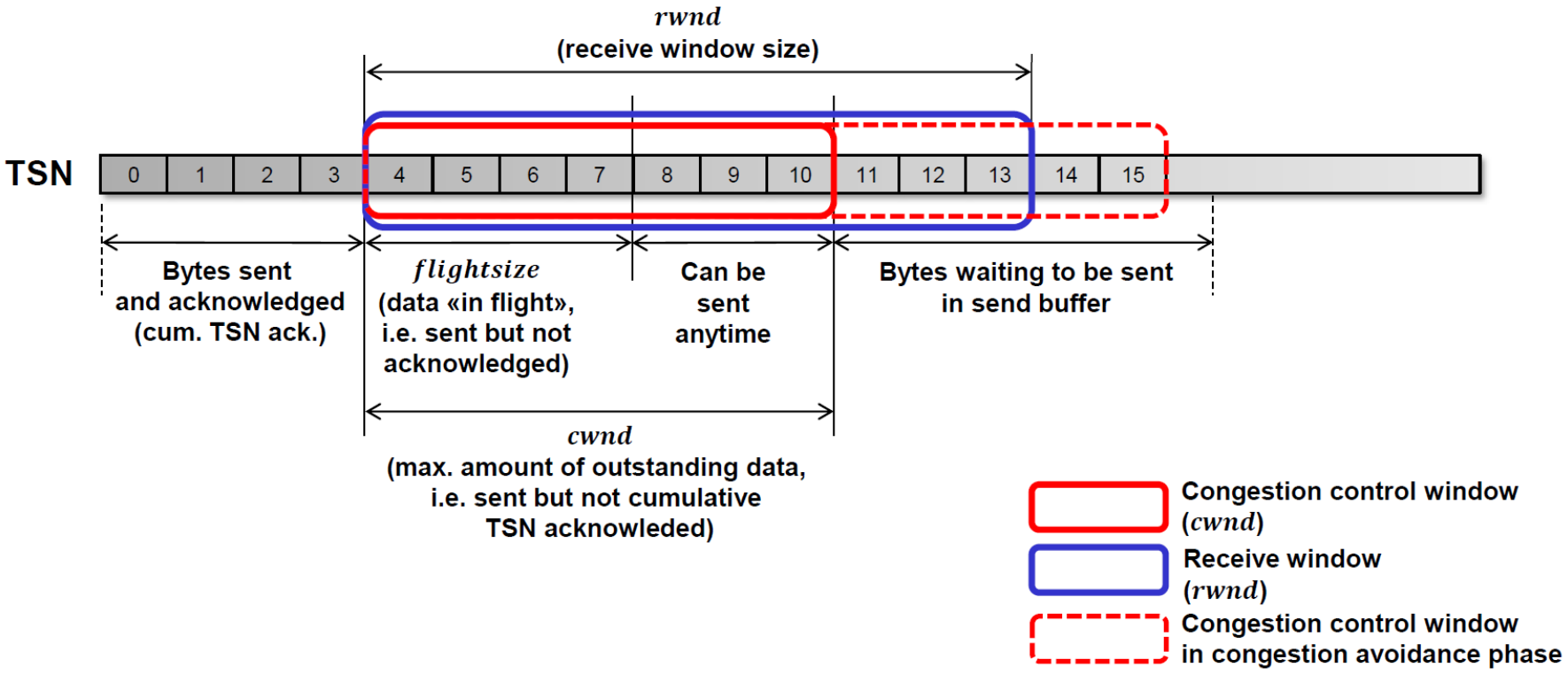
Stream Control Transmission Protocol (SCTP)

- Flow Control
  - TCP, SCTPs flow control mechanism is based on a receiver window size (rwnd)
  - The SCTP flow control algorithm guarantees that the receive buffer never experiences overflow
    - sent data always fits into the receive buffer

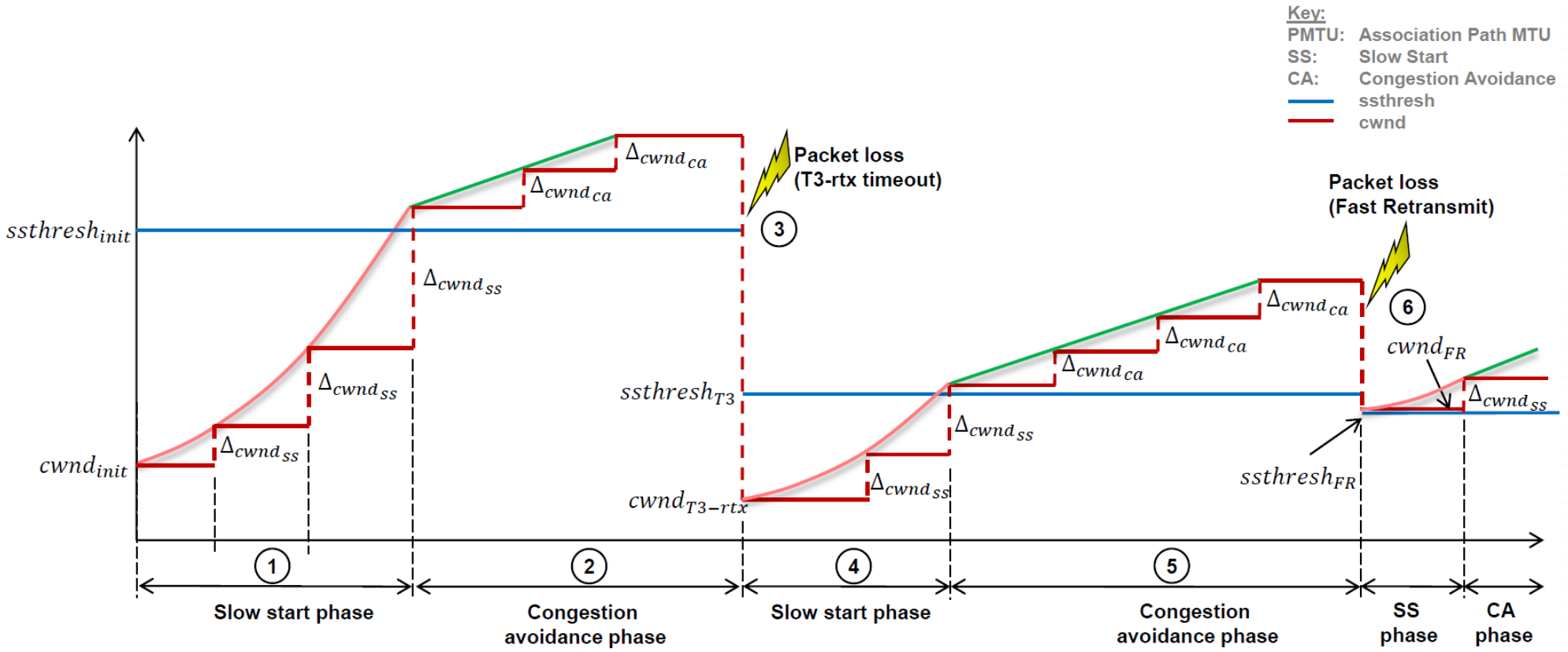


Stream Control Transmission Protocol (SCTP)

- Congestion Control
  - SCTP maintains a separate cwnd parameter for each peer destination address in multihomed scenarios



Stream Control Transmission Protocol (SCTP)



$cwnd_{init} = min(4 * PMTU, max(2 * PMTU, 4380bytes))$

$ssthresh_{init} = \infty\ (or\ a\_rwnd)$

$\Delta_{cwnd_{ss}} = min(DATA_{acked}, PMTU)$

$\Delta_{cwnd_{ca}} = 1 * PMTU\ per\ RTT$

$cwnd_{T3-rtx} = 1 * PMTU$

$cwnd_{FR} = ssthresh_{FR}$

$ssthresh_{T3} = ssthresh_{FR} = max\left(cwnd/_2, 4 * PMTU\right)$

$burst_{max} = 4$

NTUT NESL

總結

- SCTP
  - Transport layer protocol
  - Support multi-homing & streams
  - Flow control
  - Congestion control