Protocol Principles

Framing, FCS and ARQ

Link Layer Tasks

- Framing
- Frame Protection
- Optional Addressing
- Optional Error Recovery
- Connection-oriented or connectionless mode
- Optional Flow Control

Building a Frame

- Consists of
  - Data
  - Metadata (Header or "Overhead")
- Requires synchronous physical transmission (PLL)
  - Arbitrary frame lengths

Preamble

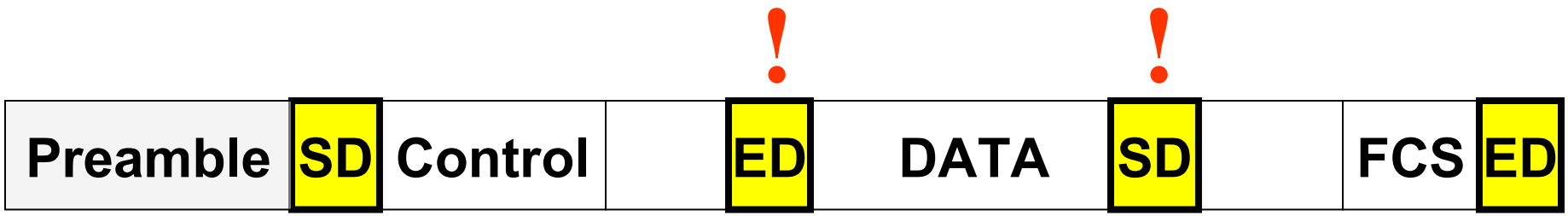
- Enables PLL synchronization
  - Typically a 0101010...-pattern
  - Example: 8 Byte preamble in Ethernet frames
- Note: Only necessary when sender- and receiver-clock are not synchronized between frames
  - Asynchronous physical layer

Frame Synchronization

- Beginning and ending of a frame is indicated by SD and ED symbols
  - bit-patterns or code-violations
  - lenght-field can replace ED (802.3)
  - Idle-line can replace ED (Ethernet)
- Also called "Framing"

Protocol Transparence

- What, if delimiter symbols occur within frame ?
- Solution:
  - Byte-Stuffing
  - Bit-Stuffing



Byte Stuffing

- Some character-oriented protocols divide data stream into frames
  - Old technique, not so important today
- Data Link Escape (DLE) character indicates special meaning of next character

Data to send:

A	B	C	DLE	E	F	G	ETX	H	I	STX	H

DLE	STX	A	B	C	DLE	DLE	E	F	G	ETX	H	I	STX	H	DLE	ETX

Bit Stuffing

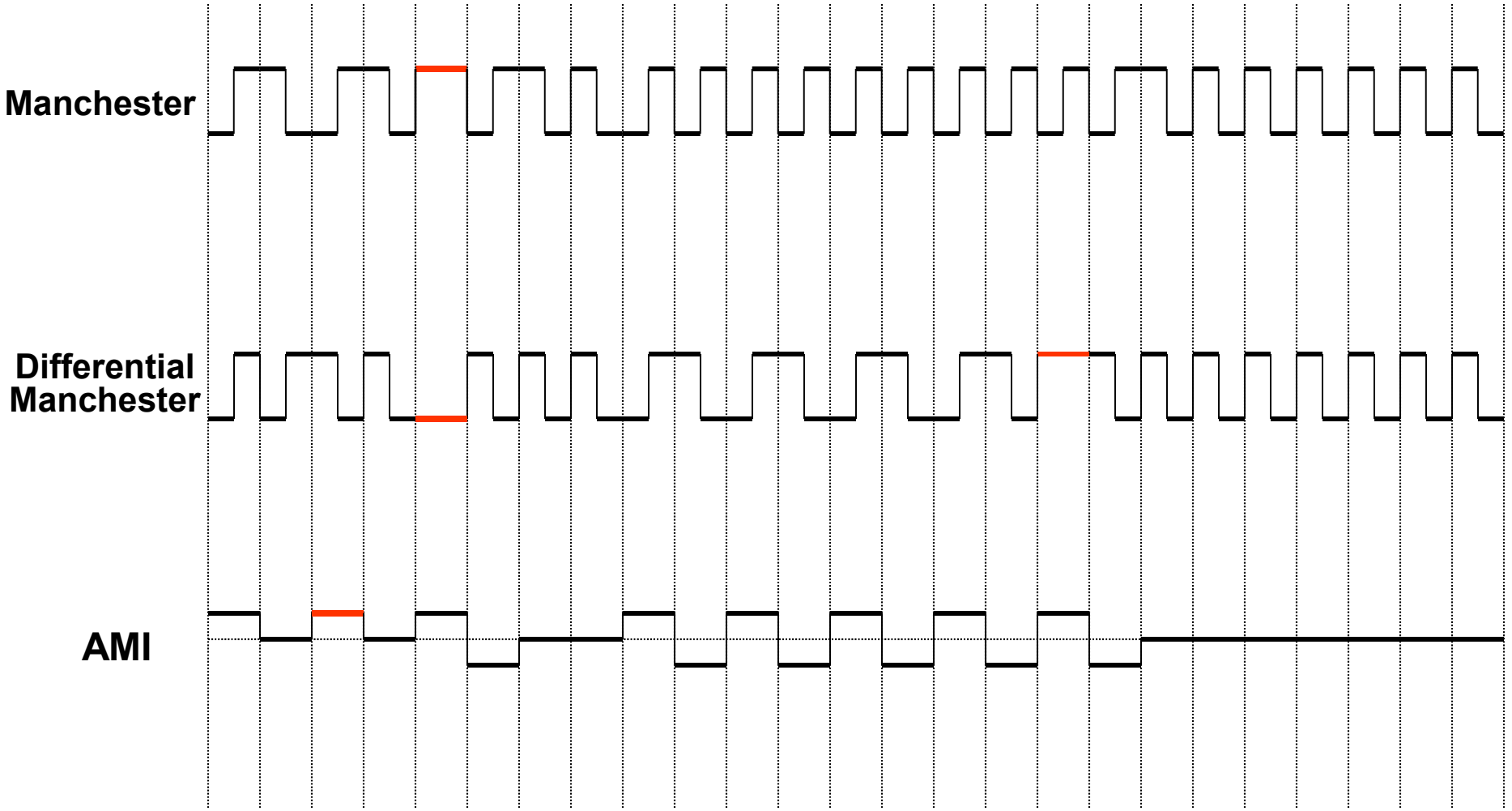
- Used in bit-oriented protocols
  - Used by most protocols
  - Bits represent smallest transmission unit
- HDLC-like framing: 01111110-pattern
- Rule:
  - Trasceiver-HW inserts a zero after five ones
  - Receiver rejects each zero after five ones

Data to send:

0100111111000111111100101100110

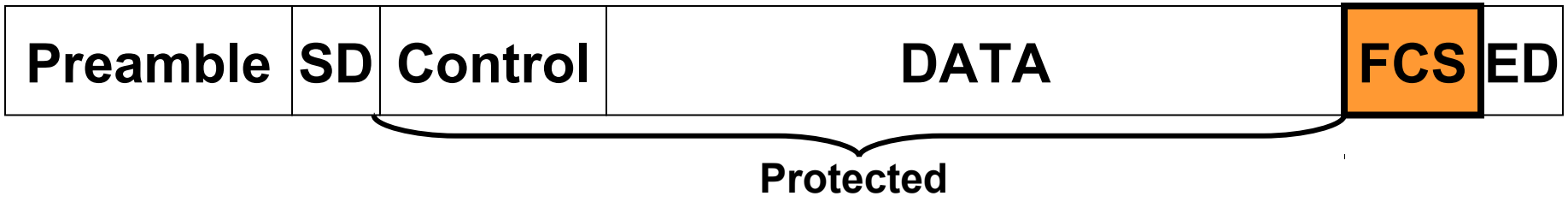
01111110	0100111111000011111011 00101100110	01111110

Code Violations



Frame Protection

- A frame check sequence (FCS) protects the integrity of our frame
  - From Sunspots, Mobile-Phones, Noise, Heisenberg and others
- FCS is calculated upon data bits
  - Different methods based on mathematical efforts: Parity, Checksum, CRC
- Receiver compares its own calculation with FCS



FCS Methods

- Parity
  - Even (100111011) or odd (100111010) parity bits
  - Examples: Asynchronous character-transmission and memory protection
- Checksum
  - Sum without carries (XOR operation)
  - Many variations and improvements
  - Examples: TCP and IP Checksums

Checksum Example: ISBN

- 100% Protection against
  - Single incorrect digits
  - Permutation of two digits
- Method:
  - 10 digits, 9 data + 1 checksum
  - Each digit weighted with factors 1-9
  - Checksum = Sum modulo 11
  - If checksum=10 then use "X"

ISBN 0-13-086388-2

0*1+1*2+3*3+0*4+8*5+6*6+3*7+8*8+8*9 = 244

244 modulo 11 = 2

Cyclic Redundancy Check

- CRC is one of the strongest methods
- Bases on polynomial-codes
- Several standardized generator-polynomials
  - CRC-16: $x^{16}+x^{15}+x^{2}+1$
  - CRC-CCITT: $x^{16}+x^{12}+x^{5}+1$

Forward Error Correction

- Required for "extreme" conditions
  - High BER, EMR
  - Long delays, space-links
- Introduces much redundancy
  - Example: Reed-Solomon codes, Hamming-codes

0	1	0	1	0
1	1	1	0	1
0	1	0	0	1
0	0	1	1	0
1	1	0	0	0

0	1	0	1	0
1	1	0	0	1
0	1	0	0	1
0	0	1	1	0
1	1	0	0	0

Control Field

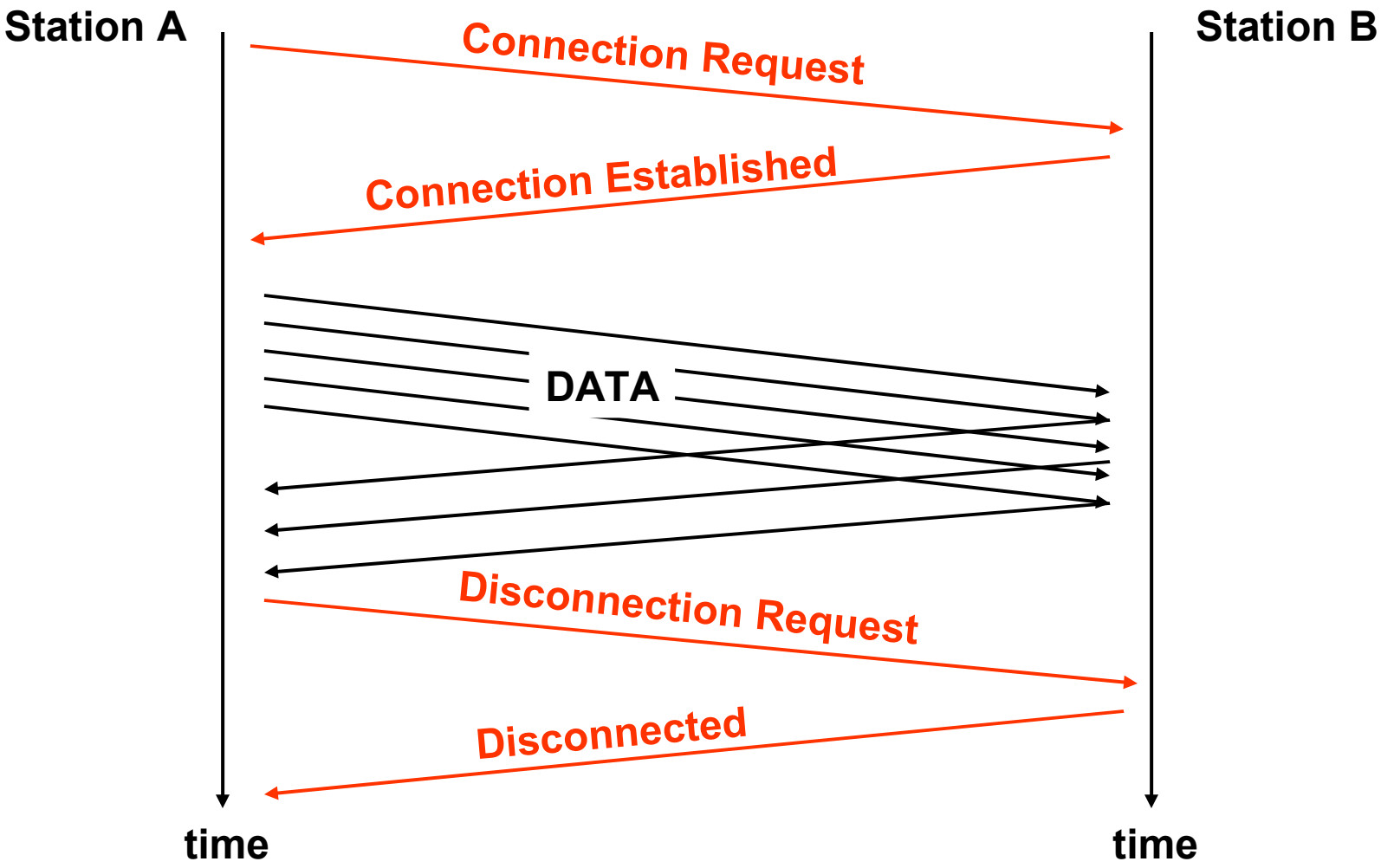
- Contains protocol information
  - Addressing
  - Sequence numbers
  - Acknowledgement Flag
  - Frame Type
  - SAP or Payload Type
  - Signalling information

Connection-Oriented Protocols

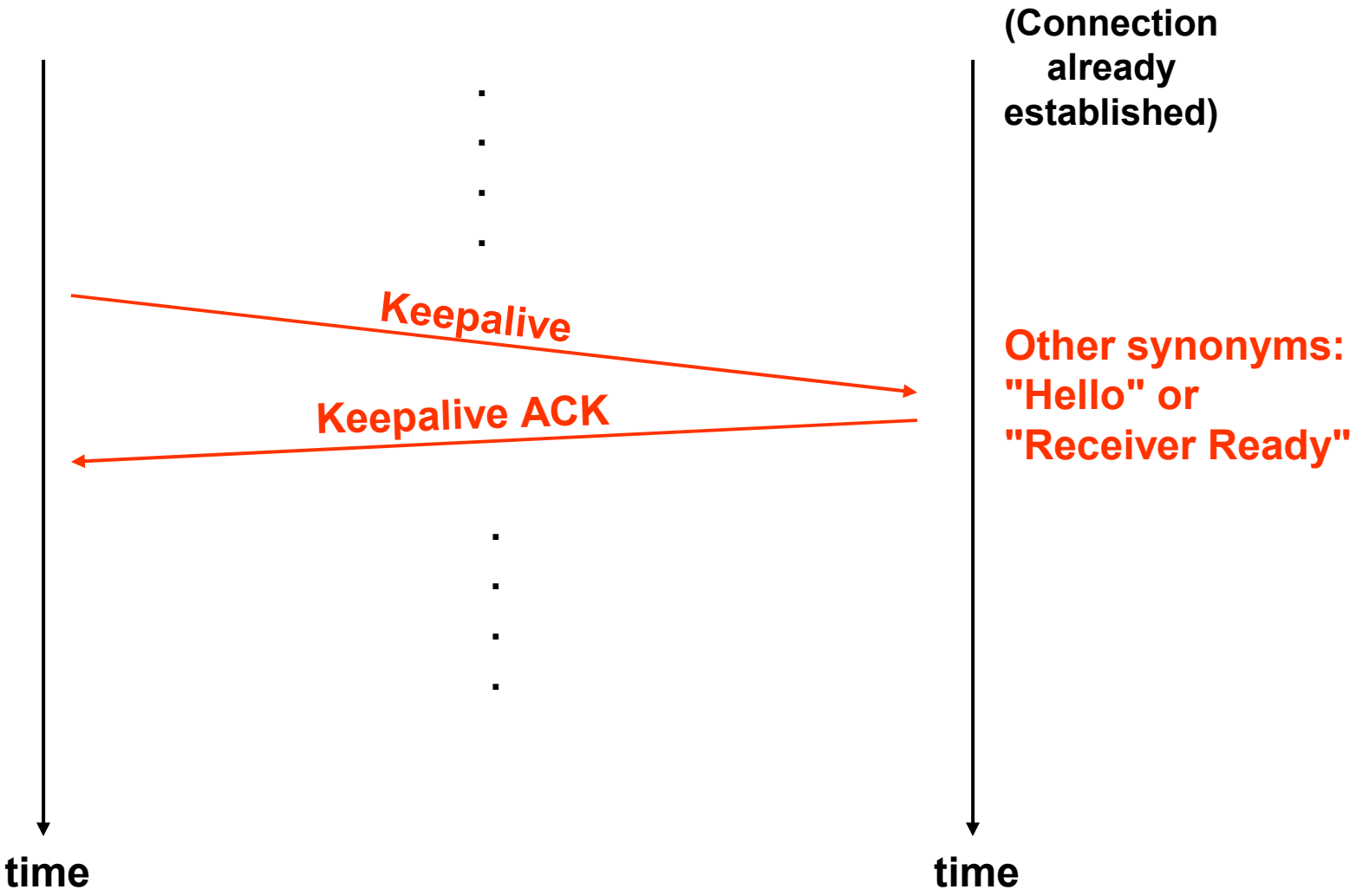
- Different definitions
  - Some say *"protocols without addressing information"* and think of circuit-switched technologies
  - Some say *"protocols that do error recovery"*

– **Correct:** "protocols that require a connection establishment before sending data and a disconnection procedure when finished"

CO-Protocol Procedures (1)



CO-Protocol Procedures (2)



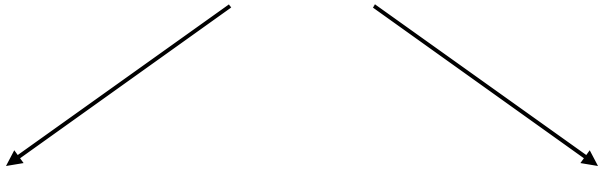
CO-Issues

- Establishment delay
- Traffic desriptor during establishment (QoS)
- Additional frame types necessary
  - Connection establishment
  - Disconnect
  - Keepalive
- ARQ possible (Error Recovery)

Automatic Repeat Request

- ARQ protocols guarantee correct delivery of data
  - Receiver sends acknowledgements
  - Acknowledgements refer to sequence numbers
  - Missing data is repeated
- When do we need this?
  - For most data traffic (FTP, HTTP, ...)
  - Not for real-time traffic (VoIP)

ARQ Variants

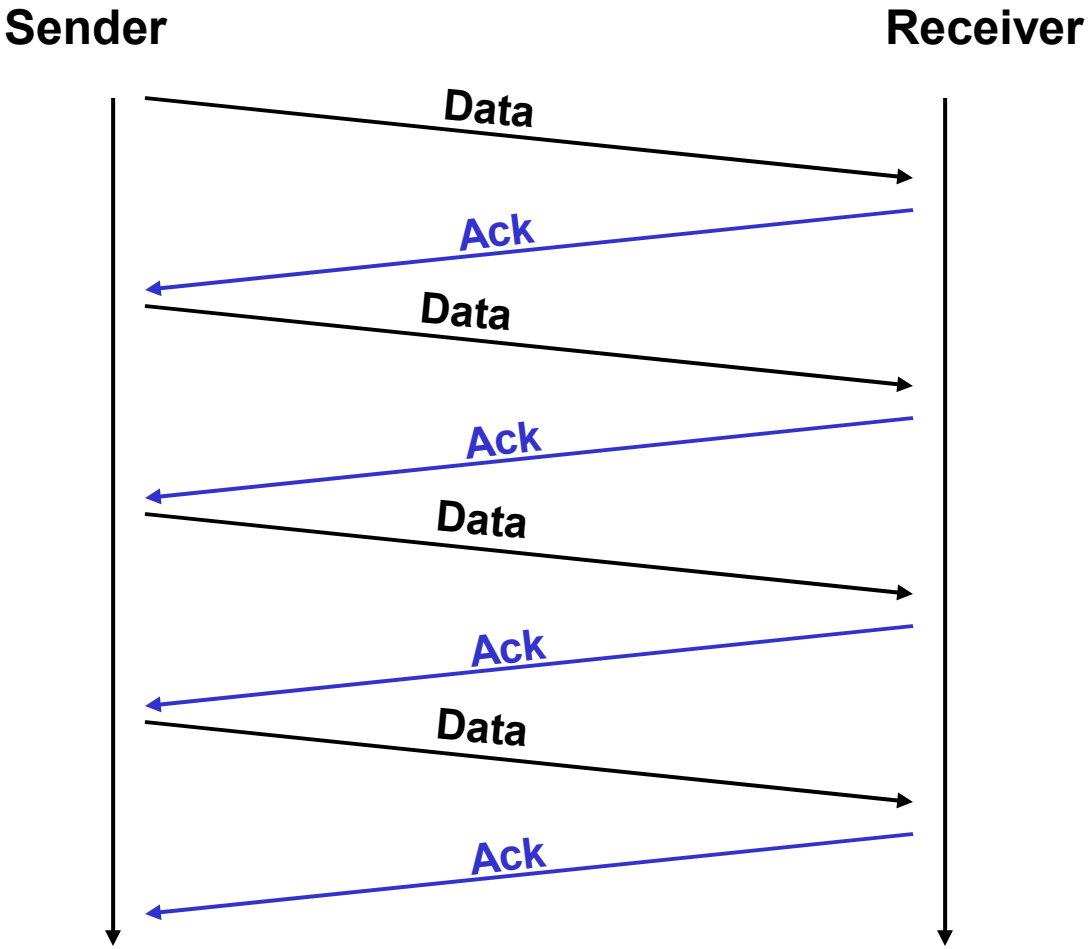


Idle-RQ

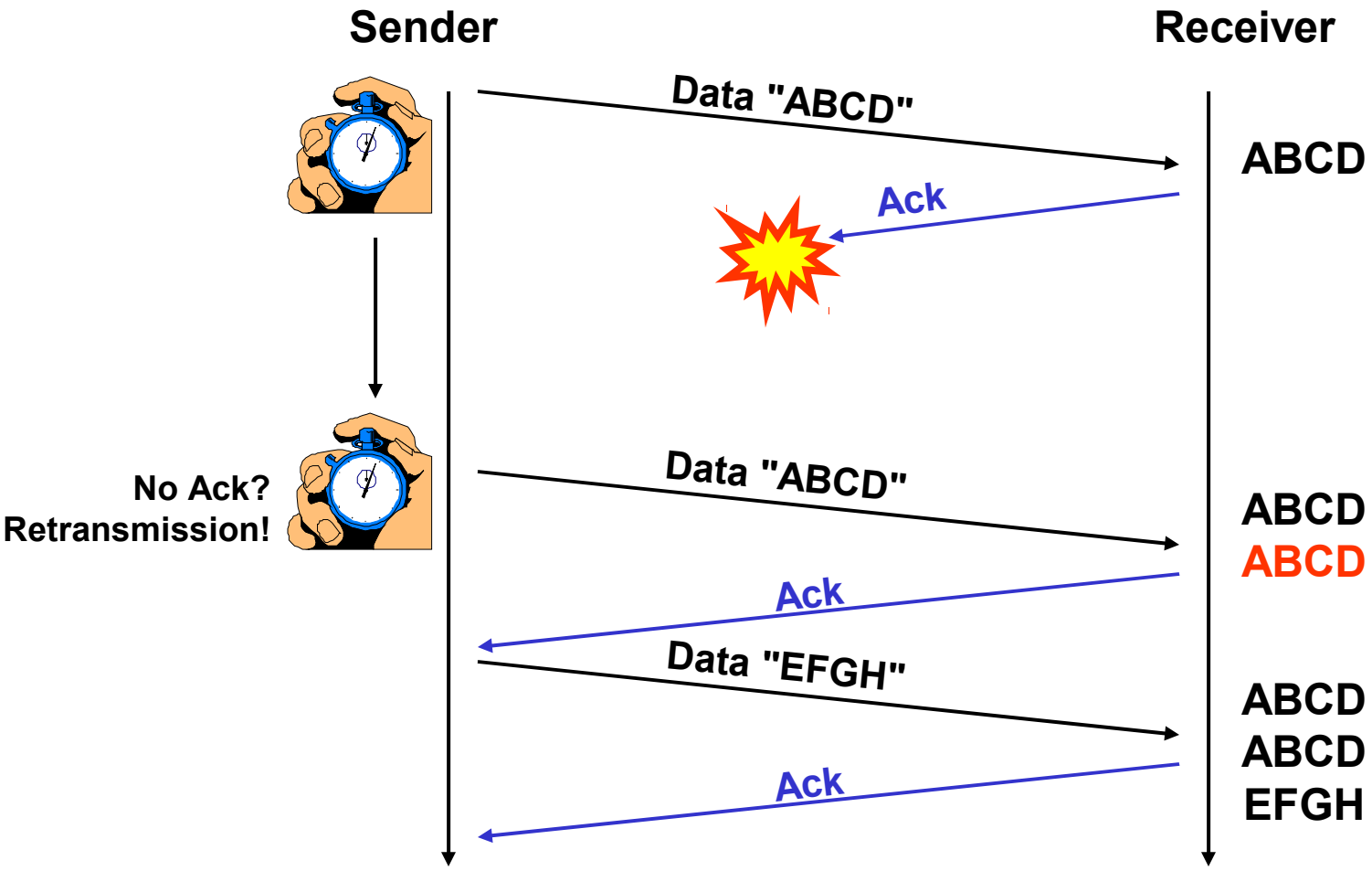
Continuous-RQ

- Selective ACK
- Positive ACK
- GoBackN
- SREJ

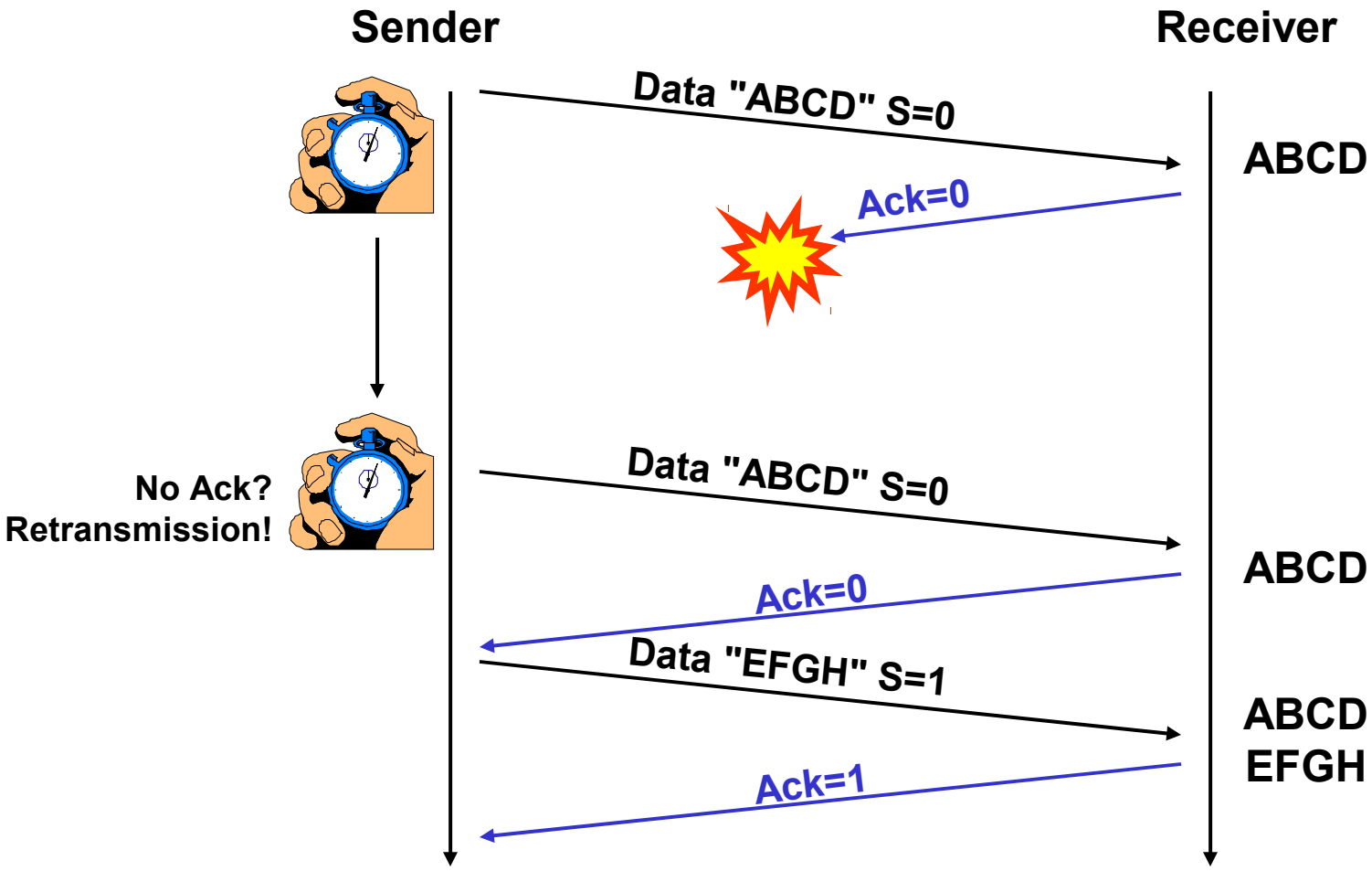
Idle-RQ



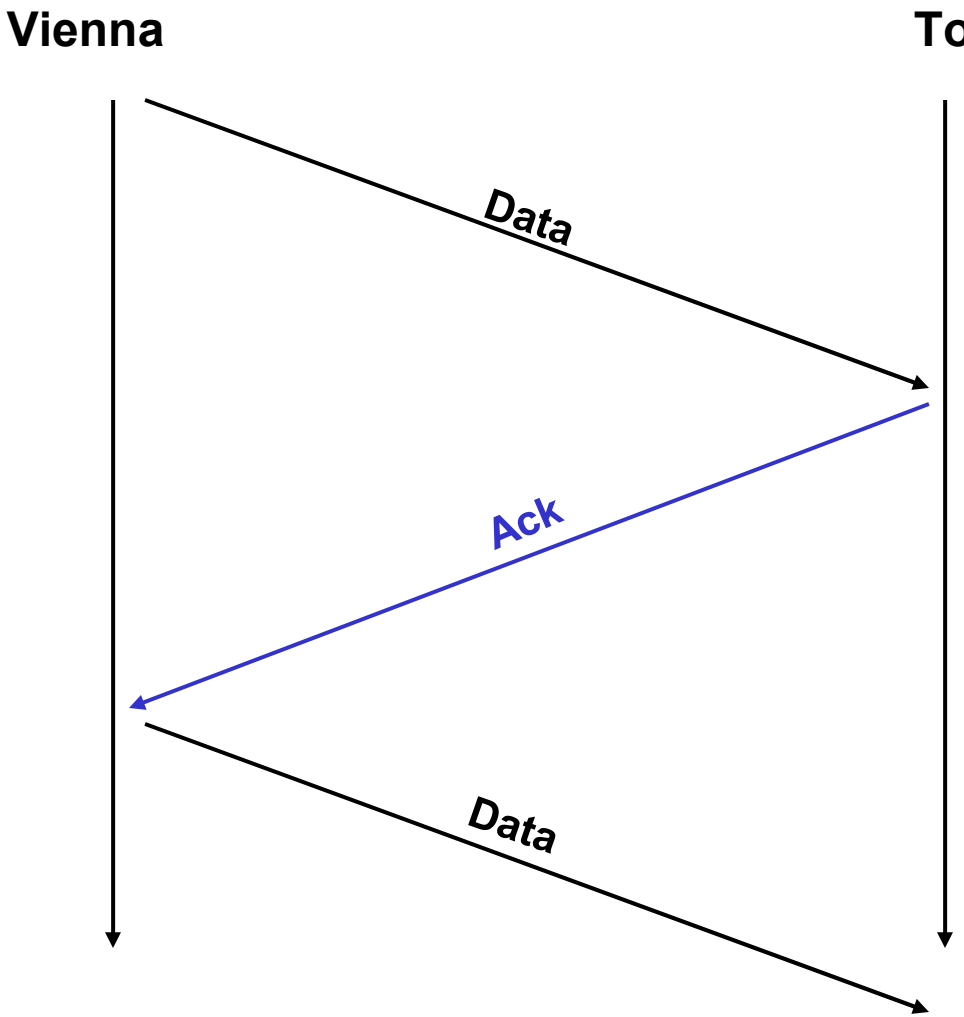
Without Sequence Numbers:



With Sequence Numbers:



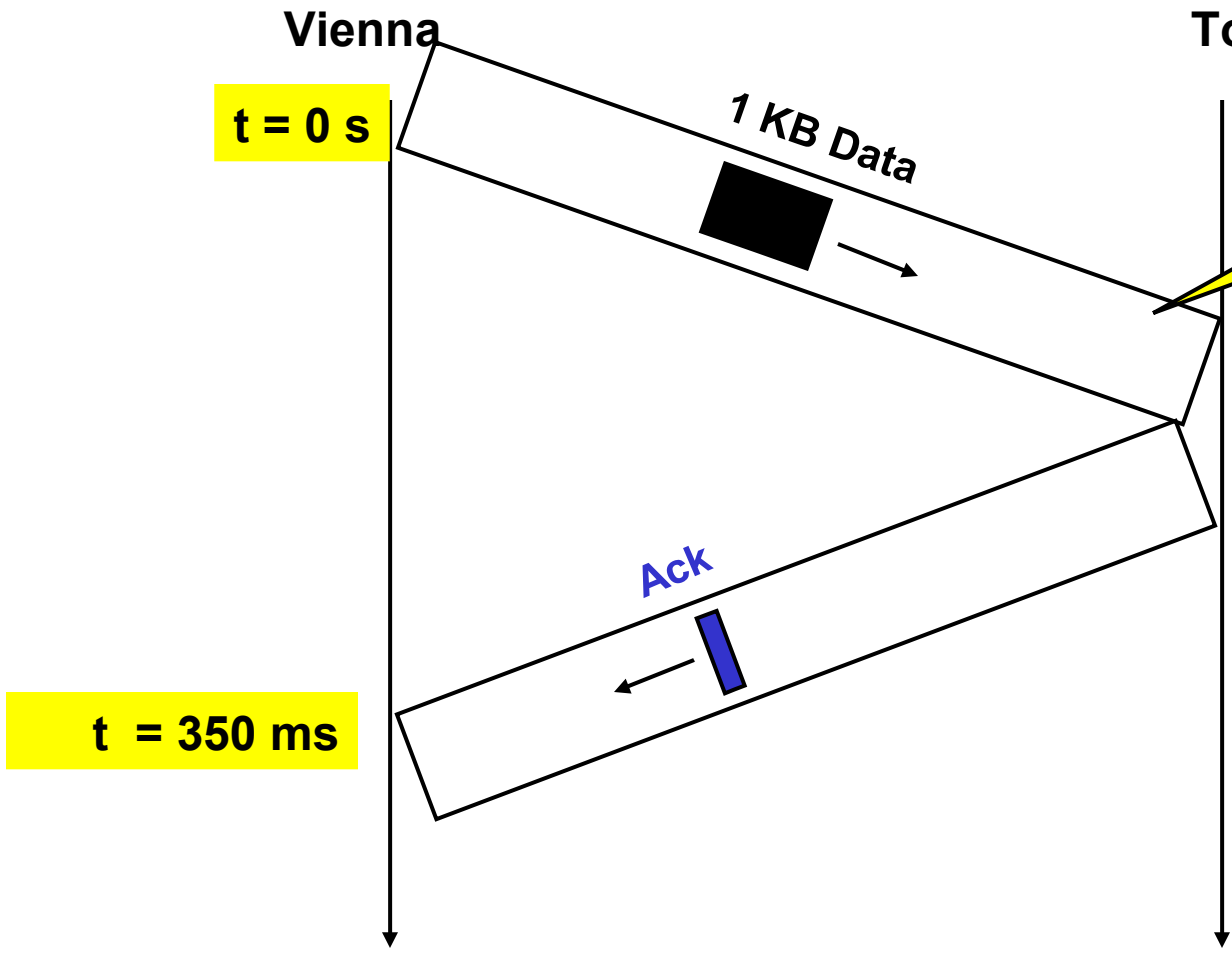
Slow !



"Stop and Wait":
Data is traveling round the earth while sender waits for the acknowledgement.

In the meantime no data can be sent !!

Empty Pipe !



This pipe allows
1.5 Mbit/s × 350 ms
= 525,000 bit ≅ 64 KB
of data.

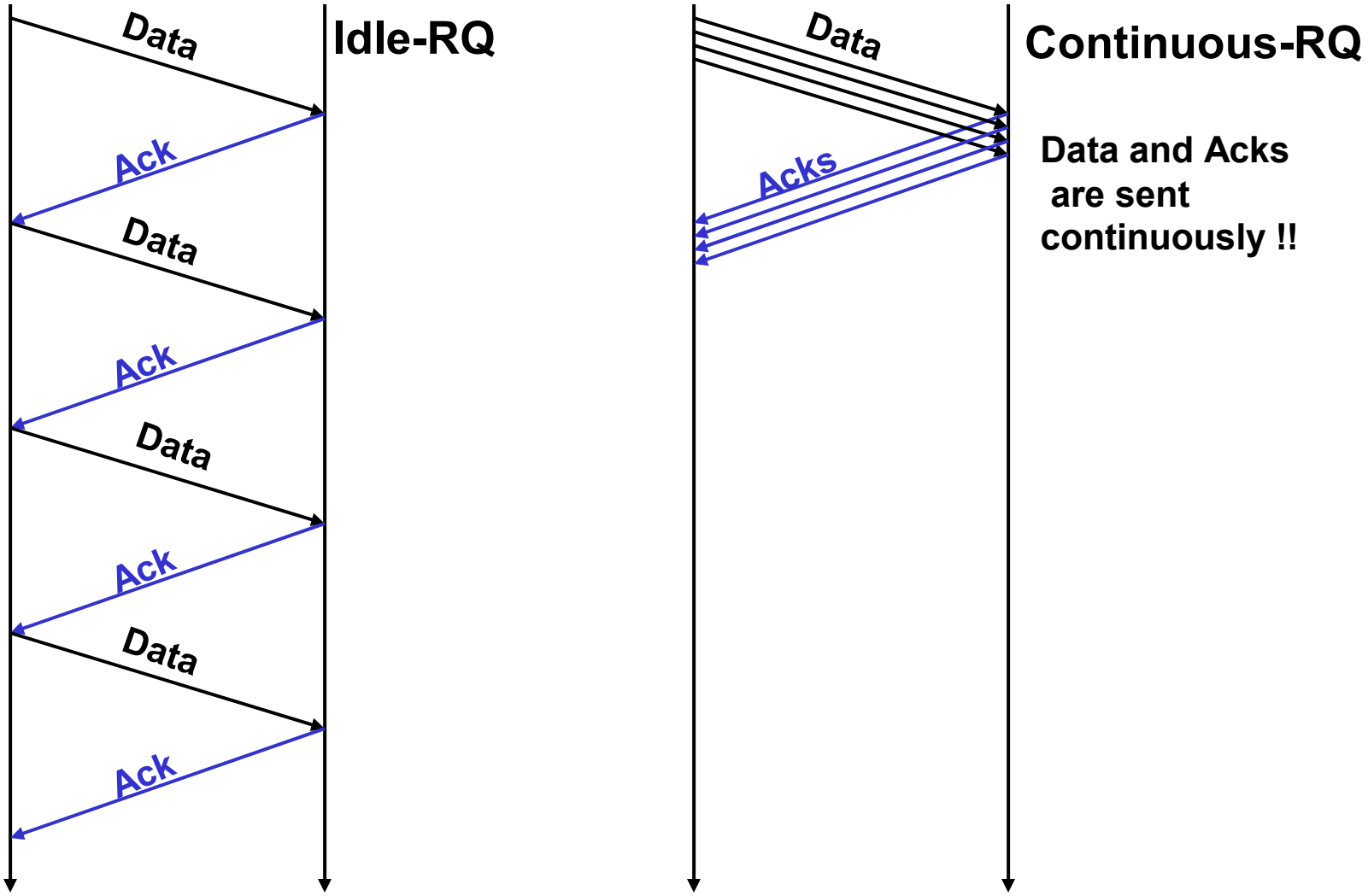
But stop-and-wait only allows one frame to be outstanding !!!

Assume MTU=1 KByte, then the max rate is
(1024 × 8) bit / 0.35 s
≅ 23 Kbit/s

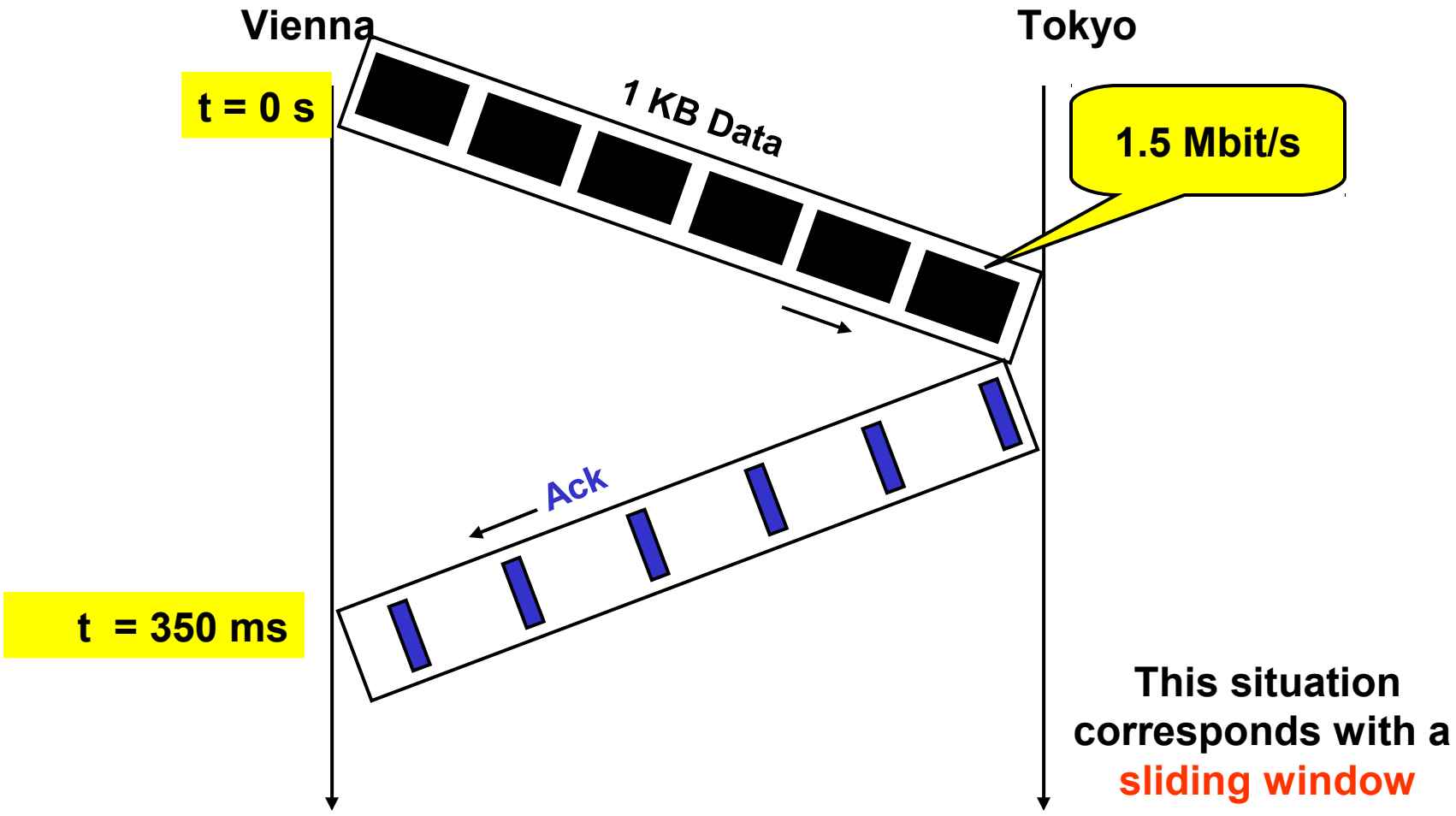
Idle-RQ Facts

- Old and slow method
  - But small code and only little resources necessary
- At least two sequence numbers necessary
  - To distinguish new from old data
- Half duplex protocol
- Example: TFTP

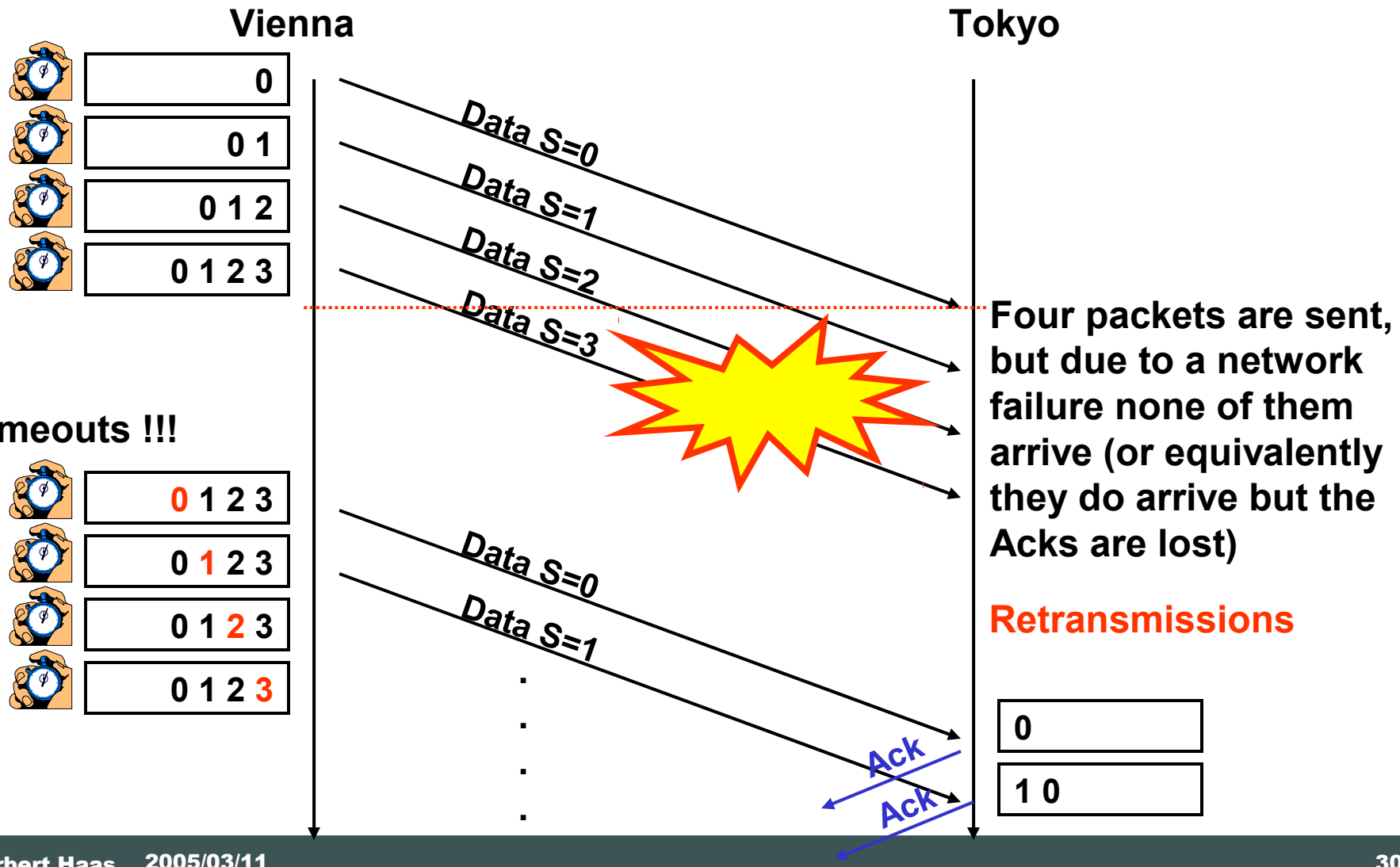
Continuous RQ



Full Pipe !


Vienna
Tokyo
t = 0 s
1 KB Data
1.5 Mbit/s
Ack
t = 350 ms
This situation corresponds with a sliding window

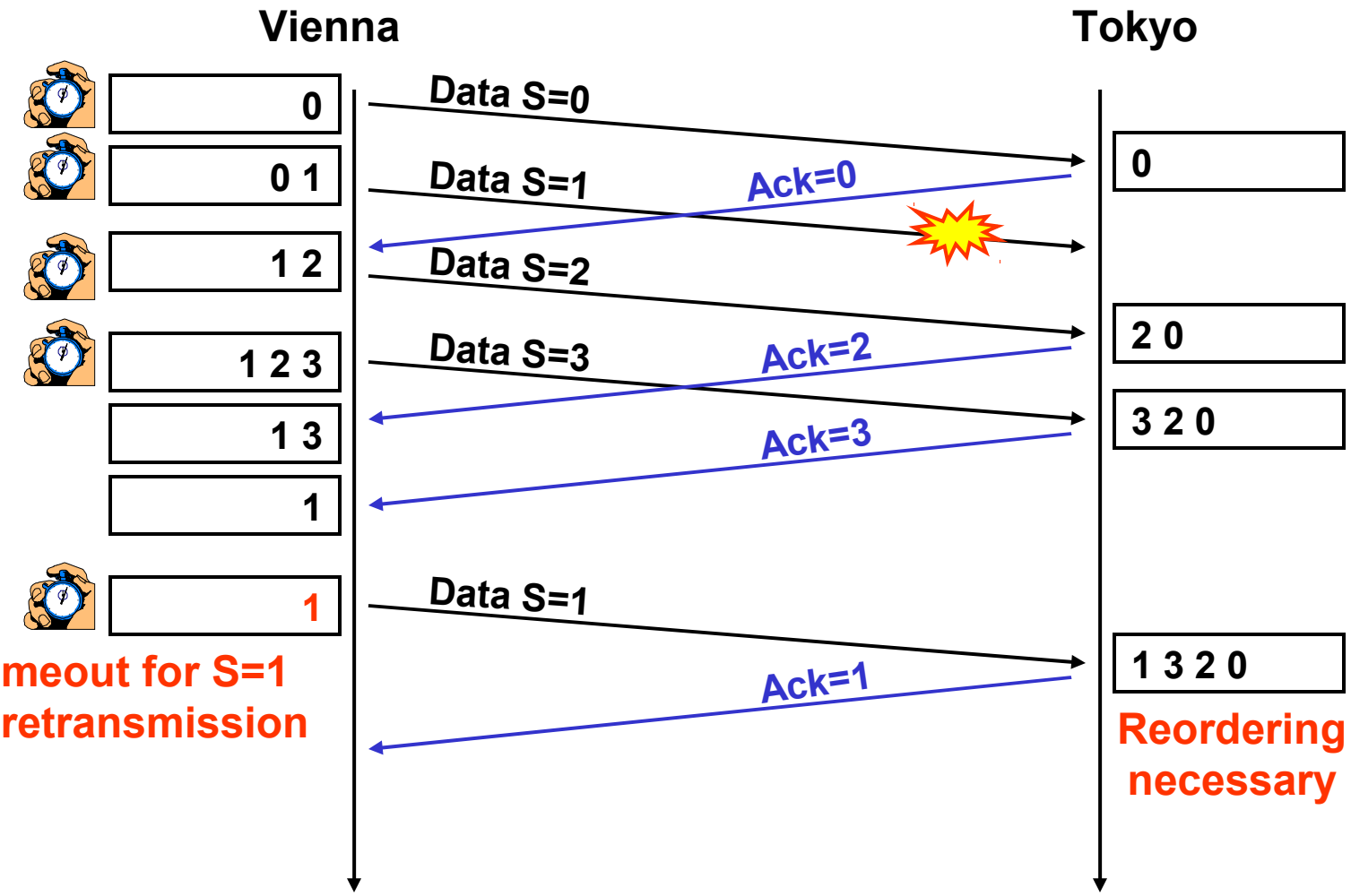
Need For Retransmission Buffer



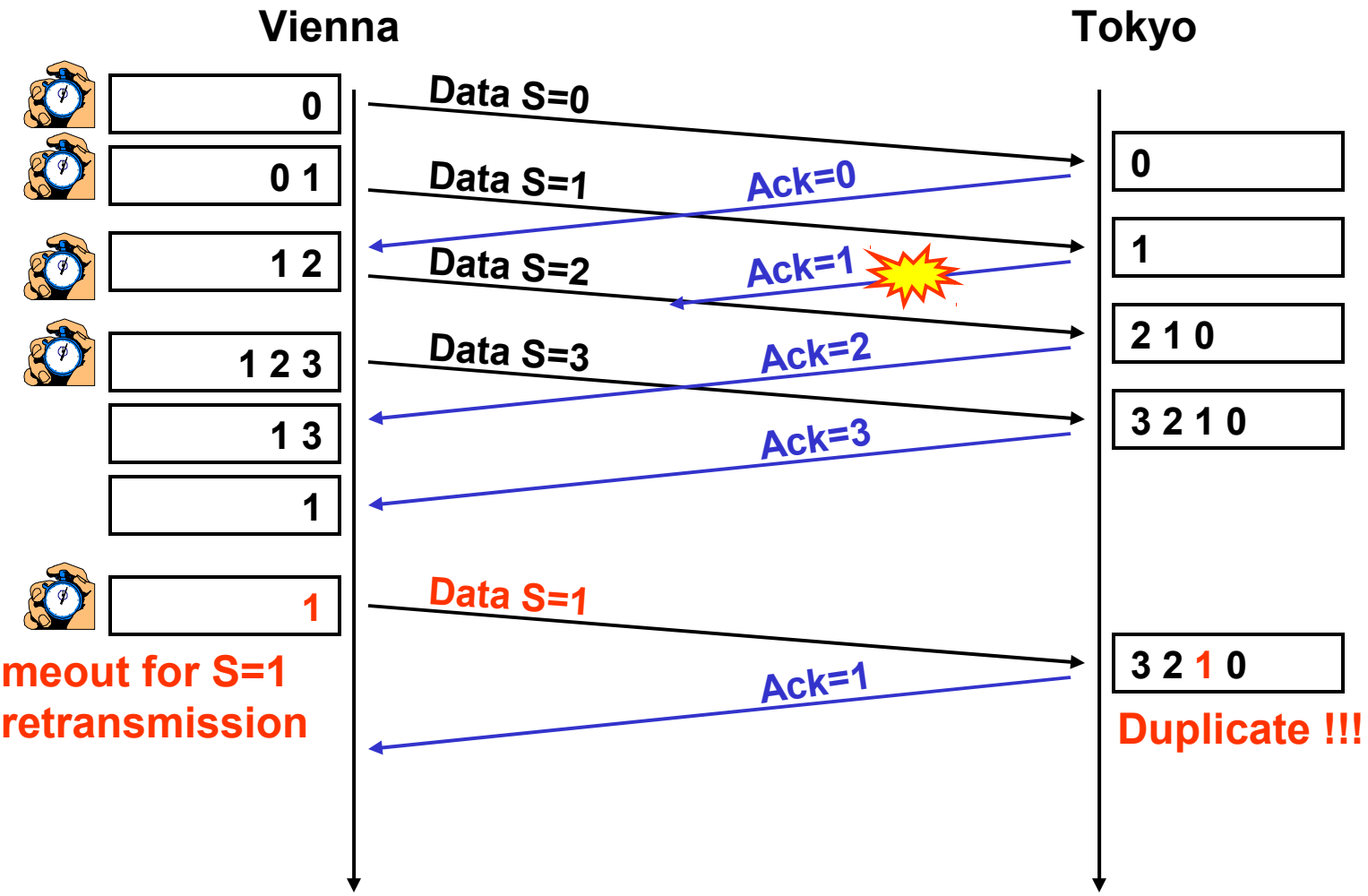
Continuous-RQ Resources

- Continuous-RQ requires ***dramatically*** more resources than Idle-RQ or connectionless protocols !!!
  - Retransmission Timers
  - Retransmission Buffers
  - Receive Buffers (to maintain sequence)
- Might result in high CPU loads !!!
  - Reason for DoS Attacks

Selective Acknowledgements



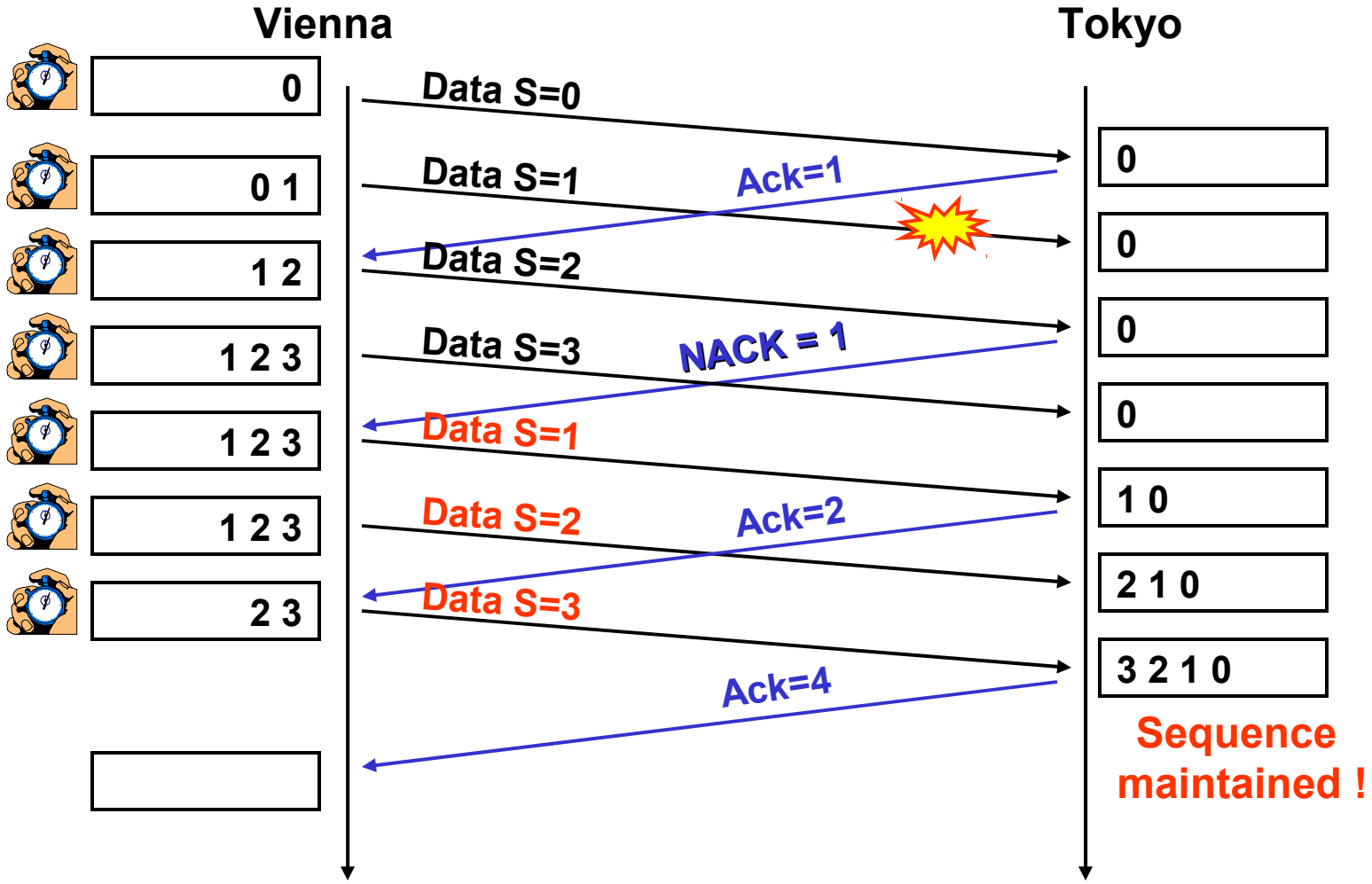
SACK: Duplicates



SACK

- Application:
  - New option for TCP to accomodate to long fat pipes with high BER
- Optionally, retransmissions might be sent immediately when unexpected (the next but one) ACK occurs
- Opposite idea: Cumulative ACK

GoBack N


Vienna
Tokyo
0
0 1
1 2
1 2 3
1 2 3
1 2 3
2 3
Data S=0
Data S=1
Ack=1
Data S=2
Data S=3
NACK = 1
Data S=1
Data S=2
Ack=2
Data S=3
Ack=4
0
0
0
0
1 0
2 1 0
3 2 1 0
Sequence maintained !

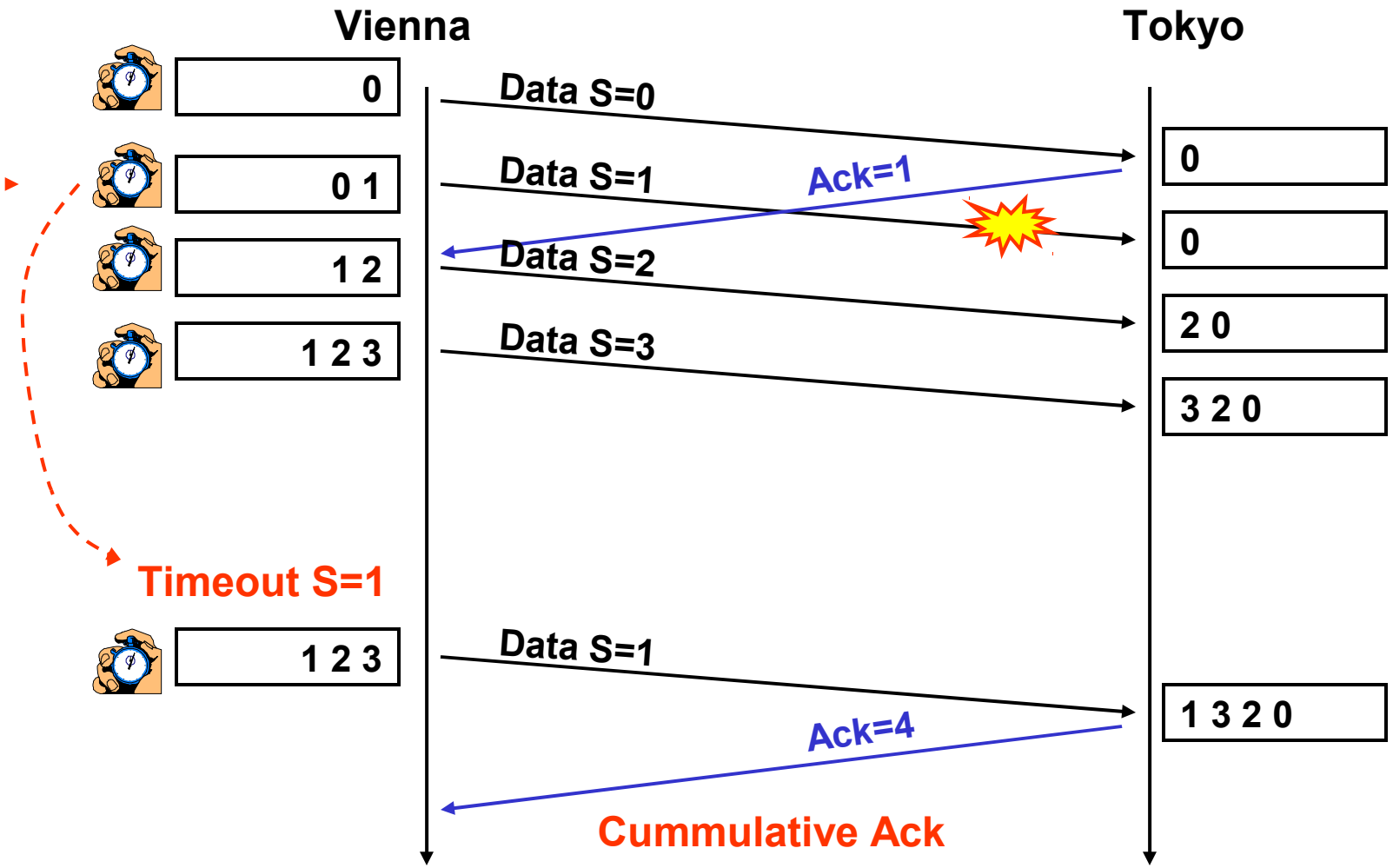
GoBack N – Facts

- Maintains order at receiver-buffer
  - Reordering was too much time-consuming in earlier days
- Still used by
  - HDLC and clones ("REJECT")
  - TCP
    - Variant known as "fast retransmit"
    - Uses duplicate acks as NACK

Selective Reject ARQ

- Modern modification of GoBack N
- Only those frames are retransmitted that receive a NACK
  - Or those that time out
- Receiver must be able to reorder frames
- Application:
  - Optional for modern HDLC clones

Positive Ack



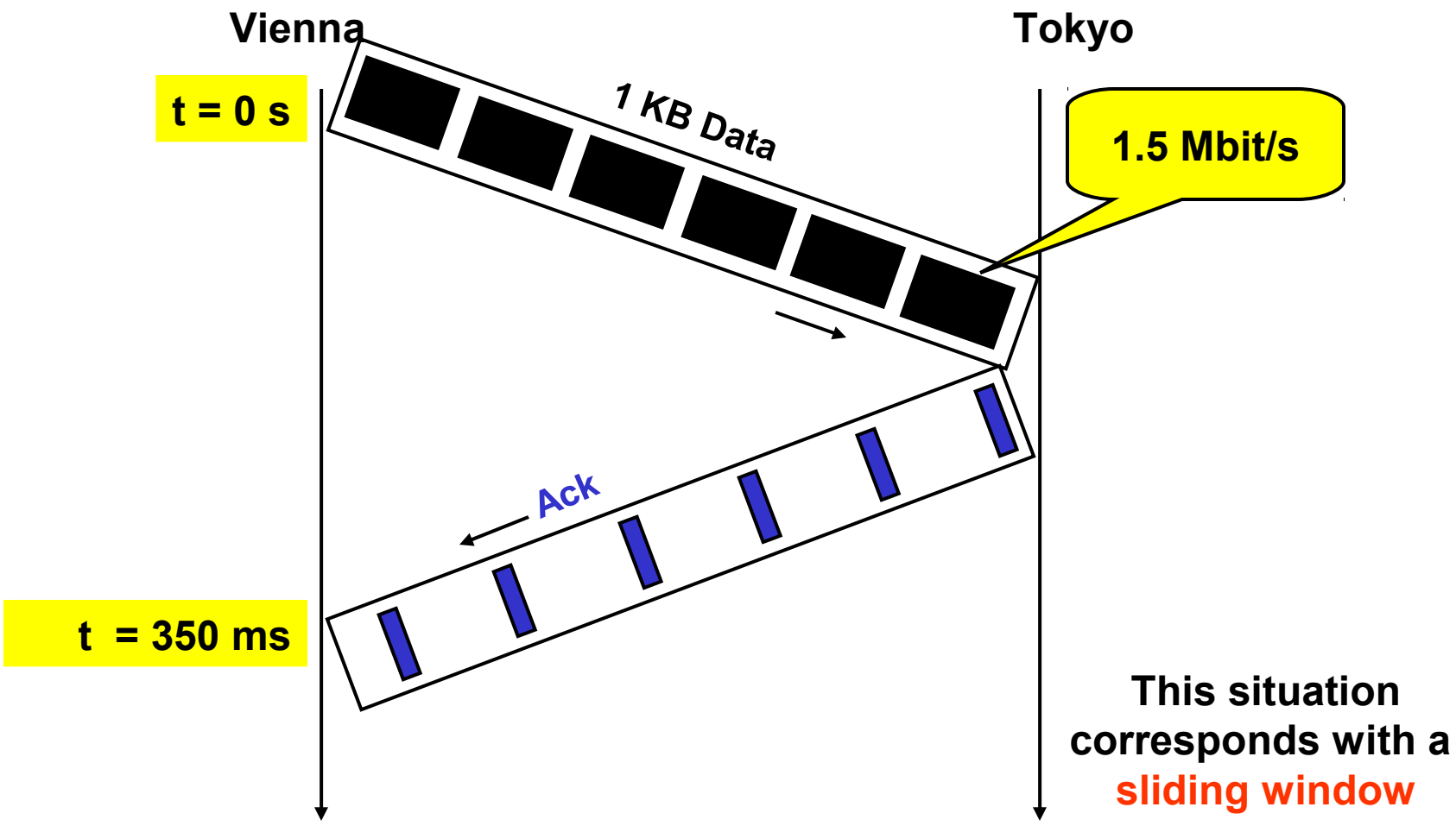
Positive Ack – Facts

- Always together with cumulative acks
  - Any frame received is buffered
  - Receiver must be able to reorder
- Problem:
  - Only timeouts trigger retransmission
- Application:
  - Early TCP

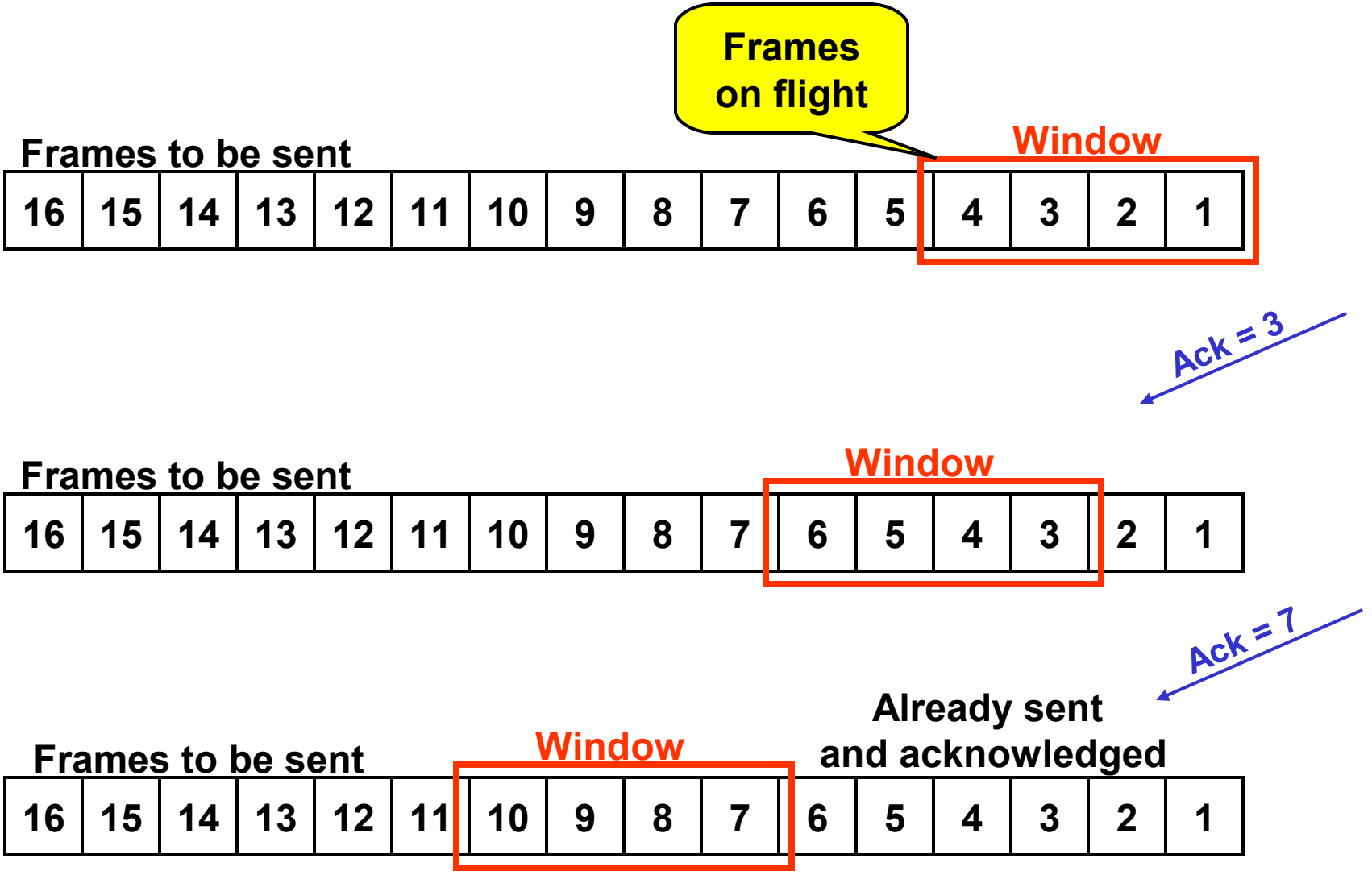
Windowing

- As shown, sender must buffer unacknowledged frames in case for retransmissions
- Necessary sender-buffer size is called "window"
- Window size depends on
  - Bandwidth of channel
  - Round-Trip-Time (RTT)

Remember: Full Pipe !



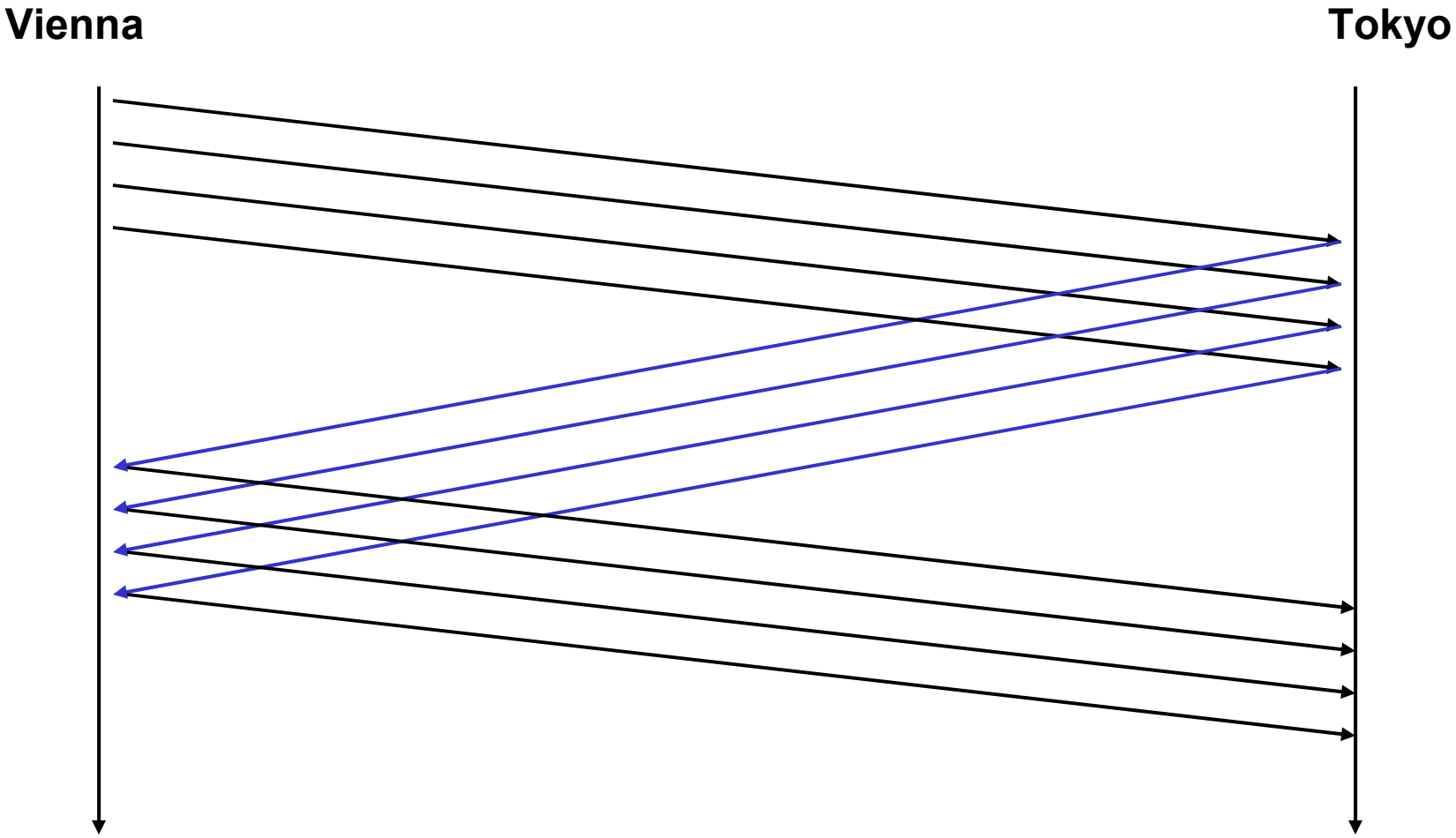
Sliding Window Basics (1)



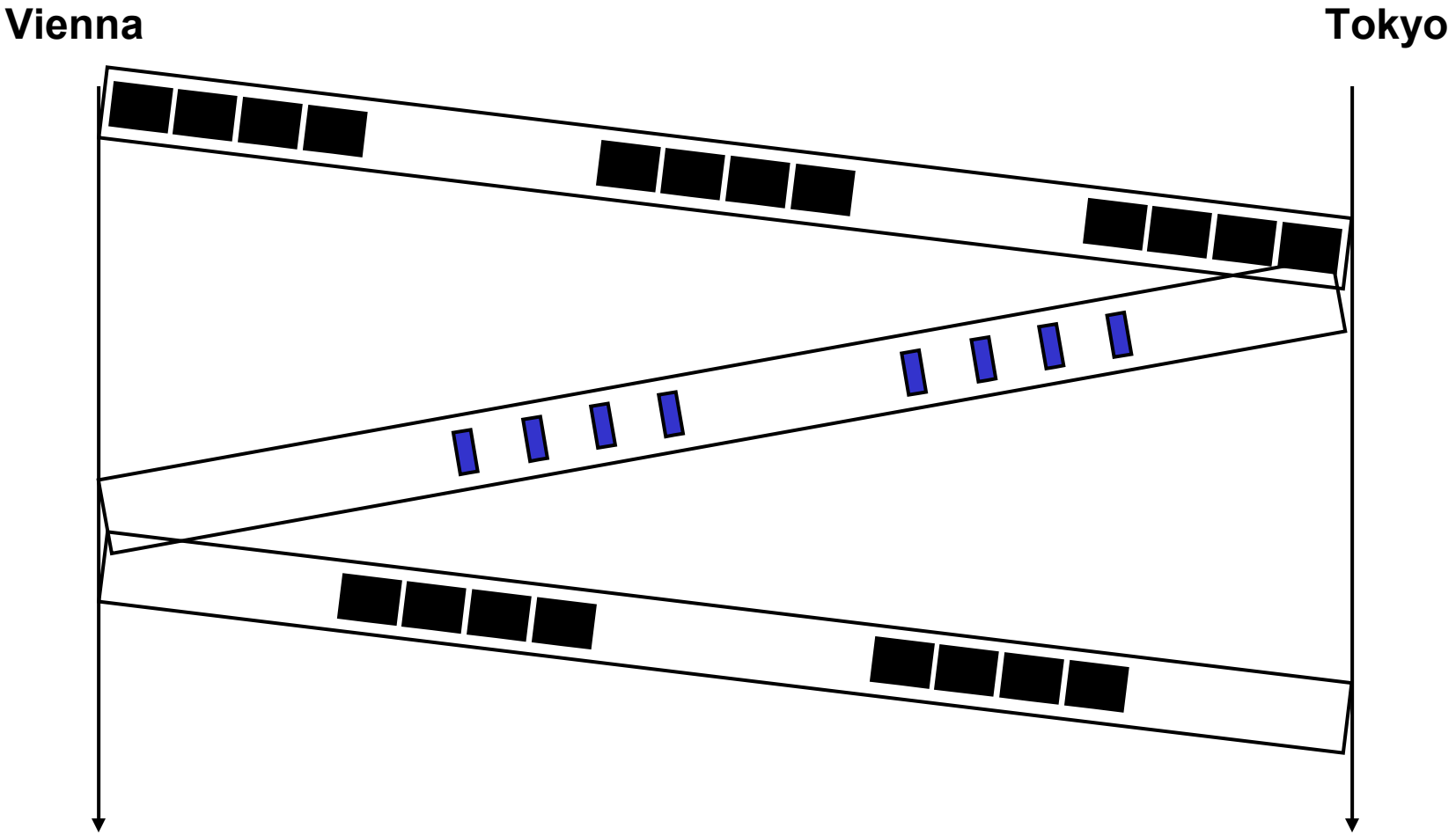
Sliding Window Basics (2)

- Window Size in Bytes = BW × RTT
  - If smaller: jumping window
  - Extreme case: Idle-RQ for W=1
- How many Identifiers?
  - At least W+1
    - If all W frames must be retransmitted, receiver must distinguish from new data
  - W < (MaxSeqNum+1) /2
    - To avoid troubles on wrap around

Jumping Window



Jumping Window



Flow Control

- Too large window sizes
  - Might require too much retransmissions (especially with GoBackN)
  - Result in network congestion (imagine thousands of users)
  - Receiver buffer overflow
- Flow control #1: Adaptive Windowing
- Flow control #2: Stop and Go

Flow Control

- Adaptive Windowing
  - The reciever adjusts the sender's window size (sent together with ack)
  - TCP's approach
- Stop and Go
  - Dedicated flow control frames
  - HDLC's approach (RR and RNR)
  - Ethernet's approach (Pause-Frame)

Medium Access

- In case of shared media
  - Collisions possible
  - Who may send?
- Basic Techniques
  - Aloha (Ethernet Principle "CSMA/CD")
  - Token (Token Ring, FDDI, Token Bus)
  - Polling (IEEE 802.12)
  - Time Slices (GSM)
- Will be discussed later together with these related technologies

Summary

- Most link layer protocols utilize CRC for frame protection
- ARQ Techniques: Idle-RQ, GoBackN, Selective-Ack, Positive-Ack
  - Additionally Cumulative-Ack possible
- Only Continuous-RQ fills pipe
- Flow control
  - Either by controlling window size
  - Or deciated stop and go messages

“ If a packet hits a pocket on a socket on a port,
And the bus is interrupted as a very last resort,
And the address of the memory makes your floppy disk abort,
Then the socket packet pocket has an error to report!
If your cursor finds a menu item followed by a dash,
And the double-clicking icon puts your window in the trash,
And your data is corrupted 'cause the index doesn't hash,
then your situation's hopeless, and your system's gonna crash!
If the label on the cable on the table at your house,
Says the network is connected to the button on your mouse,
But your packets want to tunnel on another protocol,
That's repeatedly rejected by the printer down the hall,
And your screen is all distorted by the side effects of gauss,
So your icons in the window are as wavy as a souse,
When the copy of your floppy's getting sloppy on the disk,
And the microcode instructions cause unnecessary risc,
Then you have to flash your memory and you'll want to ram your rom.
Quickly turn off the computer and be sure to tell your mom! ”

Quiz

- What's the problem when putting IP-packets directly onto ISDN?
- What are Hamming-Codes?
- What maximum bit-rate can TCP-hosts utilize when connected via satellite?
- Explain why windowing protocols are more prone to DoS-attacks