ARQ, Sliding Window, Ethernet

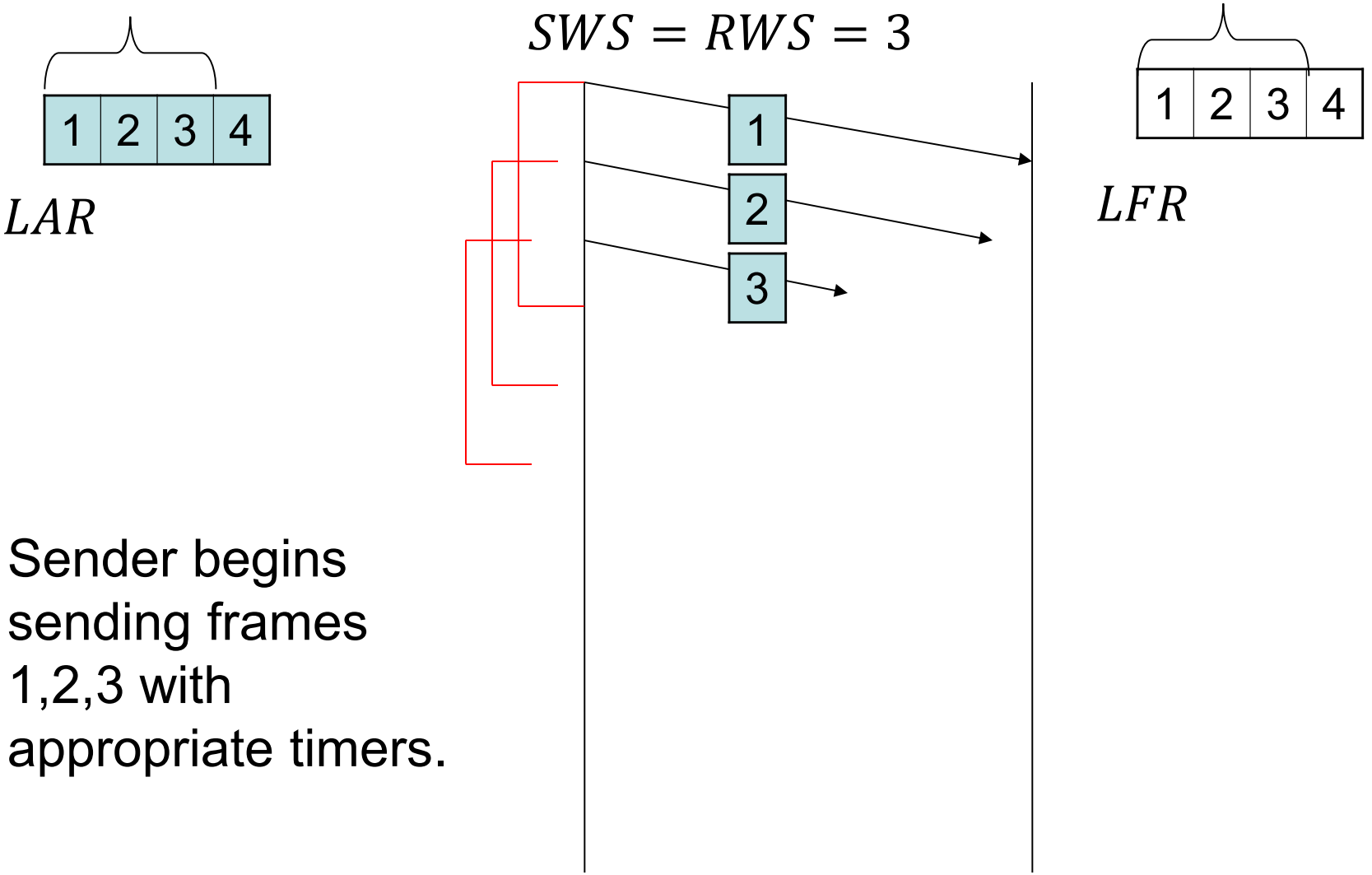
16 November 2025
Lecture 4

Topics for Today

- ARQ
  - Sliding Window
- Ethernet

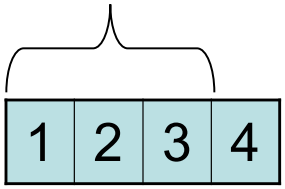
- Source: Peterson and Davie 2.1-2.5, 2.6, Tanenbaum 4.3

Example Sliding Window Protocol



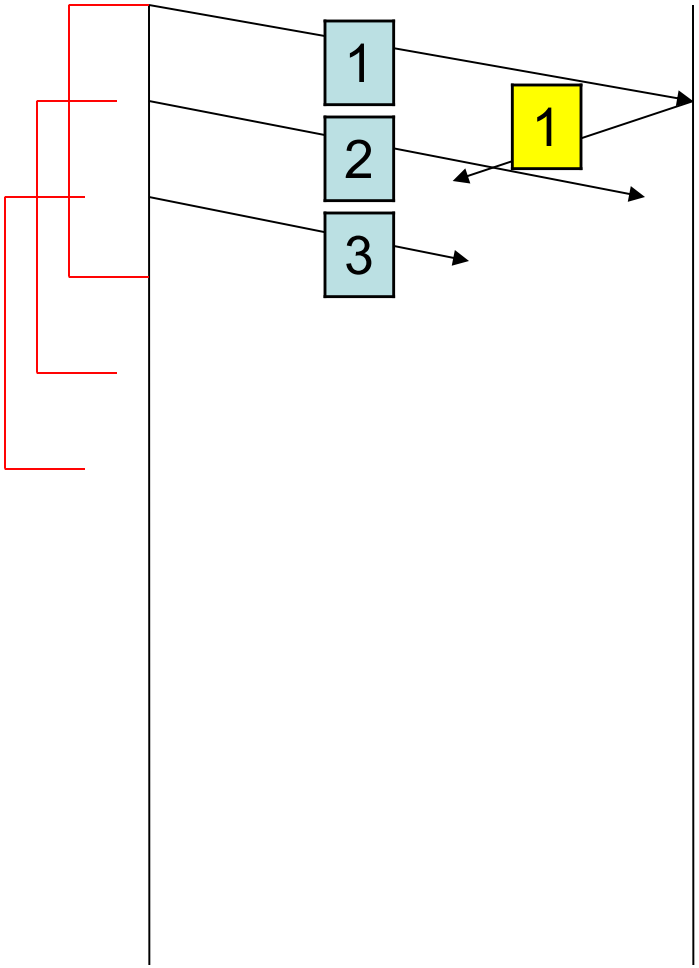
Sender begins sending frames 1,2,3 with appropriate timers.

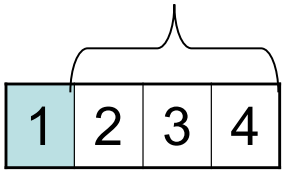
Example Sliding Window Protocol



LAR

$SWS = RWS = 3$





LFR

Receiver gets frame 1

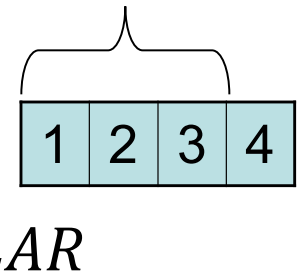
$SeqNumToAck = 1$

Receiver sends ACK(1)

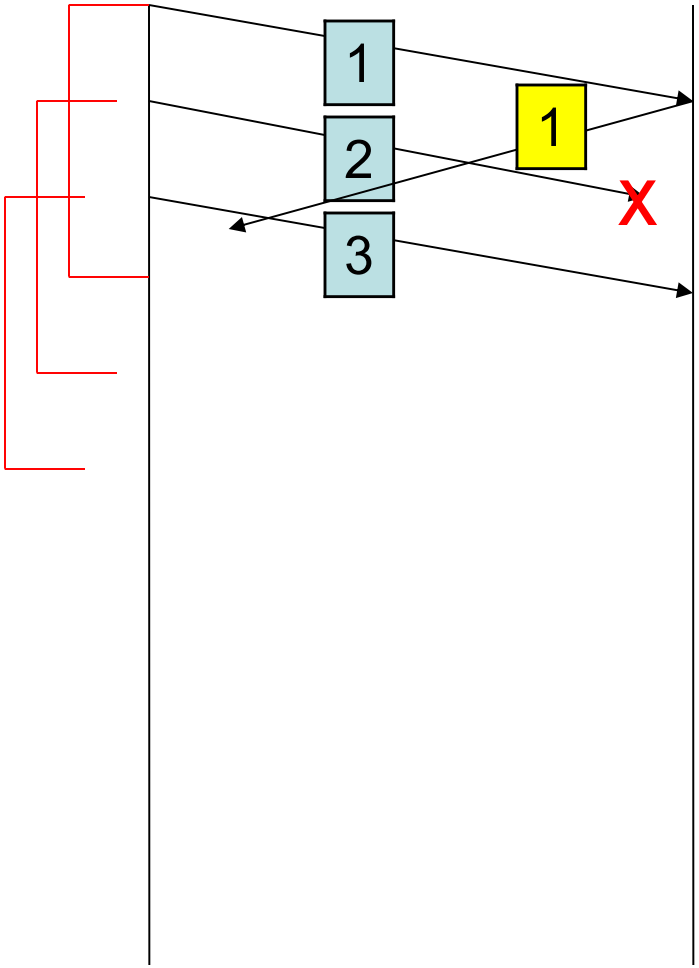
$$LFR = 1$$

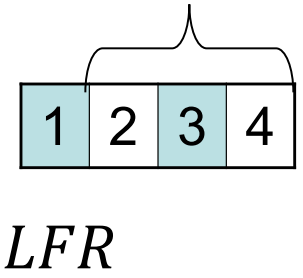
$$LAF = LFR + RWS$$

Example Sliding Window Protocol



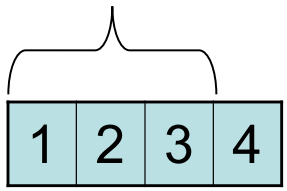
$SWS = RWS = 3$





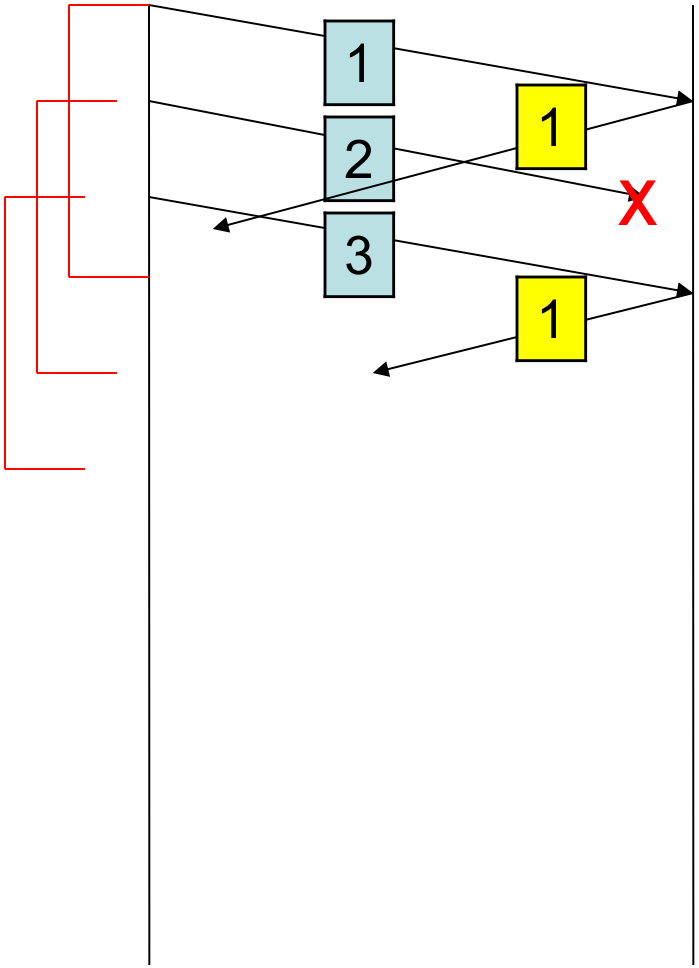
While ACK(1) is in transit, frame 2 is lost and frame 3 is accepted.

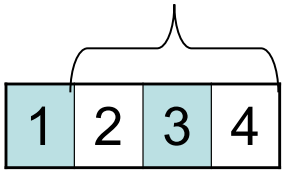
Example Sliding Window Protocol



LAR

$SWS = RWS = 3$



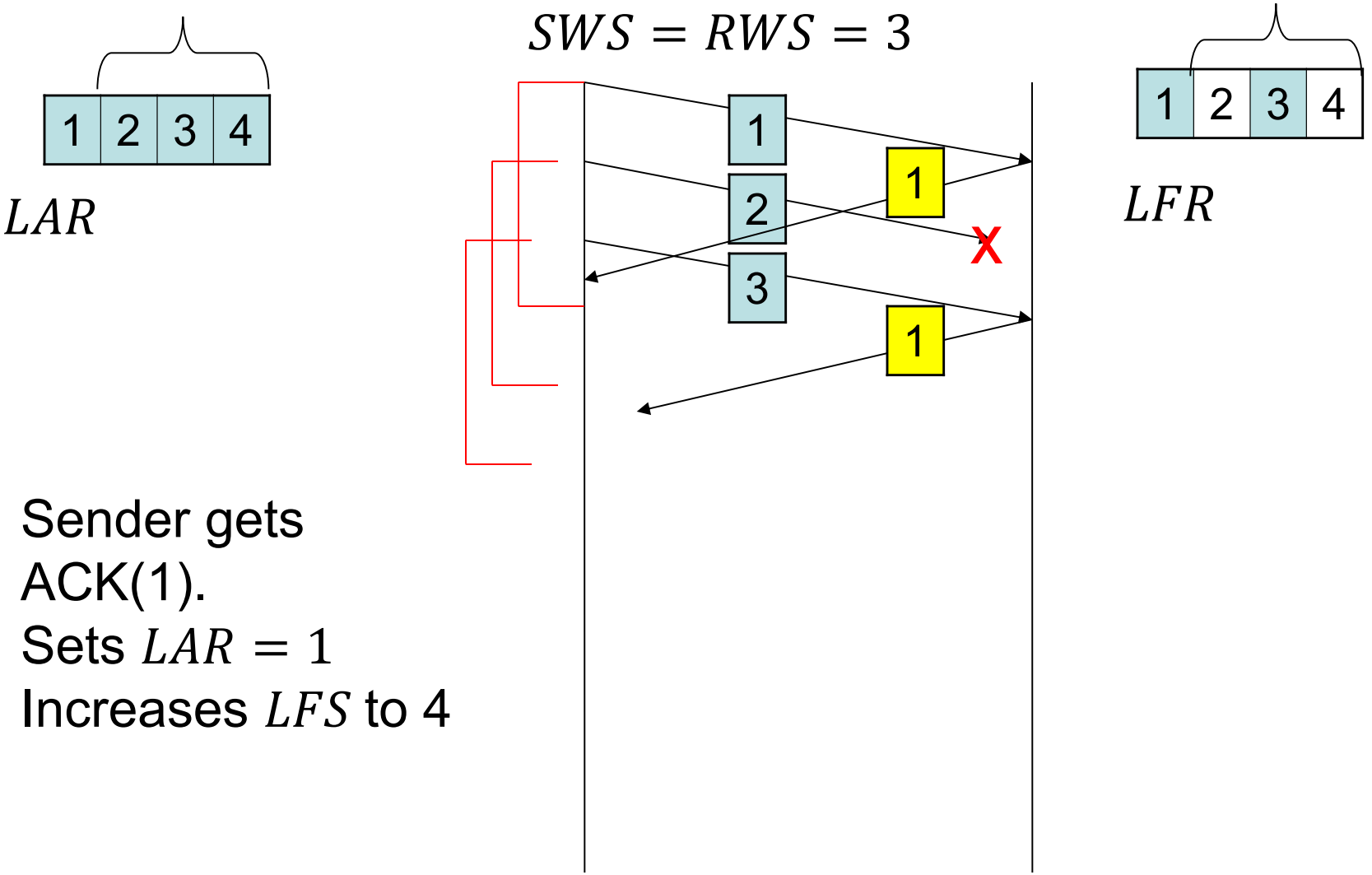


LFR

$SeqNumToAck = 1$

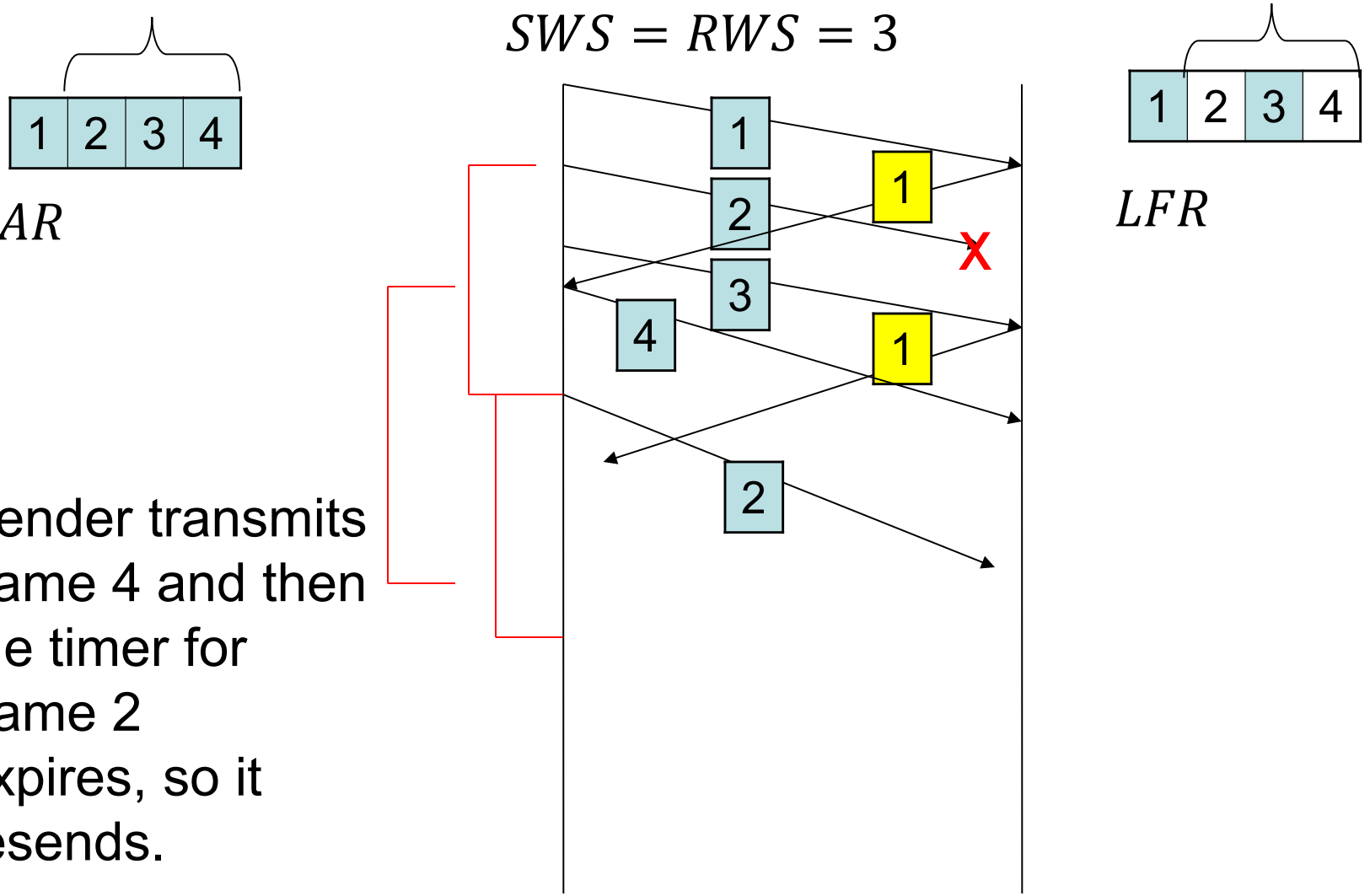
Receiver sends another Ack(1) message.

Example Sliding Window Protocol



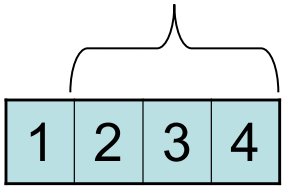
Sender gets ACK(1).
Sets $LAR = 1$
Increases LFS to 4

Example Sliding Window Protocol



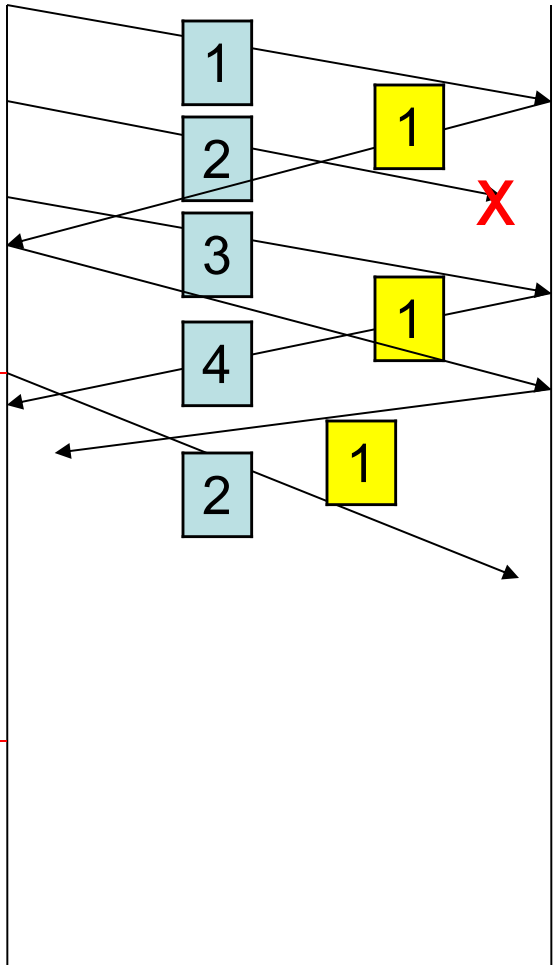
Sender transmits frame 4 and then the timer for frame 2 expires, so it resends.

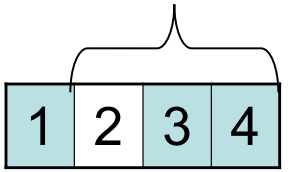
Example Sliding Window Protocol



LAR

SWS = RWS = 3





LFR

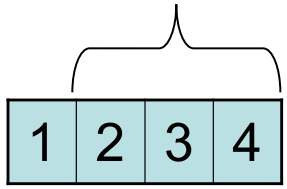
Sender gets ACK(1) again—ignores it.

Receiver gets frame 4

$SeqNumToAck = 1$

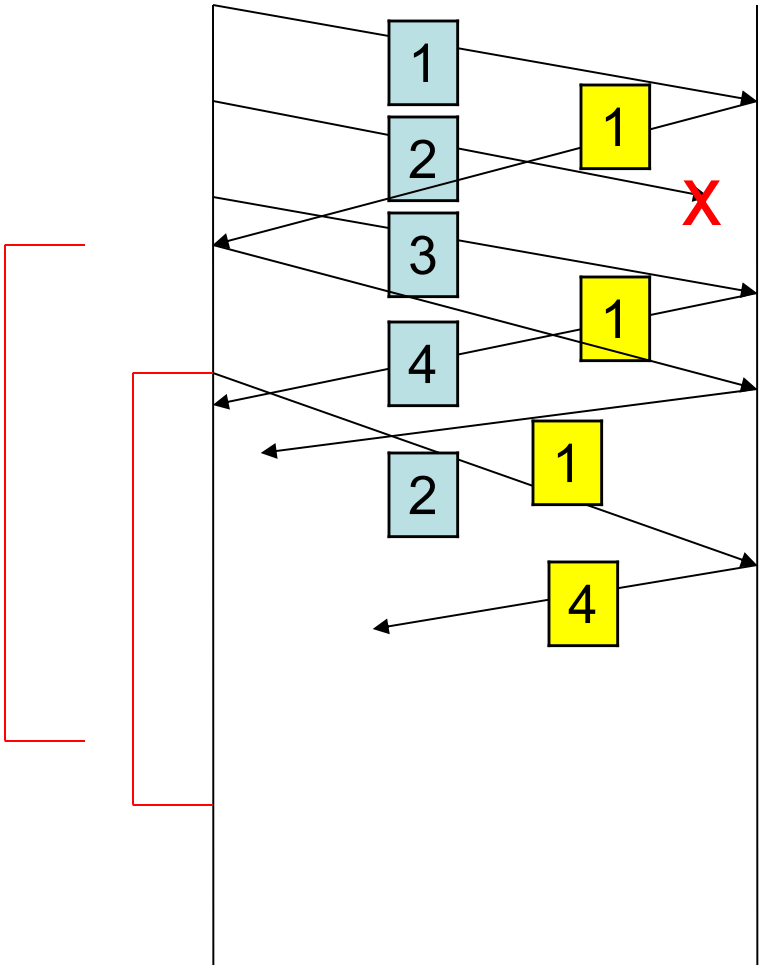
Receiver sends ACK(1)

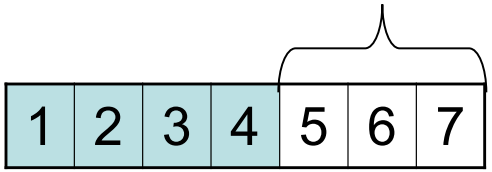
Example Sliding Window Protocol



LAR

$SWS = RWS = 3$





LFR

Receiver gets frame 2

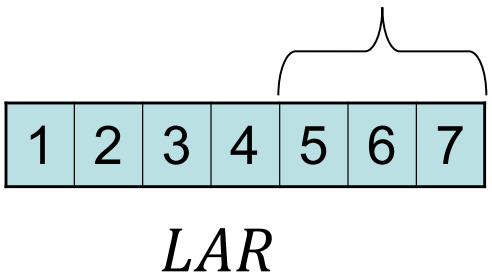
$SeqNumToAck = 4$

Receiver sends ACK(4)

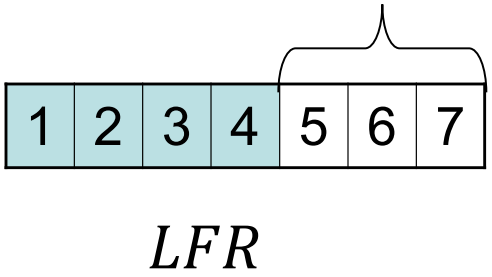
$LFR = 4$

$LAF = LFR + RWS$

Example Sliding Window Protocol

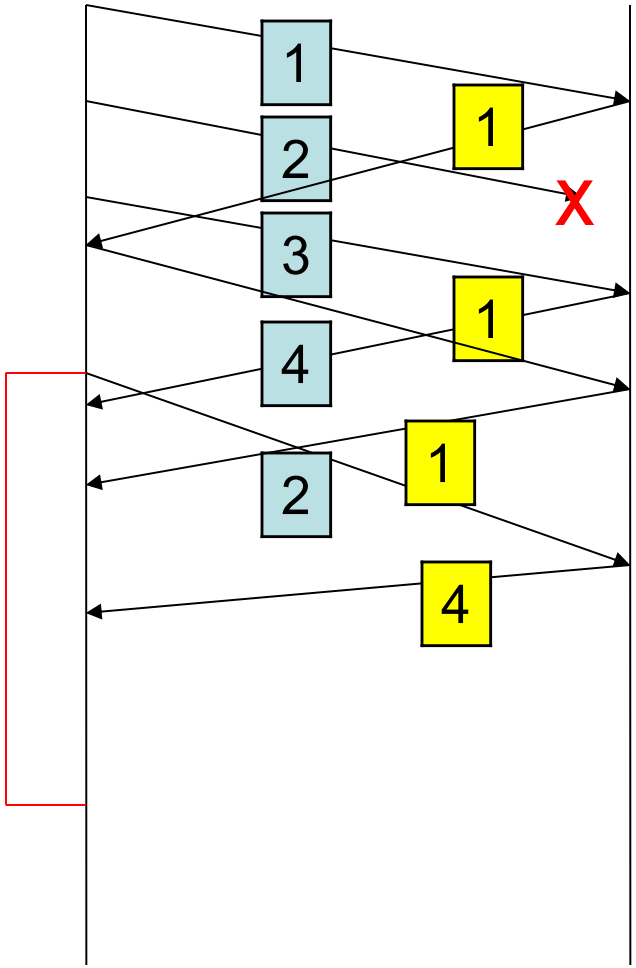


$SWS = RWS = 3$



Sender gets ACK(1) again—ignores it.

Sender gets ACK(4)
Sets $LAR = 4$
Increases LFS



Variants on Sliding Window

Receiver doesn't transmit redundant ACKs

– Just ignore out of order arrivals

Receiver transmits *selective ACKS*

– ACK indicates exactly which frames have been accepted

What size for the window?

- If $RTT \times Bandwidth$ product known, ideal is:

$$SWS = \frac{RTT}{2} \times \frac{Bandwidth}{Framesize}$$

- Common receive window size settings:

$RWS = 1$	$RWS = SWS$	Note: $RWS > SWS$ is not sensible
• No buffering of out-of-order frames	• Buffer as many as can be in flight	

Finite Sequence Numbers

We've assumed infinite sequence numbers so far.

Real packets have finite size "Sequence Number" field

What do we do?

Image source: reddit.com

What's a sufficient SeqNum Field size?

Principle: Re-use sequence numbers

- 8-bit example
- They wrap: 0,1, 2, …, 254, 255, 0,1, 2, …, 254, 255, 0,1, 2, …

Recall:

- For Stop-and-Wait we need 2 sequence values (0/1)
- 1 bit of space

What about for Sliding Window with X packets?

Suppose $SWS = RWS$

- How many sequence numbers should there be?
- Is $SWS + 1$ sufficient?

Sufficient MaxSeqNum

- Frame i's sequence num is $i \ mod \ MaxSeqNum$

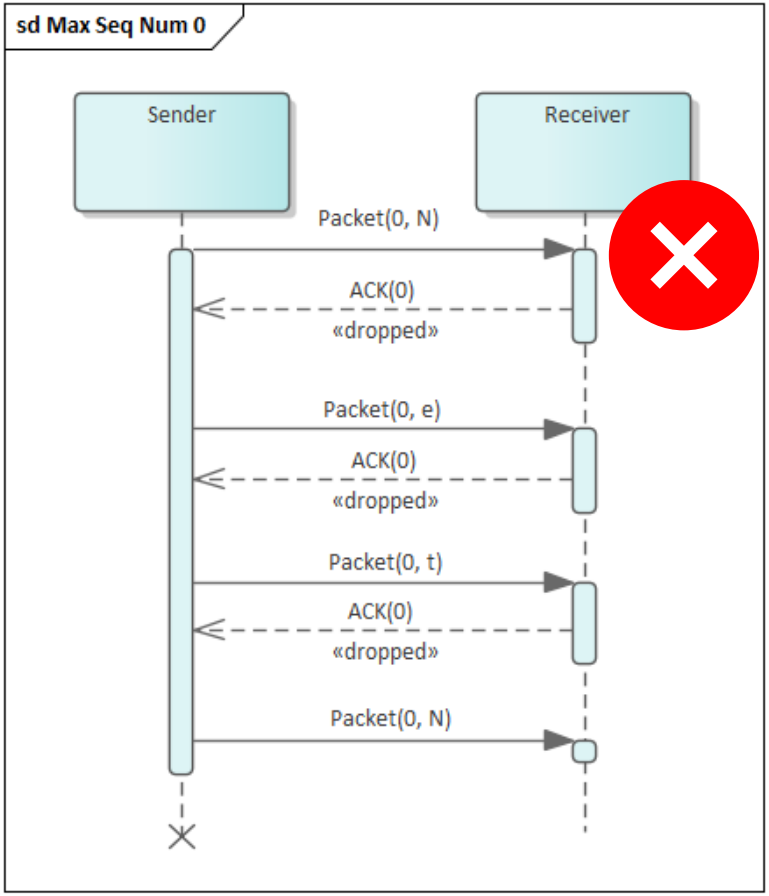
- Assuming $SWS = RWS$
- $SWS < \frac{MaxSeqNum+1}{2}$
- $MaxSeqNum > (2 \times SWS) - 1$

- Why?
  - Consider case where all the ACKS are lost.
  - Suppose $SWS = RWS = 3$
  - $MaxSeqNum \geq 5$ since (0,1,2,3,4) are insufficient

Sequence Number Worst Case

0	0	0	0	0	0	0	0
N	e	t	w	o	r	k	s

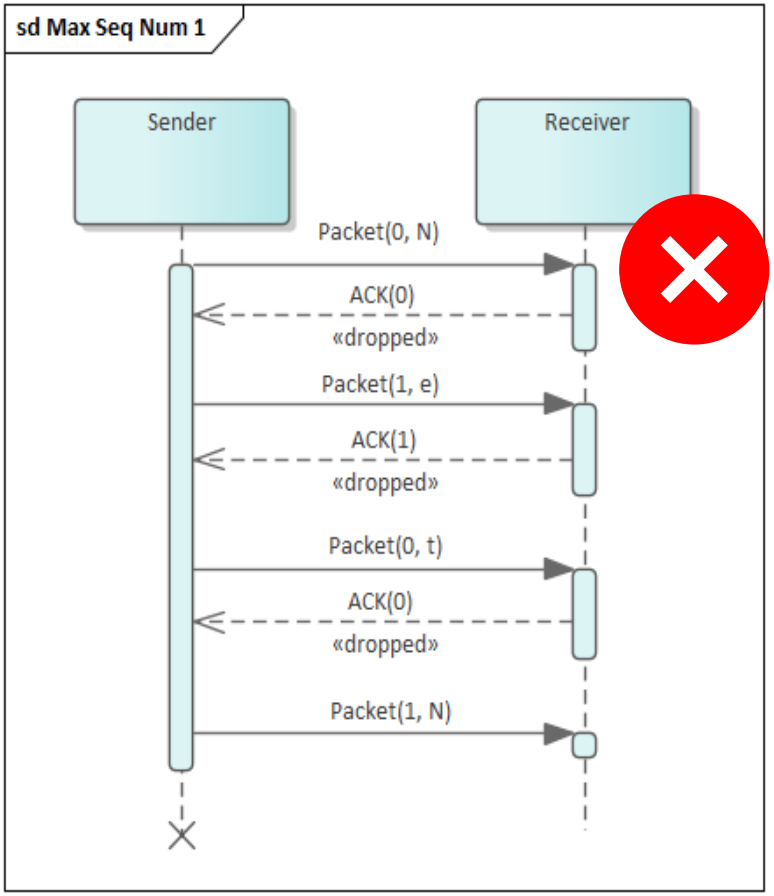
0	0	0	0	0	0	0	0



Sequence Number Worst Case

0	1	0	1	0	1	0	1
N	e	t	w	o	r	k	s

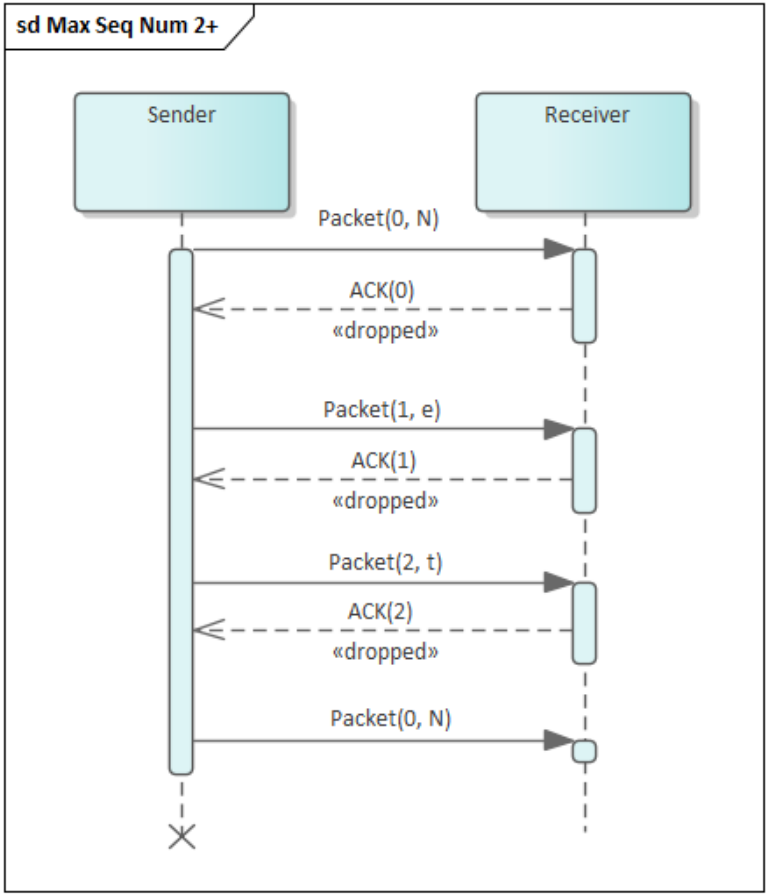
0	1	0	1	0	1	0	1



Sequence Number Worst Case

0	1	2	0	1	2	0	1
N	e	t	w	o	r	k	s

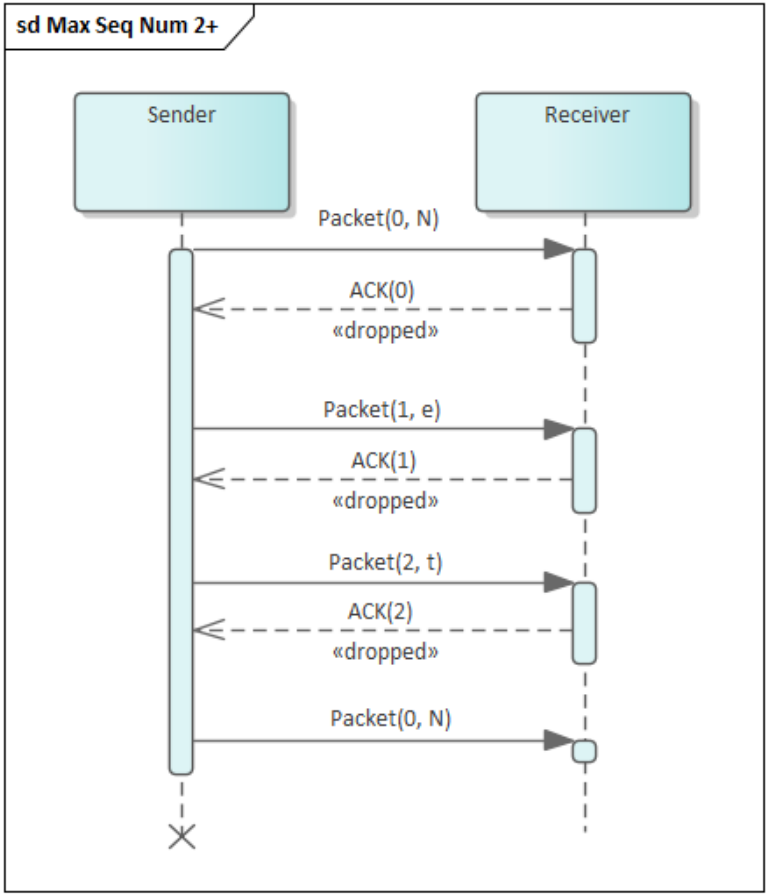
0	1	2	0	1	2	0	1



Sequence Number Worst Case

0	1	2	0	1	2	0	1
N	e	t	w	o	r	k	s

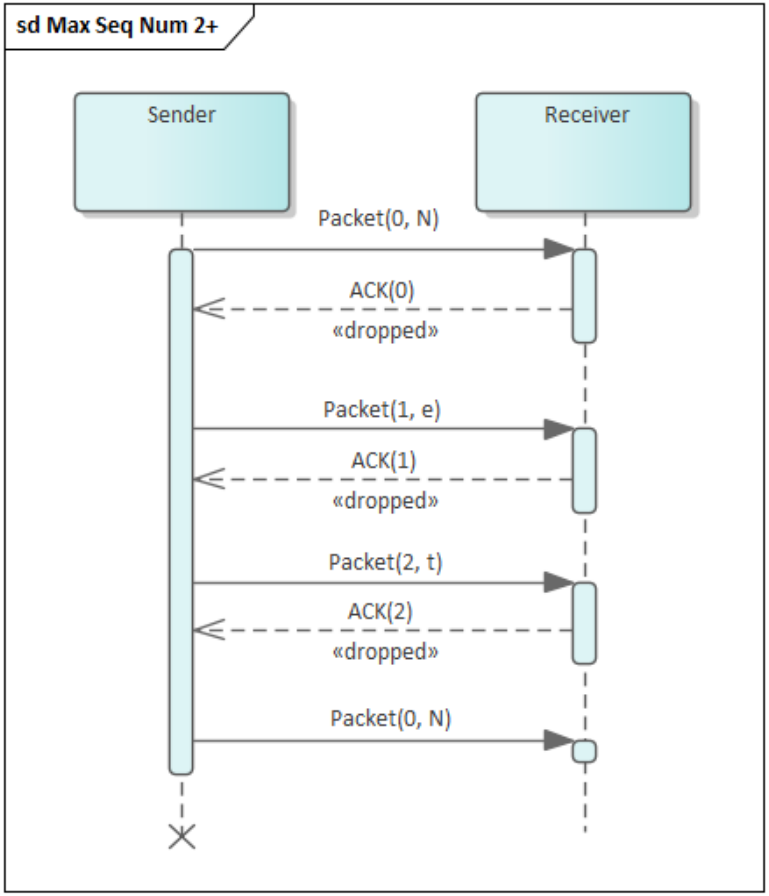
0	1	2	0	1	2	0	1
N							



Sequence Number Worst Case

0	1	2	0	1	2	0	1
N	e	t	w	o	r	k	s

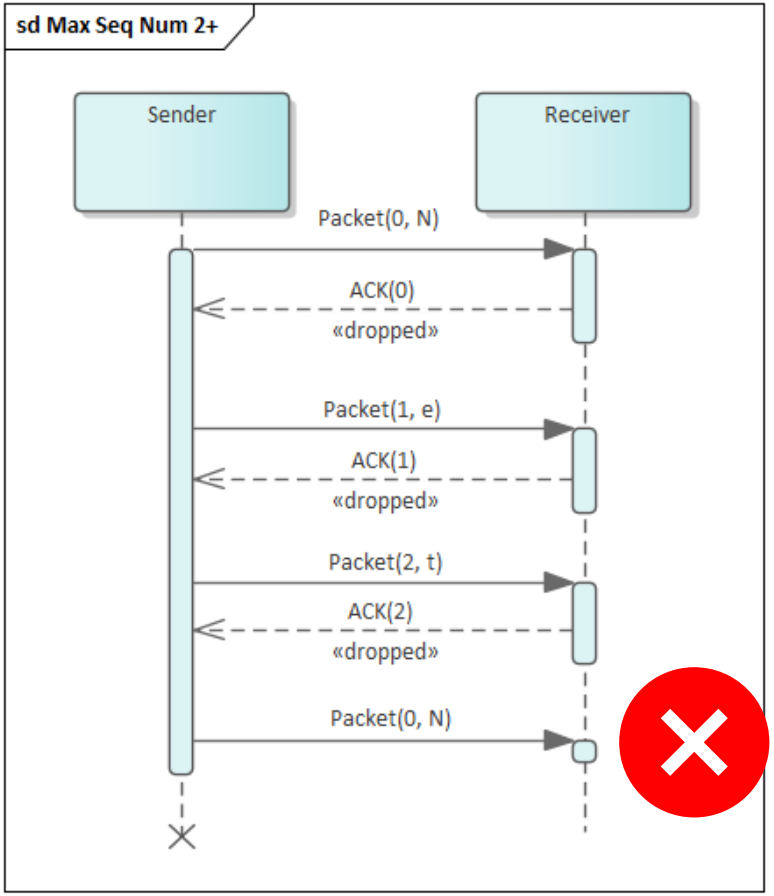
0	1	2	0	1	2	0	1
N	e						



Sequence Number Worst Case

0	1	2	0	1	2	0	1
N	e	t	w	o	r	k	s

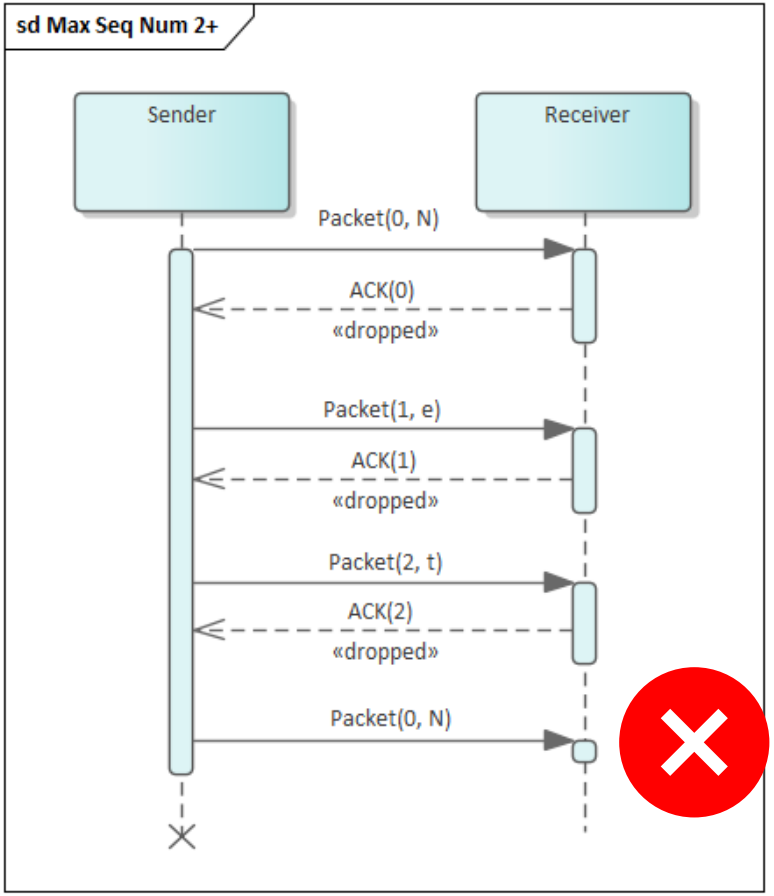
0	1	2	0	1	2	0	1
N	e	t					



Sequence Number Worst Case

0	1	2	3	0	1	2	3
N	e	t	w	o	r	k	s

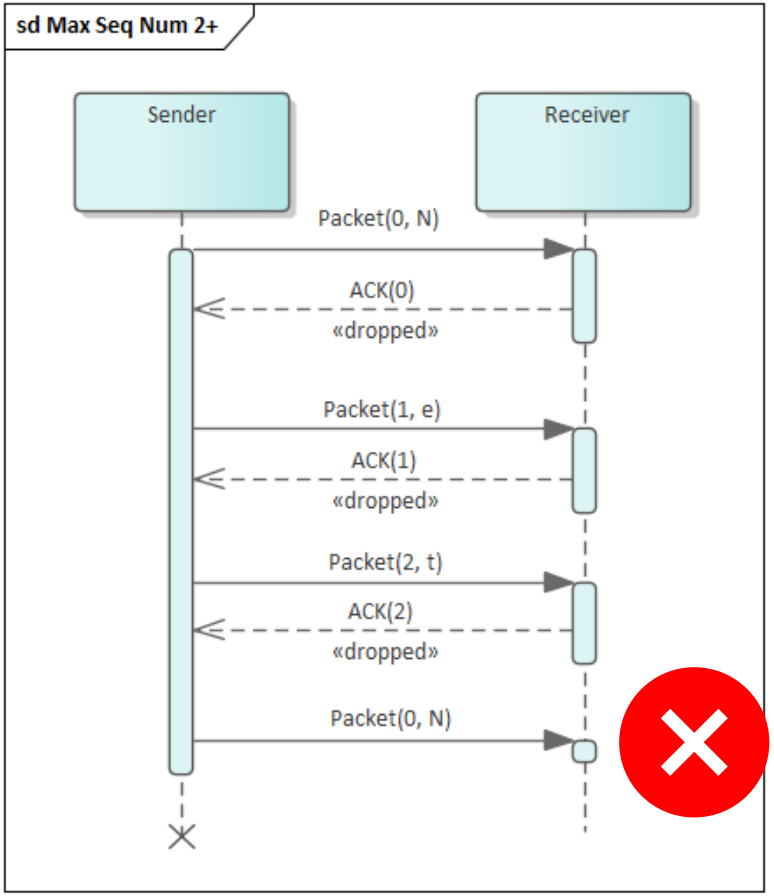
0	1	2	3	0	1	2	3
N	e	t					



Sequence Number Worst Case

0	1	2	3	4	0	1	2
N	e	t	w	o	r	k	s

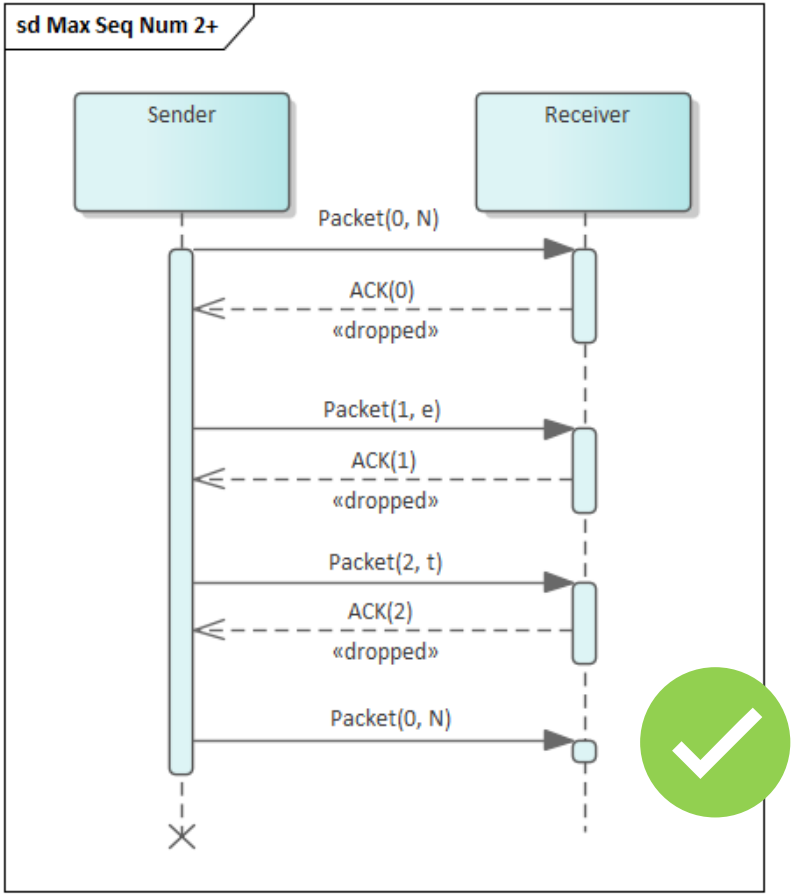
0	1	2	3	4	0	1	2
N	e	t					



Sequence Number Worst Case

0	1	2	3	4	5	0	1
N	e	t	w	o	r	k	s

0	1	2	3	4	5	0	1
N	e	t					



We're doing this backward

Max Sequence Number determined by the protocol design

- Fixed field size

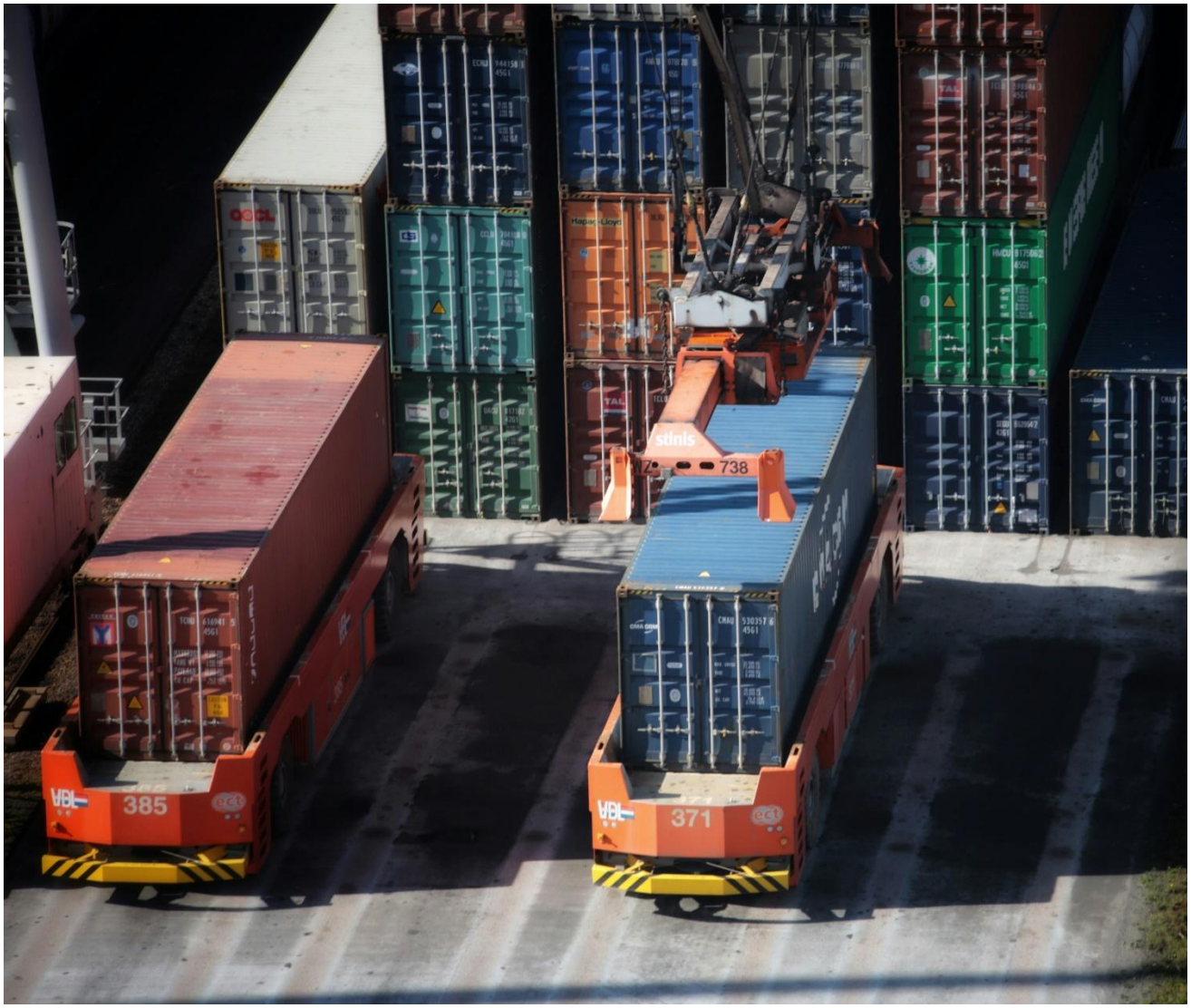
Must select SWS and RWS accordingly

- $SWS \approx 0.5 * MaxSeqNum$
- $RWS \approx 0.5 * MaxSeqNum$
- $RWS \leq SWS$

Sequence numbers wrap fast

- Imagine a 100Gbps connection
- 12.5 GB per second
- 32 bit sequence number field ~4 billion possibilities
- Number wraps ~3 times per second

Photo by Bernd 📷 Dittrich on Unsplash



Roles of Sliding Window Algorithm

Reliable delivery

- It provides an efficient retransmission protocol for dealing with errors

In-order delivery

- The receiver buffers frames and delivers them in sequence number order

Flow control

- It sends ACKs back to give hints to sender
- More sophisticated version could give # of frames the receiver has room for → throttles the sender

Sliding window in practice

TCP (Transmission Control Protocol)

Uses sliding window algorithm

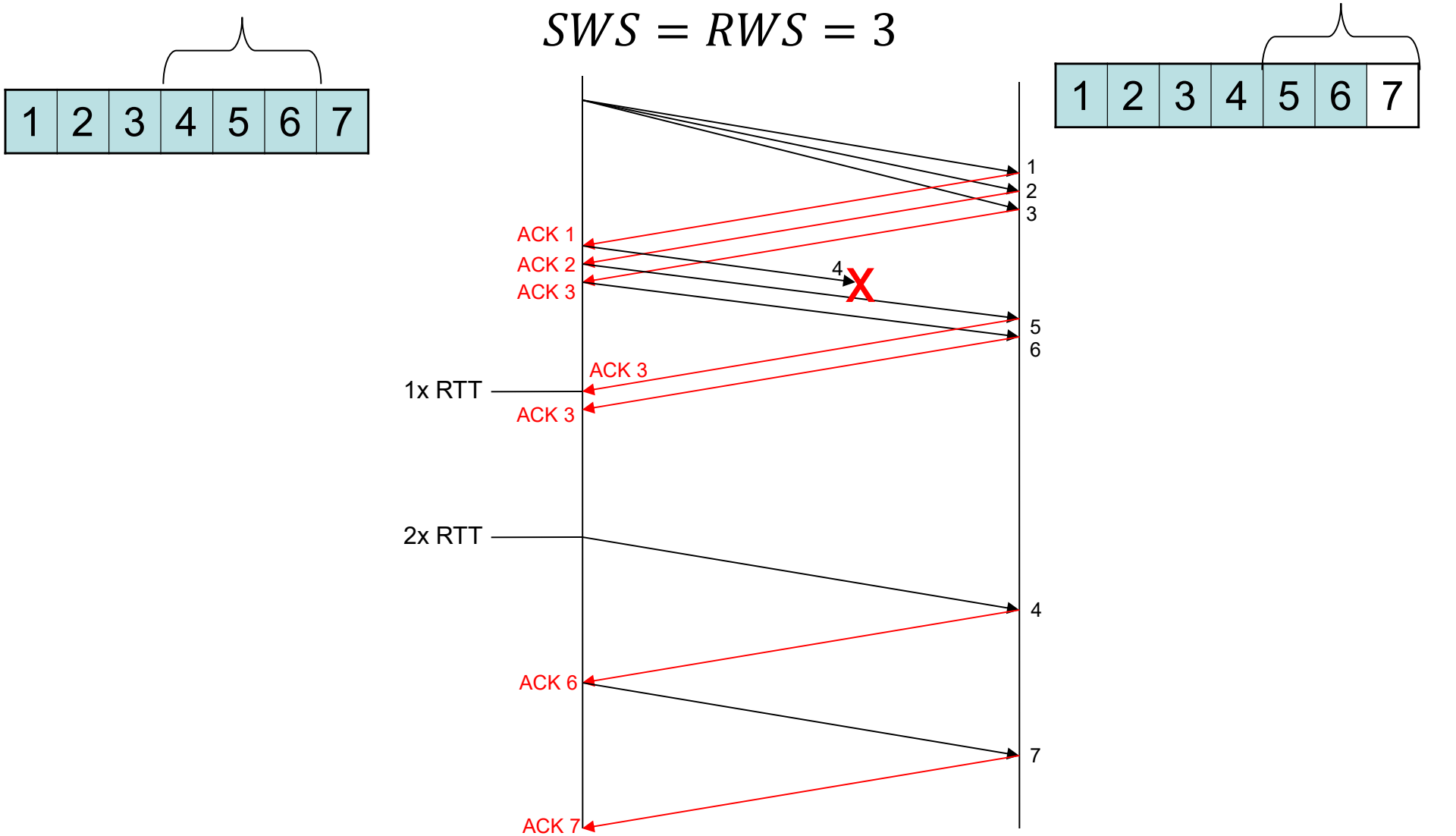
More complex because it's used in internet – not over a direct link

$Bandwidth \times delay$ not known

Dynamically changes timeouts

Larger buffers for in-order delivery

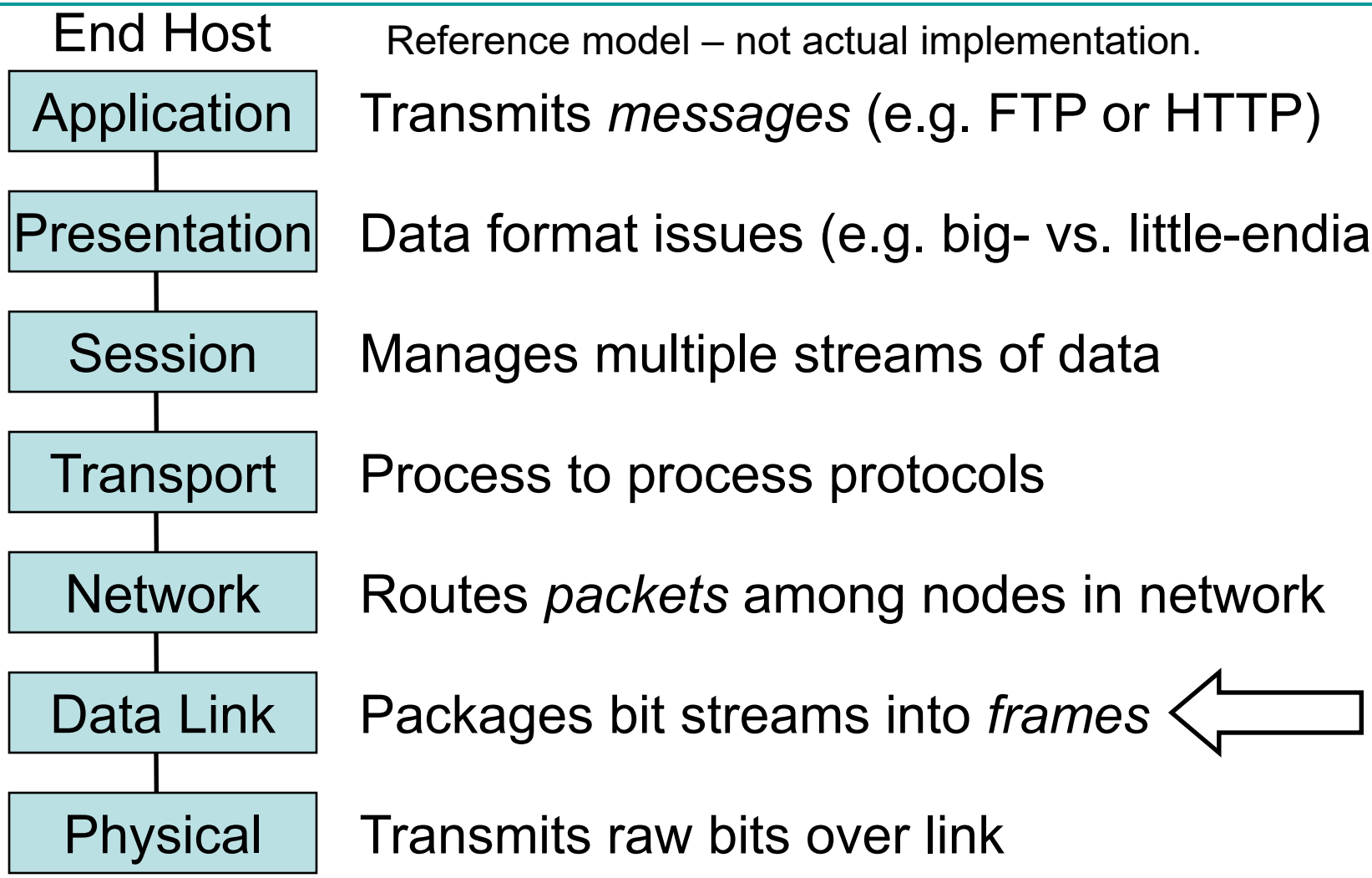
Example: SWS=RWS=3, 4 drops



So Far

- ARQ
  - Sliding Window
- Ethernet

Open Systems Interconnection (OSI)



IEEE 802 network standards

The IEEE 802 committee produces standards & specifications for Local Area Networks (LAN):

- **802.3 CSMA/CD Networks (Ethernet)**
- 802.4 Token Bus Networks
- 802.5 Token Ring Networks
- 802.6 Metropolitan Area Networks
- **802.11 Wireless LAN (Wifi)**

Ethernet (802.3)

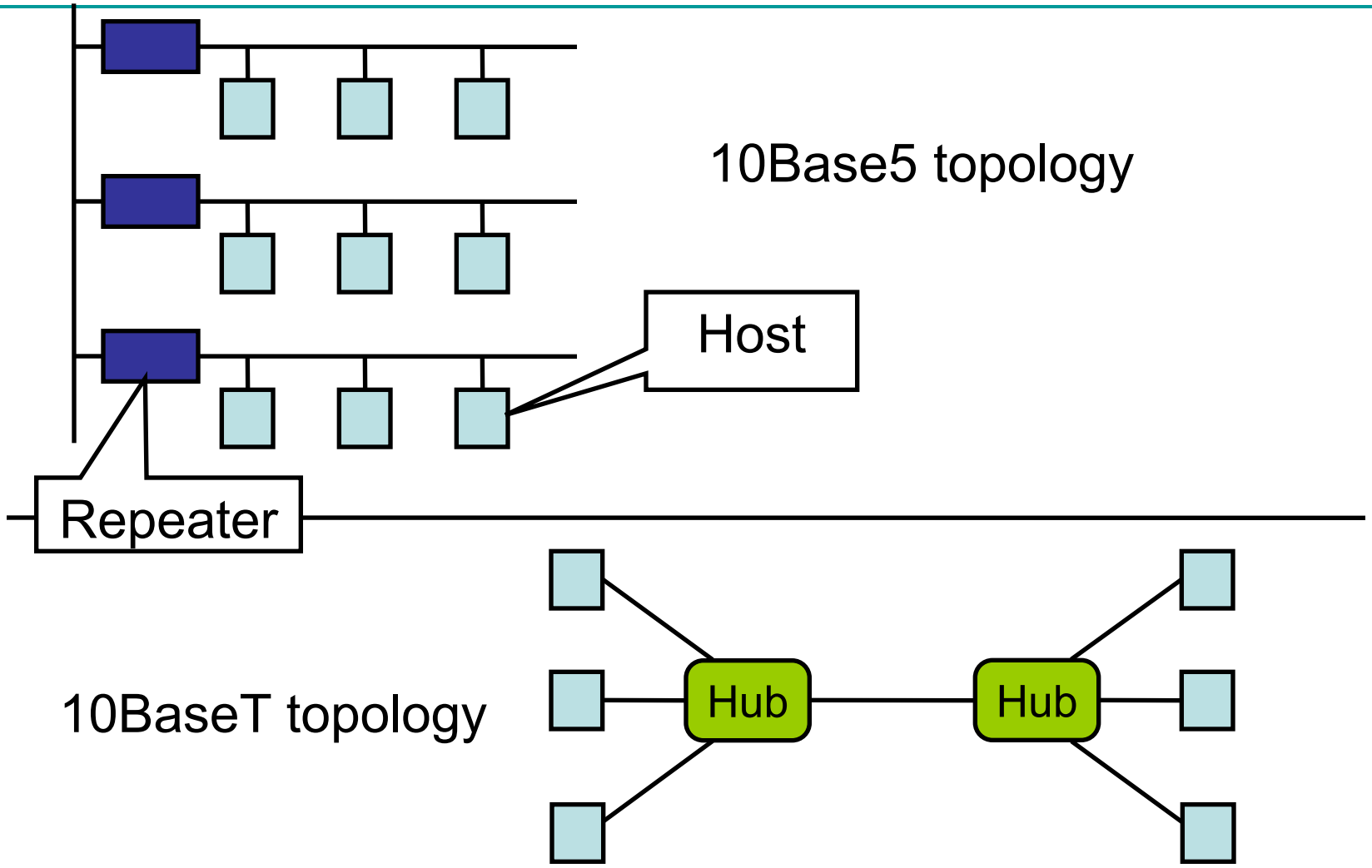
- A standard for local area networks (LAN)
- Developed in mid-70's at Xerox PARC
  - Descendent of Aloha, a U. of Hawaii radio packet network
  - DEC, Intel, and Xerox standard: 1978 for 10Mbps
  - IEEE 802.3 standard grew out of that
- Physical implementations:
  - 10Base5, 10Base2, 10BaseT, 10BaseF, 100BaseT, 1000BaseT…
  - Speed: 10Mbps,100Mbps, 1000Mbps, …

Name	Cable	Max Seg. Length	Nodes per Seg.	Advantages
10Base5	Thick coaxial	500 m	100	Original cable; obsolete
10Base2	Thin coaxial	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings, secure

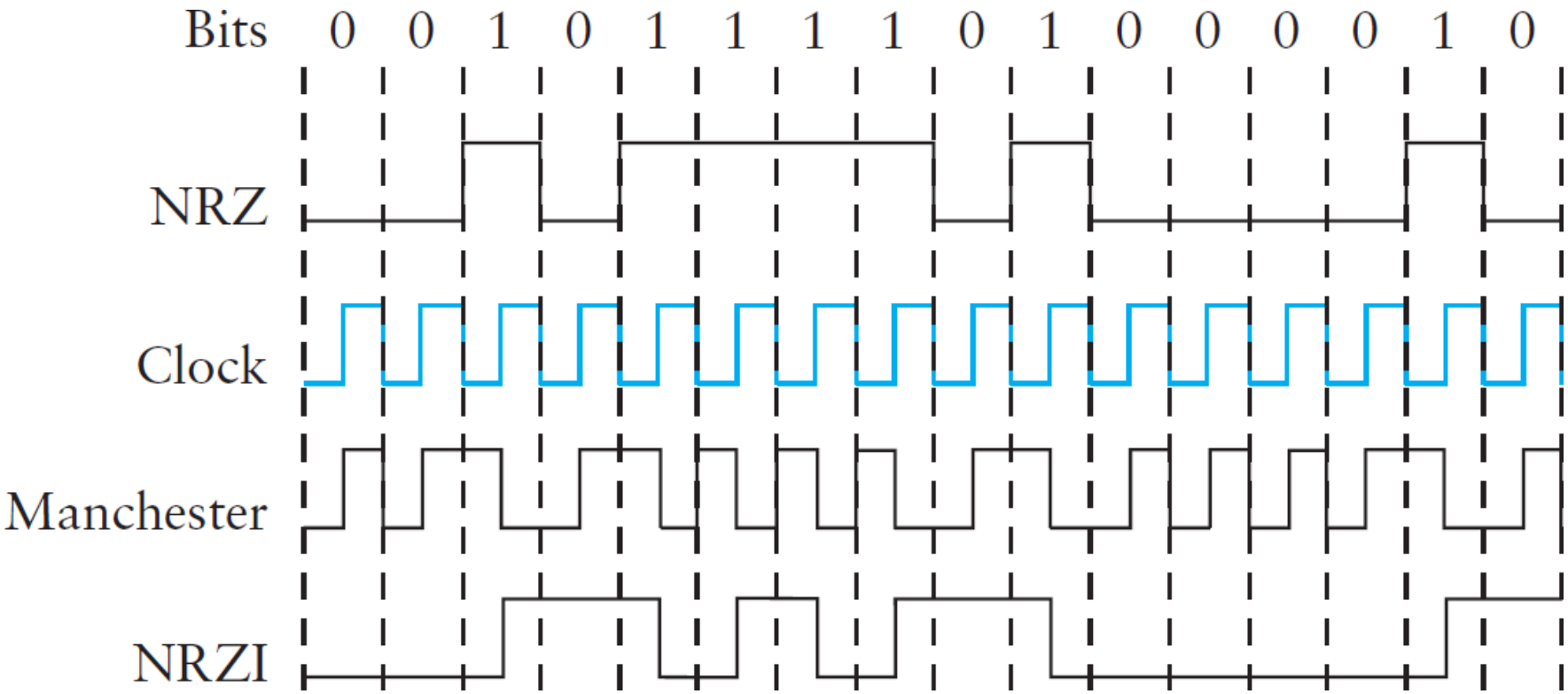
Ethernet Physical links

- Originally used “Thick-net” 10Base5
  - $10 = 10Mbps$
  - 5 = maximum of 500 meters segments
  - Up to 4 repeaters between two hosts
    $= 2500m$ max
- More common: 10BaseT
  - $10 = 10Mbps$
  - T = Twisted pair (typically Category 5), Maximum of 100 meter segments
  - Connected via *hubs* (still $2500m$ max)
- Today’s standards: 100BaseT, 1000BaseT

Ethernet topologies



Ethernet Encoding: Machester



Ethernet basics

The Ethernet link is *shared*

- Signal transmitted by one host reaches *all* hosts

Method of operation: **CSMA/CD**

- Carrier Sense, Multiple Access, with Collision Detection

Hosts competing for the same link are said to be in the same **collision domain**

Good news: easy to exchange data

Bad news: must regulate link access

Protocol: **Media Access Control (MAC)**


https://boudewijnhuijgens.getarchive.net/amp/media/uss-george-washington-936b83

Ethernet Addresses

- Every adapter manufactured has a unique address

6 bytes (48 bits) usually written in hexadecimal

Examples

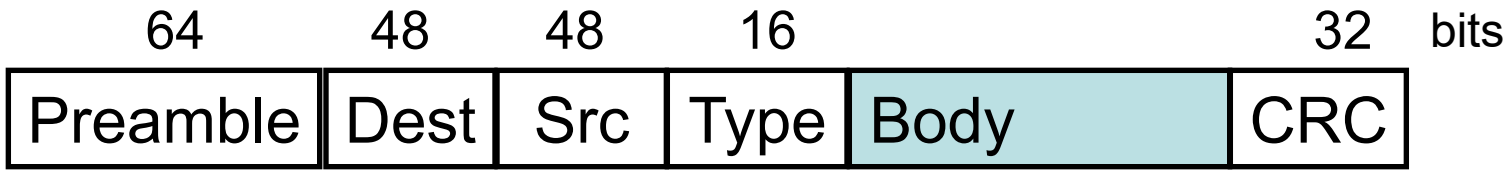
- `00-40-50-B1-39-69`
- `8:0:2b:e4:b1:2`

Each manufacturer is assigned 24bit prefix

Manufacturer ensures unique suffixes

- https://www.wireshark.org/tools/oui-lookup.html

Ethernet Frame Format



- Preamble – repeating pattern of 0's & 1's
  - Used by receiver to synchronize on signal
  - In Manchester this is $10MHz$ square wave for $6.4\ \mu s$
- Destination and Source – Ethernet Addresses
- Type – demultiplexing key
  - Identifies higher-level protocol
- Body – payload
  - Minimum $46\ Bytes$
  - Maximum $1500\ Bytes$

Conclusion

- ARQ
  - Sliding Window
- Ethernet