

CS 250: FOUNDATIONS OF COMPUTER SYSTEMS

[NETWORKING]

The Receiver's Buffer

Small it may be
But throttle the mightiest sender
It can

Not just the *how much*
But also the *when*
Or if *at all*

SHRIDEEP PALLICKARA
Computer Science
Colorado State University

COMPUTER SCIENCE DEPARTMENT



1

Frequently asked questions from the previous class survey

- Can a machine have both an IPv4 and IPv6 address?
 - ▣ Yes; dual-stack
- Checksums



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.2

2

Topics covered in this lecture

- TCP



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.3

3

So close, no matter how far
Couldn't be much more from the heart
Forever trusting who we are
And nothing else matters
Never opened myself this way
Life is ours, we live it our way
All these words, I don't just say
And nothing else matters
Trust I seek and I find in you
Every day for us something new
Open mind for a different view
And nothing else matters

Nothing Else Matters; James Alan Hetfield & Lars Ulrich; Metallica.

RELIABLE BYTE STREAM TCP (TRANSMISSION CONTROL PROTOCOL)

COMPUTER SCIENCE DEPARTMENT



COLORADO STATE UNIVERSITY

4

Components of Reliable delivery

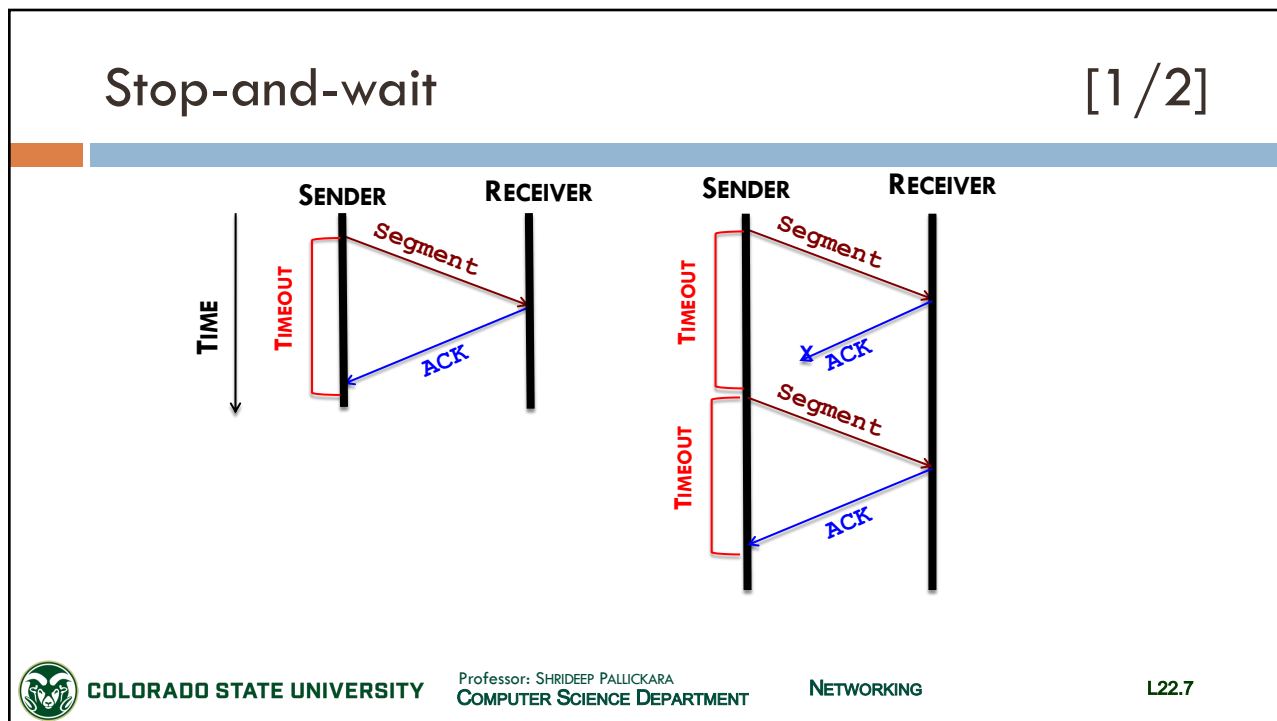
- **Acknowledgements**
 - Confirm receipt of data [with an ACK]
- **Timeouts**
 - *Retransmit* if ACK not received within a specified time
- Use of ACKs and timeouts to implement reliable delivery
 - Sometime called ARQ (**a**utomatic **r**epeat **r**equ**e**st)



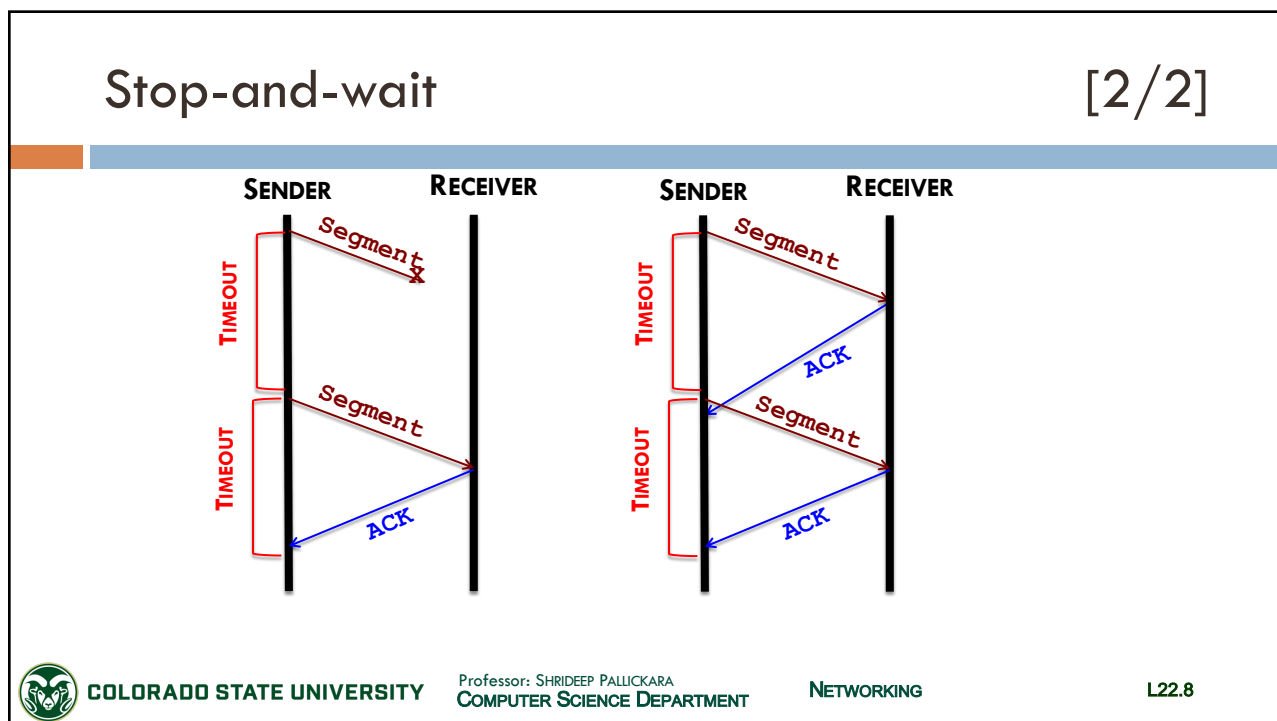
Simplest ARQ is the stop-and-wait algorithm

- After transmitting one segment
 - Sender **waits** for ACK before transmitting the next segment
- If the ACK does not arrive after a period of time
 - Sender **retransmits** the original segment





7



8

Sliding window: Try to fill the network pipe

- DELAY x BANDWIDTH product is 8 KB
- Data segments = 1KB
- Sender could transmit 9th segment
 - When ACK for the 1st segment arrives



COLORADO STATE UNIVERSITY

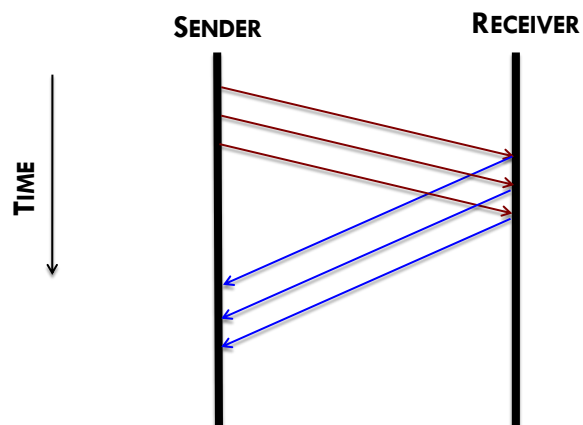
Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.9

9

Timeline for the sliding window



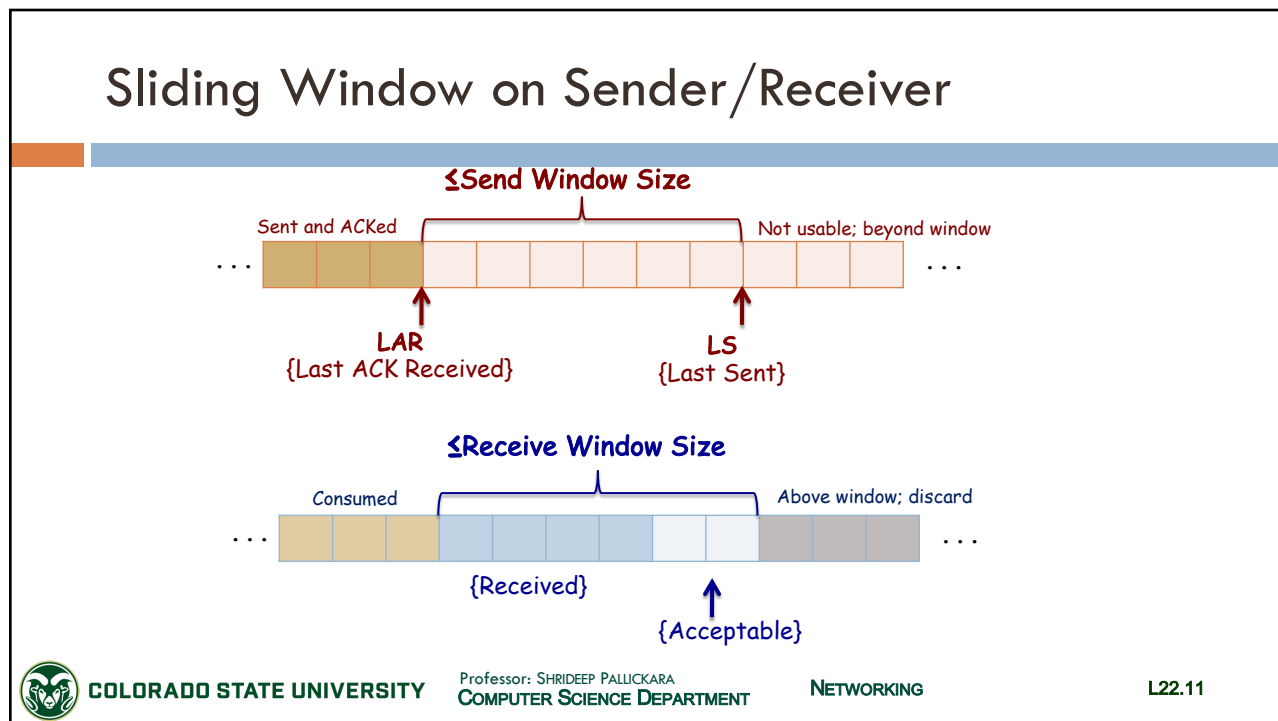
COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.10


10



11

Transmission Control Protocol (TCP) [1 / 2]

- **Reliable, in-order** delivery of byte streams
- **Full duplex** protocol
 - Each connection supports a pair of byte streams
 - Flowing in different directions
- Includes **flow control** mechanism
 - Allows receiver to limit the data sender
 - Control how much data can be transmitted at a time

 **COLORADO STATE UNIVERSITY** Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT NETWORKING L22.12

12

Transmission Control Protocol (TCP)

[2/2]

- Includes **multiplexing** mechanism
 - ▣ Multiple applications on a given host

- Implements a **congestion-control** mechanism
 - ① *Throttle* how fast TCP sends data
 - ② Keep sender from *overloading* the network



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.13

13

Flow control and congestion control

- **Flow control** is an end-to-end issue
 - ▣ Don't overrun capacity of *receiver*

- **Congestion control** is about how hosts & networks interact
 - ▣ Don't cause *switches* and *links* to be overloaded



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.14

14

TCP: Setup and Teardown

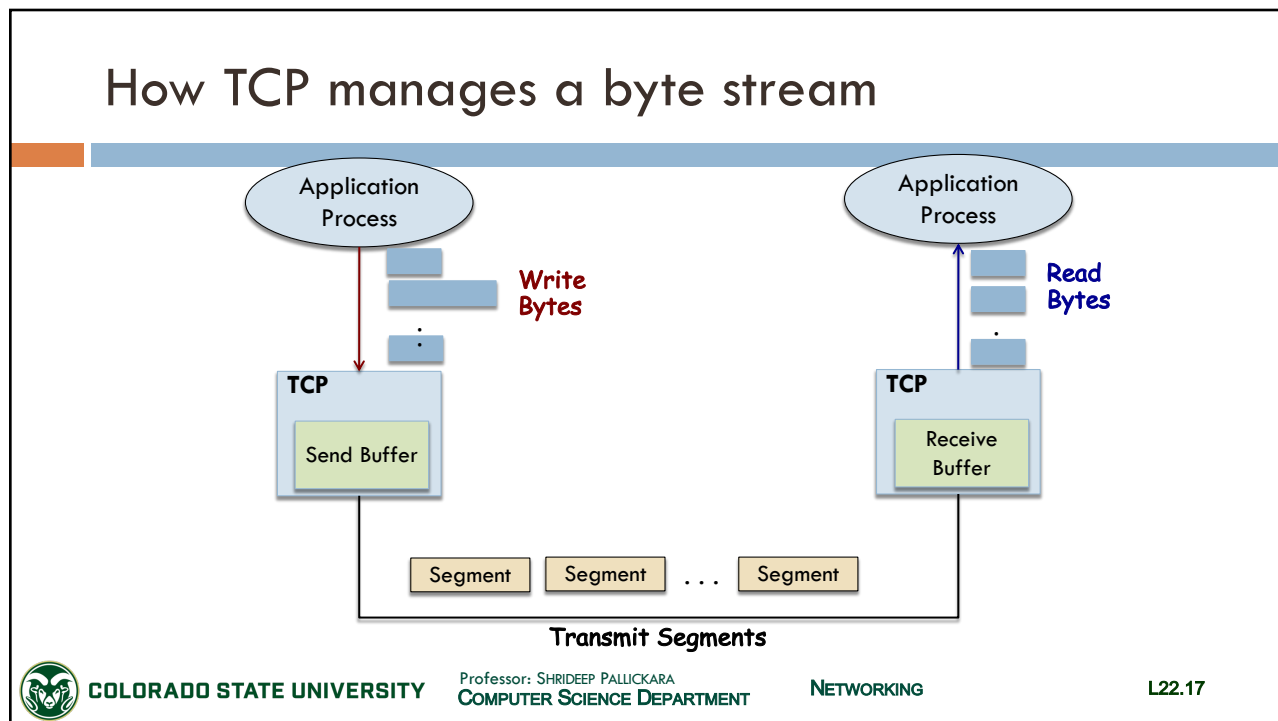
- Two sides of the connection *agree* to exchange data
 - ▣ Establish **shared state**
 - ▣ 3 packets exchanged (SYN, SYN-ACK, ACK)
- Connection teardown
 - ▣ Let each host know it is OK to *free* the shared state
 - ▣ 4 packets exchanged (FIN, ACK, FIN, ACK)



TCP Segments & how they come about

- TCP
 - ▣ Accepts data from a data stream
 - ▣ Breaks it up into chunks
 - ▣ Adds a TCP header ... creating a **TCP segment**
- Segment is then *encapsulated* in an IP datagram
- TCP packet is a term that you will often hear
 - ▣ Segment is more precise, packets are generally datagrams, frames are at the link layer





17

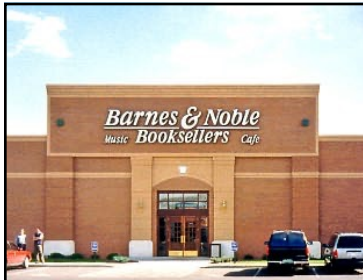
Maximum Segment Size (MSS)

- To avoid fragmentation in the IP layer, a host must specify the MSS as equal to the largest IP datagram that the host can handle (**minus** the IP and TCP header sizes)
- The **minimum** requirements (in bytes) at the hosts are as follows
 - IPv4: $576 - 20 - 20 = 536$
 - 576 bytes is the classic minimum size that every IPv4 destination host must be able to receive or reassemble (RFC 791)
 - IPv6: $1280 - 40 - 20 = 1220$
 - IPv6 requires every link to be able to carry at least a 1280-byte packet
- Each direction of the data flow can use a different MSS

576 and 1280 are not TCP numbers; they are IP-layer minimum size guarantees inherited by TCP when it computes a safe MSS.

COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALICKARA COMPUTER SCIENCE DEPARTMENT NETWORKING

18



TCP: A BOOKSTORE ANALOGY

COMPUTER SCIENCE DEPARTMENT



19

Our bookstore analogy

[1/2]

- The bookstore streams pages to you
- Each page is labeled not by page number, but by the line number of its first line
 - ▣ Let's assume each page (segment) carries 40 lines
 - ▣ Each page is labeled not by a page #, but by the line number of its first line
 - Page 1 carries lines 1–40 → sequence number 1
 - Page 2 carries lines 41–80 → sequence number 41
 - Page 3 carries lines 81–120 → sequence number 81
 - Page 4 carries lines 121–160 → sequence number 121



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.20

20

Our bookstore analogy

[2/2]

- The bookstore is sending you pages from a book
- It is allowed to have a few pages “in flight” before hearing back from you
 - ▣ That is the sliding window
- You do not say, “I got page 3”
 - ▣ Instead, you say: “The next line I need is line 121” That is the ACK number
- Let’s me now depict several TCP cases on the tablet ...



COLORADO STATE UNIVERSITY

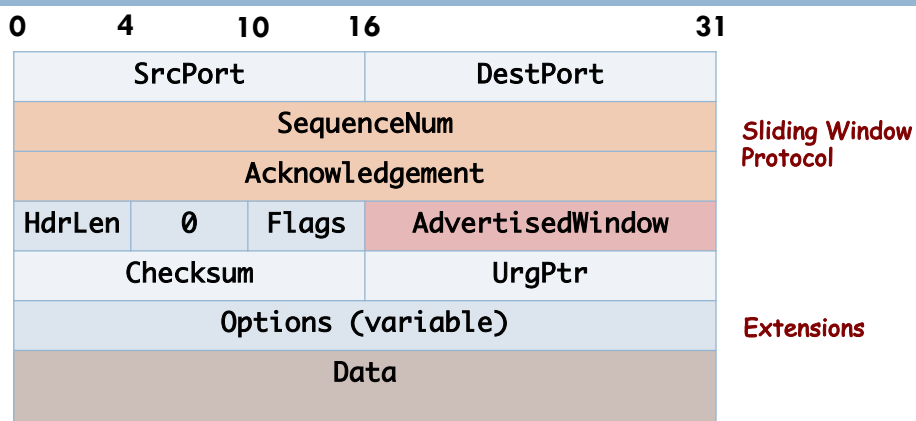
Professor: SHRIDEEP PALLICKARA
 COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.21

21

TCP Header Format



SourceAddr and DestinationAddr from IP



COLORADO STATE UNIVERSITY

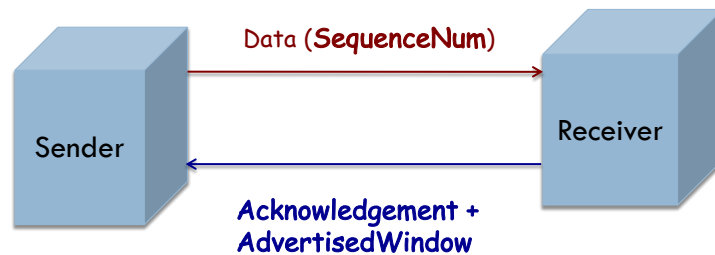
Professor: SHRIDEEP PALLICKARA
 COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.22

22

Relationship between SequenceNum, Acknowledgement and AdvertisedWindow



Each byte of data has a sequence number
SequenceNum contains sequence number for first byte of data in segment



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.23

23

TCP Sliding Window

[1/2]

- Guarantees **reliable** delivery of data
- Data is delivered in **order**
- Enforces **flow control** between the sender and receiver



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.24

24

TCP Sliding Window

[2/2]

- Sender has a **limit** on unacknowledged data
 - Limited to no more than **AdvertisedWindow** bytes of unacknowledged data
- Receiver **selects** **AdvertisedWindow**
 - Based on memory set aside for connection's buffer space



COLORADO STATE UNIVERSITY

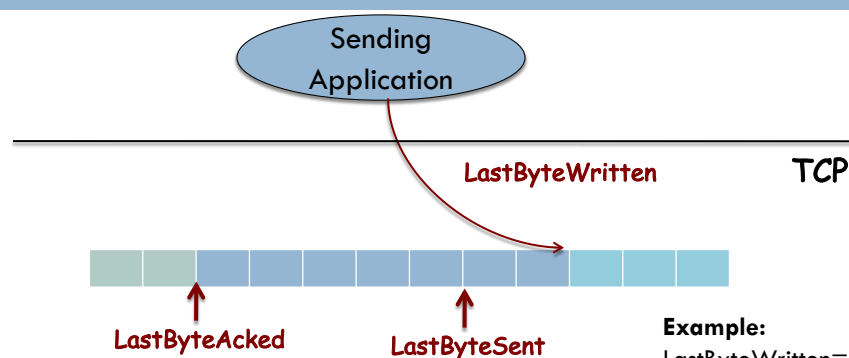
Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.25

25

TCP Send Buffer



$LastByteAcked \leq LastByteSent$
 $LastByteSent \leq LastByteWritten$

Example:

LastByteWritten= 3000

LastByteSent=2800

LastByteAcked:= 2400

How many unacknowledged bytes?
(3000-2400)= 600



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.26

26

TCP Receive Buffer

$LastByteRead < NextByteExpected$
 $NextByteExpected \leq LastByteRcvd + 1$

Example:
 LastByteRead= 2400
 LastByteExpected=2405
 LastByteRcvd:= 2407

COLORADO STATE UNIVERSITY
Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT
NETWORKING
L22.27

27

Flow Control: Buffers are of finite size

MaxSendBuffer and MaxRcvBuffer

- Receiver **throttles** sender
 - Advertises a window
 - No bigger than what it can buffer

Example:
 MaxRcvBuffer = 8 bytes
 LastByteRead= 2400
 NextByteExpected= 2405
 AdvertisedWindow = $8 - ((2405 - 1) - 2400) = 4$ bytes

$LastByteRcvd - LastByteRead \leq MaxRcvBuffer$
 AdvertisedWindow =
 $MaxRcvBuffer - ((NextByteExpected - 1) - LastByteRead)$

Space Utilized in the receiver's buffer

COLORADO STATE UNIVERSITY
Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT
NETWORKING
L22.28

28

The advertised window may potentially shrink

- If the process is reading data as fast as it arrives?
 - ▣ The advertised window *stays open*
 - i.e., $\text{AdvertisedWindow} = \text{MaxRcvBuffer}$
- If the receiving process falls behind?
 - ▣ Advertised window becomes *smaller* with every segment that arrives
 - ▣ Until it becomes \emptyset



Flow Control: Buffers are of finite size MaxSendBuffer and MaxRcvBuffer

- On the sender size, TCP **adheres** to the advertised window from the receiver

$$\text{LastByteSent} - \text{LastByteAked} \leq \text{AdvertisedWindow}$$

$$\text{EffectiveWindow} = \text{AdvertisedWindow} - (\text{LastByteSent} - \text{LastByteAked})$$

EffectiveWindow should be $> \emptyset$ before source can send more data



Reliability is achieved by the sender detecting lost data and retransmitting it

- TCP uses two primary techniques to identify loss
 - Retransmission timeout (RTO)
 - Duplicate cumulative acknowledgements (DupAcks)
 - If the sender receives three duplicate acknowledgements, it retransmits the last unacknowledged packet



Selective Acknowledgements (SACK)

- Using SACK, a receiver informs the sender of **non-contiguous blocks** of data that have been received and queued successfully
- So, the sender need retransmit only the segments that have actually been lost



ISSUES WITH TCP

COMPUTER SCIENCE DEPARTMENT



COLORADO STATE UNIVERSITY

33

Protecting against wraparound: 32-bit sequence space

- TCP assumes each segment has a max lifetime
 - ▣ **Maximum segment lifetime** (MSL)
 - ▣ Currently this is 120 seconds
- Sequence number used on a connection might wrap-around
 - ▣ Within the MSL



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.34

34

Time until 32-bit sequence number wraps around

Bandwidth	Time until wraparound
T1 (1.5 Mbps)	6.4 hours
Ethernet (10 Mbps)	57 minutes
T3 (45 Mbps)	13 minutes
FDDI (100 mbps)	6 minutes
STS-3 (155 Mbps)	4 minutes
STS-12 (622 Mbps)	55 seconds
STS-24 (1.2 Gbps)	28 seconds

STS : Synchronous Transport Signal

FDDI: Fiber Distributed Data Interface



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.35

35

Keeping the pipe full

- **AdvertisedWindow** field (16-bits) must be big enough
 - To allow sender to keep the pipe full
 - 16 bit allows us a max window size of 64 KB (2^{16})
- If receiver has unlimited buffer space?
 - **AdvertisedWindow** dictated by DELAY X BANDWIDTH product



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

NETWORKING

L22.36

36

Required Window Size for 100 ms delay

Bandwidth	Delay x Bandwidth Product
T1 (1.5 Mbps)	18 KB
Ethernet (10 Mbps)	122 KB
T3 (45 Mbps)	549 KB
FDDI (100 mbps)	1.2 MB
STS-3 (155 Mbps)	1.8 MB
STS-12 (622 Mbps)	7.4 MB
STS-24 (1.2 Gbps)	14.8 MB

STS : Synchronous Transport Signal

FDDI: Fiber Distributed Data Interface



TCP extensions: Use 32-bit timestamp to extend sequence number space

- **Distinguish** between different incarnations of the same sequence number
- Timestamp not treated as part of sequence number
 - For ordering etc.
 - Just protects against wraparound



TCP Extension: Allow TCP to advertise larger window

- Fill larger DELAY X BANDWIDTH pipes
- Include option that defines a **scaling** factor
- Option allows TCP endpoints to agree that **AdvertisedWindow** counts **larger chunks**



A caveat regarding Options

- You cannot solve all problems with Options
- TCP Header has room for only **44 bytes of options**
 - HdrLen is 4 bits long, so header length cannot exceed $16 \times 32\text{-bit} = 64$ bytes
 - Adding a TCP option that extends the space available for options?



The contents of this slide-set are based on the following references

- *Computer Networks: A Systems Approach. Larry Peterson and Bruce Davie. 4th edition. Morgan Kaufmann. ISBN: 978-0-12-370548-8. [Chapter 5, 6]*

