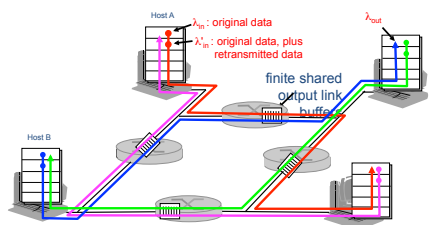


Causes/costs of congestion: scenario 3

- four senders
- multihop paths
- timeout/retransmit

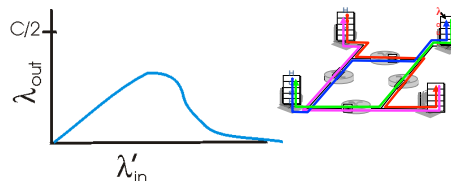
Q: what happens as λ_{in} and λ'_{in} increase?



Transport Layer

3-1

Causes/costs of congestion: scenario 3



Another "cost" of congestion:

- when packet dropped, any "upstream transmission capacity used for that packet was wasted!"

Transport Layer

3-2

Approaches towards congestion control

Two broad approaches towards congestion control:

End-end congestion control:

- no explicit feedback from network
- congestion inferred from end-system observed loss, delay
- approach taken by TCP

Network-assisted congestion control:

- routers provide feedback to end systems
 - single bit indicating congestion (SNA, DECbit, TCP/IP, ATM)
 - explicit rate sender should send at

Transport Layer

3-3

TCP Slow Start

- When connection begins, **CongWin** = 1 MSS
 - Example: MSS = 500 bytes & RTT = 200 msec
 - initial rate = 20 kbps
- available bandwidth may be \gg MSS/RTT
 - desirable to quickly ramp up to respectable rate

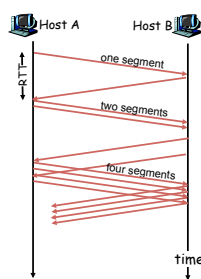
- When connection begins, increase rate exponentially fast until first loss event

Transport Layer

3-4

TCP Slow Start (more)

- When connection begins, increase rate exponentially until first loss event:
 - double **CongWin** every RTT
 - done by incrementing **CongWin** for every ACK received
- **Summary:** initial rate is slow but ramps up exponentially fast



Transport Layer

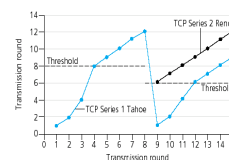
3-5

Refinement

- Q: When should the exponential increase switch to linear?
- A: When **CongWin** gets to 1/2 of its value before timeout.

Implementation:

- Variable Threshold
- At loss event, Threshold is set to 1/2 of CongWin just before loss event



Transport Layer

3-6

Refinement: inferring loss

- After 3 dup ACKs:
 - CongWin** is cut in half
 - window then grows linearly
- But after timeout event:
 - CongWin** instead set to 1 MSS;
 - window then grows exponentially
 - to a threshold, then grows linearly

Philosophy:

- 3 dup ACKs indicates network capable of delivering some segments
- timeout indicates a "more alarming" congestion scenario

Transport Layer

3-7

Summary: TCP Congestion Control

- When **CongWin** is below **Threshold**, sender in **slow-start** phase, window grows exponentially.
- When **CongWin** is above **Threshold**, sender is in **congestion-avoidance** phase, window grows linearly.
- When a **triple duplicate ACK** occurs, **Threshold** set to **CongWin/2** and **CongWin** set to **Threshold**.
- When **timeout** occurs, **Threshold** set to **CongWin/2** and **CongWin** is set to 1 MSS.

Transport Layer

3-8

TCP sender congestion control

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

Transport Layer

3-9

Delay modeling

Q: How long does it take to receive an object from a Web server after sending a request?

Ignoring congestion, delay is influenced by:

- TCP connection establishment
- data transmission delay
- slow start

Notation, assumptions:

- Assume one link between client and server of rate R
- S: MSS (bits)
- O: object size (bits)
- no retransmissions (no loss, no corruption)

Window size:

- First assume: fixed congestion window, W segments
- Then dynamic window, modeling slow start

Transport Layer

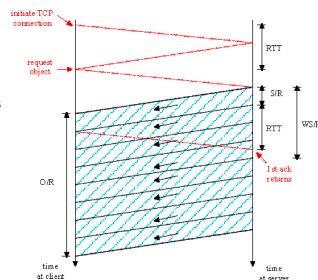
3-10

Fixed congestion window (1)

First case:

$WS/R > RTT + S/R$: ACK for first segment in window returns before window's worth of data sent

$$\text{delay} = 2RTT + O/R$$



Transport Layer

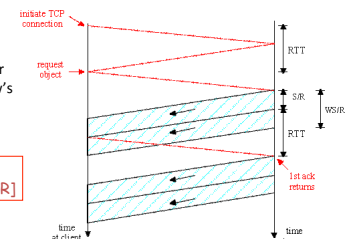
3-11

Fixed congestion window (2)

Second case:

- $WS/R < RTT + S/R$: wait for ACK after sending window's worth of data sent

$$\text{delay} = 2RTT + O/R + (K-1)[S/R + RTT - WS/R]$$



Transport Layer

3-12

TCP Delay Modeling: Slow Start (1)

Now suppose window grows according to slow start

Will show that the delay for one object is:

$$\text{Latency} = 2RTT + \frac{O}{R} + P \left[RTT + \frac{S}{R} \right] - (2^P - 1) \frac{S}{R}$$

where P is the number of times TCP idles at server:

$$P = \min\{Q, K - 1\}$$

- where Q is the number of times the server idles if the object were of infinite size.
- and K is the number of windows that cover the object.

Transport Layer

3-13

TCP Delay Modeling: Slow Start (2)

Delay components:

- 2 RTT for connection estab and request
- O/R to transmit object
- time server idles due to slow start

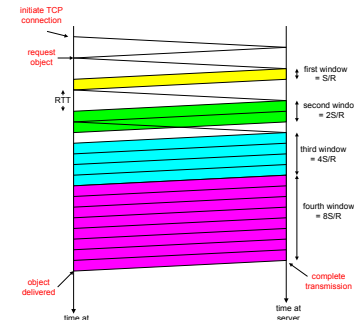
Server idles:

$$P = \min\{K-1, Q\} \text{ times}$$

Example:

- $O/S = 15$ segments
- $K = 4$ windows
- $Q = 2$
- $P = \min\{K-1, Q\} = 2$

Server idles $P=2$ times



Transport Layer

3-14

TCP Delay Modeling (3)

$\frac{S}{R} + RTT$ = time from when server starts to send segment until server receives acknowledgement

$2^{k-1} \frac{S}{R}$ = time to transmit the k th window

$\left[\frac{S}{R} + RTT - 2^{k-1} \frac{S}{R} \right]^+$ = idle time after the k th window

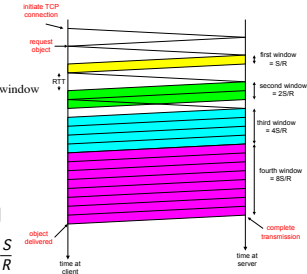
$$\text{delay} = \frac{O}{R} + 2RTT + \sum_{p=1}^P \text{idleTime}_p$$

$$= \frac{O}{R} + 2RTT + \sum_{p=1}^P \left[\frac{S}{R} + RTT - 2^{p-1} \frac{S}{R} \right]$$

$$= \frac{O}{R} + 2RTT + P \left[RTT + \frac{S}{R} \right] - (2^P - 1) \frac{S}{R}$$

Transport Layer

3-15



TCP Delay Modeling (4)

Recall K = number of windows that cover object

How do we calculate K ?

$$K = \min\{k: 2^0 S + 2^1 S + L + 2^{k-1} S \geq O\}$$

$$= \min\{k: 2^0 + 2^1 + L + 2^{k-1} \geq O/S\}$$

$$= \min\{k: 2^k - 1 \geq \frac{O}{S}\}$$

$$= \min\{k: k \geq \log_2(\frac{O}{S} + 1)\}$$

$$= \left\lceil \log_2(\frac{O}{S} + 1) \right\rceil$$

Calculation of Q , number of idles for infinite-size object, is similar.

Transport Layer

3-16