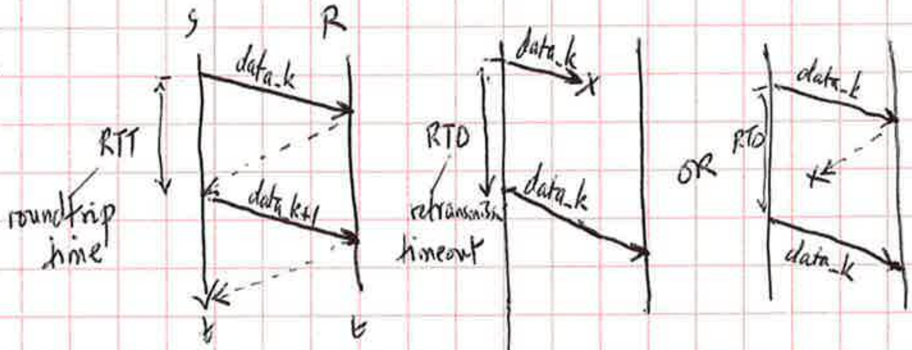
Recitation 22

12-6-2011

Last Time:

Stop-and-Wait ARQ

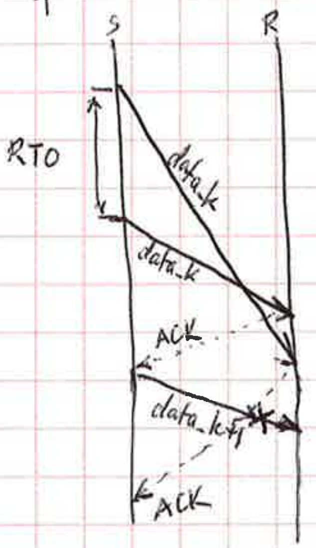


Normal Behavior

Data loss + retransmission

duplicate packet reception (use rcv_seqnum to prevent duplicate forwarding to application)

(*) If no sequence_number in ACK,



- Receiver sends ACK for data_k retransmission
- Sender thinks ACK for data_k+1 which never arrived!

Throughput of Stop-and-Wait Protocol

Case 1: No packet loss:

Throughput = 1 packet per RTT ~ round-trip time.

ie $$\boxed{\text{Throughput} = \frac{1}{RTT}}$$ packets/s

Case 2: With packet loss:

$$\text{Throughput} = \frac{1}{T}, \text{ where } \boxed{T = (1-\ell)RTT + \ell(RTO + T)}$$

(T: expected time to send packet and get ACK)

where ℓ = packet loss rate (prob of packet OR ACK lost)

So that $$T = RTT + \frac{\ell}{1-\ell} RTO$$

and $$\text{Throughput} = \frac{1}{T}$$

Example: Given: Network between Boston and SFO

RTT = 100 ms ; Network bit-rate = 1 Mbps.

Suppose: packet size = 10,000 bits ; RTO = 120 ms ; Packet loss rate ℓ = 1%

Q: Calculate throughput of stop-and-wait protocol.

Soln: Expected $T = RTT + \frac{\ell}{1-\ell} RTO = 100\,ms + \frac{1\%}{1-1\%}(120\,ms) \simeq 101\,ms.$

So that throughput $= \frac{1}{T} = \frac{1}{101\,ms} \approx 9.9$ Packets/s.

$= 9.9 \frac{\text{Pack}}{s} \cdot 10000 \frac{\text{bits}}{\text{Pack}} = 99{,}000$ bits/s = 99 kbps

So we see that utilization $= \frac{99\text{ kbps}}{1\text{ Mbps}}$ = Only 9.9% of maximum channel bit-rate ~ inefficient

Defns: T_{P_SR} ~ Propagation delay → 1st bit sent by S (to) 1st bit received by R

T_{TRANS_PKT} ~ Transmission delay (of packet) → 1st PKT at R (to) Last PKT at R

(Eg2) For Stop-and-Wait protocol, the receiver computes CRC after it gets the entire data packet. We assume the processing time to calculate CRC is constant. Suppose the transmission time and processing time for the ACK is negligible, and the propagation delay for each packet is negligible. The link bit rate is 54 Mbps and each packet is 540 bits long. What is the maximum CRC processing time that ensures that the protocol will achieve a throughput of at least 50% of the link bit-rate, in the absence of data packet and ACK loss?

Given:

ACK transmission ~ $T_{TRANS\_ACK} = 0$

ACK processing ~ $T_{PROC\_ACK} = 0$

No loss! (case 1).

$T_{P\_SR} = 0$ ~ Propagation delay of packet btn a sender & receiver.

Link bit-rate $R = 54$ Mbps

Packet-size $P = 540$ bits.

Required: Max $T_{PROC\_CRC} = ?$ for $\lambda = 0.5R$

Soln: Since no loss, $\lambda = \frac{1}{RTT}$

where RTT = PKT transmission Time (T_{TRANS_PKT}) = Need calculate
+ CRC processing Time (T_{PROC_CRC}) = ?
+ ACK Transmit Time (T_{TRANS_ACK}) = 0
+ ACK Processing Time (T_{PROC_ACK}) = 0
+ PKT Propagation Time (T_{P_SR}) = 0

Need compute T_{TRANS_PKT}

for $\lambda = (0.5)(R) = (0.5)(54\text{Mbps}) = 27$ Mbps.

i.e. $\lambda = \frac{1}{RTT} \geq 27$ Mbps.

$$= \frac{1}{(T_{TRANS\_PKT} + T_{PROC\_CRC}(max))} = 27\text{Mbps}.$$

$$T_{TRANS\_PKT} = \frac{\text{Packet-size } P}{\text{link bit-rate } R} = \frac{P}{R} = \frac{540 \text{ bits/pkt}}{54 \text{Mbps}} = 10^{-5} s.$$

So $\frac{1}{(10^{-5} + T_{PROC\_CRC})} = 27$ Mbps so that $T_{PROC\_CRC} \leq 10\mu s$

Sliding Window Protocol

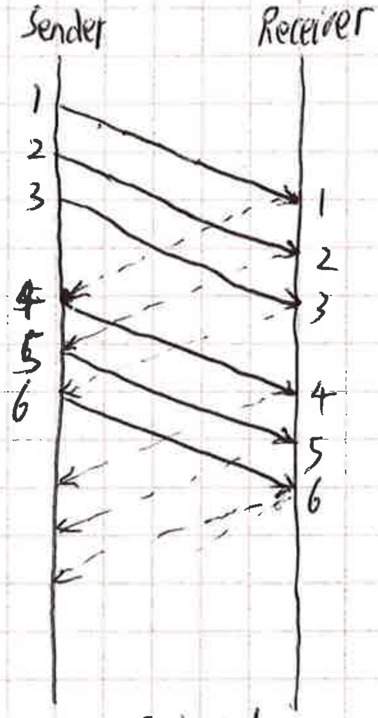
① Use a window

* Allow W packets outstanding in the network at any time.
  (W: sliding window size)
* overlap transmission with ACKs.

②. Sender advances the window by 1 for each in-sequence ack it receives.

③ The receiver is almost exactly the same as in stop-and-wait case.

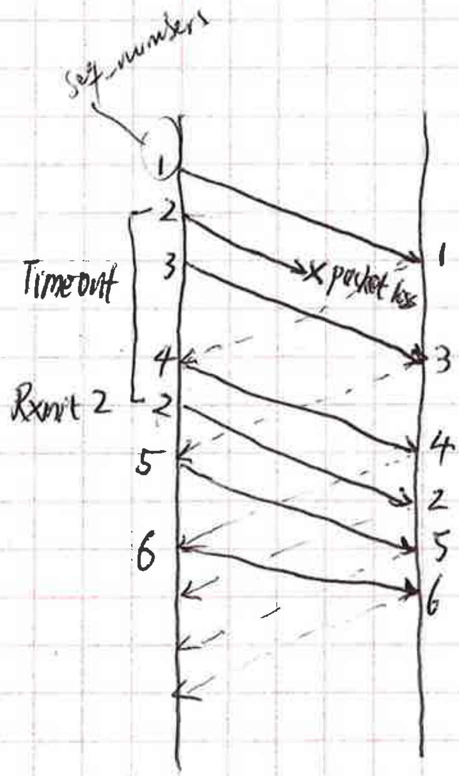
~~except it must buffer~~



No Packet loss

~~Sliding Window Protol~~

(window size = 3)



Pack loss

(Window Size = 3)

Throughput of Sliding Window Protocol

5

1. No packet loss.

throughput $= \min(\mu, \frac{W}{RTT})$ — Little's Law $\lambda = \frac{N}{D}$

where: μ ~ the rate of bottleneck link.

W: sliding window size.

RTT: round-trip time

2. With Packet loss.

① Select W that is slightly larger than $\mu \cdot RTT_{min}$ to ensure bottleneck link is busy even if there are packet losses.

② Total expected # of transmission, T, for successfully delivery.

with probability $1-l$, we need one transmission

with probability $l(1-l)$, we need 2 transmission

with probability $l^2(1-l)$, we need 3 transmission

⋮

(where l: round-trip loss rate)

So: $T = (1-l)\cdot 1 + l(1-l)\cdot 2 + l^2(1-l)\cdot 3 + l^3(1-l)\cdot 4 + \cdots$

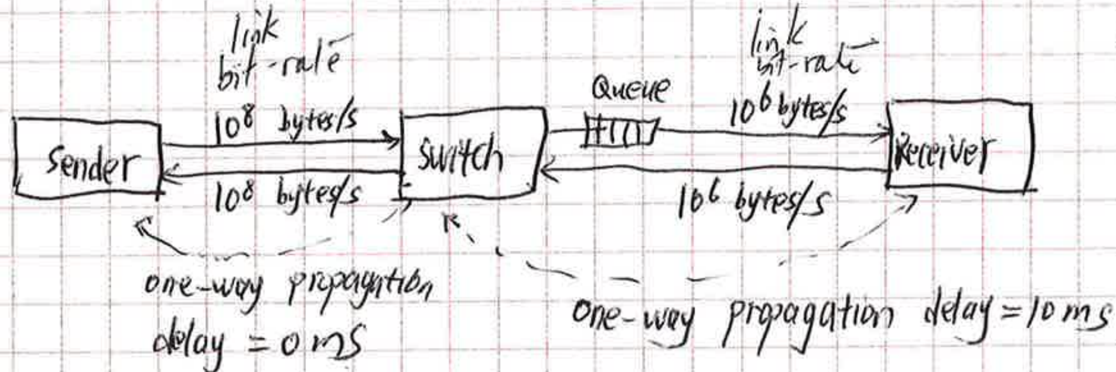
$= 1-l + 2l - 2l^2 + 3l^2 - 3l^3 + 4l^3 - 4l^4 + \cdots$

$= 1 + l + l^2 + l^3 + \cdots$

$= \frac{1}{1-l}$

Throughput (Utilization) $= \frac{1}{T} = 1-l$

Example.



packet size = 1000 bytes

ACK size = 40 bytes

sender window size = 10 packets

No other traffic ; No packet loss, No processing delay

Q: At what approximate rate (in packets per second) will the protocol deliver a multi-gaga byte file from the sender to the receiver

Solution: [throughput = $\min\{\mu, \frac{W}{RTT}\}$] $\approx \frac{\text{packet size}}{\text{bottleneck link rate}}$

round-trip time RTT = propagation delay + transmission delay

$$= 20\,ms + \frac{1000\,bytes}{10^6\,bytes/s}$$

$$= 21\,ms.$$

window size W = 10 packets.

bottleneck link rate μ = 10^6 bytes/s = 10^3 packets/s. (packet size = 1000 bytes)

So. throughput = $\min\left(\mu, \frac{W}{RTT}\right)$

$$= \min\left(1000, \frac{10}{21\,ms}\right)$$

$$= 476\ \text{packets/s}.$$