Last Time:

- Throughput of Stop-and-Wait
  
  Case 1: No packet loss
  \[ \lambda = \frac{1}{RTT} \text{ pkts/s} \]

  Case 2: With packet loss: rate \( \mu \)
  \[ T = (1-\mu)RTT + \mu(RTO+T) \]
  
  Expected time to send Pkt and receive ACK.

  \[ \lambda = \frac{1}{T} \] where
  \[ T = RTT + \frac{1}{1-\mu}RTO \]

Today:

- Throughput of Sliding Window
  
  Case 1: No packet loss:
  \[ \lambda = \min\left(\mu, \frac{W}{RTT_{\text{min}}}\right) = \begin{cases} \mu & W \geq \mu \times RTT_{\text{min}} \\ \frac{W}{RTT_{\text{min}}} & W < \mu \times RTT_{\text{min}} \end{cases} \]

  Case 2: With packet loss:

Suppose: Set window size \( W > \mu \times RTT_{\text{min}} \) so that bottleneck link busy!
Suppose prob. of bidirectional data packet or ACK loss = \( l \)

Total expected # transmissions \( T \) for successful delivery of a packet & its ACK:

- With prob. \( (1-l) \), need 1 transmission
- \( 1 - l(1-l) \) transmissions
- \( 1 - l^2(1-l) \) transmissions
- \( 1 - l^3(1-l) \) transmissions
- ... 

So that:

\[
T = (1-l) + (1-l) + l^2(1-l) + 3 + l^3(1-l) + 4 + \ldots \\
= (1-l) + (2l - 2l^2) + (3l^2 - 3l^3) + (4l^3 - 4l^4) + \ldots \\
= 1 + l + l^2 + l^3 + \ldots \\
= \frac{1}{1-l}
\]

And thus:

\[
\lambda = \frac{1}{T} = \frac{1}{1-l}
\]

(Throughput / Utilization)

\[
\forall = 1 - l
\]

Recall: (Last time) Stop-N-Wait protocol with BDS and SFO \( \sim 10\% \) utilization

For Sliding Window, one can obtain utilization \( \sim 100\% \) for small \( l \).

\[ \text{RTT} = 100 \text{ms}, \]

Bottleneck link bandwidth \( \mu = 1 \text{ Mbyte/s} \)

Packet size = 1000 bytes

Assume: No packet loss.

 regard: 

Q1: What’s the highest throughput of the stop-\&-wait protocol?

Q2: To improve performance, we implement sliding window. What should \( W \) be in order to saturate the bottleneck link capacity?

Q3: What is the throughput of the sliding window protocol in Q2?

\[ \text{Soln:} \]

1. SNW: (No pkt loss): 
\[ \lambda = \frac{1 \text{ PKT}}{\text{RTT}} = \frac{1 \text{ PKT}}{100 \text{ ms}} = 10 \text{ PKT/s} \]

and since 1 PKT = 1000 bytes,
\[ \lambda = \frac{10,000 \text{ Bytes/s}}{\text{s}} = 10 \text{ KBytes/s} \]

2. Make \( W = \mu * \text{RTT}_{\text{min}} \) (bandwidth-delay product) 
\[ W = (1 \text{ Mbyte/s}) * (100 \text{ ms}) = 100 \text{ KBytes} = 100 \text{ Packets} \]

3. When \( W = \mu * \text{RTT}_{\text{min}}, \quad \lambda = \mu = 1 \text{ MBytes/s} \)

Comparing throughput of SNW to that of SW protocol, 10 KBytes/s vs 1 MBytes/s
Ex. 2: Consider the following network setup:

![Network Diagram]

- Link bit-rate: $10^8$ bytes/s
- Queue: $10^6$ bytes
- Switch: $10^8$ bytes/s
- One-way propagation delay: 0 ms
- One-way propagation delay: 10 ms

Given:
- Packet size = 1000 bytes ($W$)
- ACK size = 40 bytes
- Sender Window Size = 10 packets
- No other traffic; no packet loss; no processing delay.

Question: At what approximate rate (in PKTs/s) will the protocol deliver a multi-gigabyte file from the sender to the receiver?

Solution:
Since no packet loss,

\[ \lambda = \min \left( \frac{W}{RTT_{\text{min}}} \right) \]

\[ W = 1000 \text{ bytes} \]

\[ m = 10^6 \text{ bytes/s} = 10^3 \text{ PKTs/s} \]

Need $RTT_{\text{min}}$

\[ RTT = \text{Propagation delays} + \text{Transmission delays} + \text{Processing delays} \]

- (PKT and ACK) transmission delay is negligible (small)
- (PKT and ACK) propagation delays are negligible (small)

\[ RTT = 20 \text{ ms} + \frac{1000 \text{ bytes}}{10^8 \text{ bytes/s}} = 21 \text{ ms} \]

So that \[ \frac{W}{RTT_{\text{min}}} = \frac{10^{10}}{21 \text{ ms}} = 476 \text{ PKTs/s} \]

Thus, \[ \lambda = \min \left( m, \frac{W}{RTT_{\text{min}}} \right) = \min \left( 10^3, 476 \right) \text{ PKTs/s} = 476 \text{ PKTs/s} \]
Consider a reliable transport connection using sliding window protocol.

Given: \( RTT_{\min} = 0.1 \text{ s} \) (\( RTT_{\min} \) is RTT in the absence of queuing delay).

\( \text{PKT size} = 1000 \text{ bytes} \)

\( \text{No other traffic, no pkt loss} \)

Questions:

Q1: If bottleneck link rate \( \mu = 100 \text{ packets/s} \) and window size \( W = 8 \text{ packets} \), what is the throughput?

Q2: If bottleneck link rate remains the same (\( \mu = 100 \text{ packets/s} \)), but window size \( W \) increases to 16 Packets, what is the throughput?

Q3: What is the smallest window size for which the connection's RTT exceeds \( RTT_{\min} \)?

Q4: Suppose we set \( W = \mu \times RTT_{\min} \). If we use an 8-bit field for the sequence number in each packet, what is the smallest value of the bottleneck link bandwidth (\( \mu \)) that will cause the protocol to stop working correctly?

Solutions:

1. Since no PKT loss, \( \lambda = \min \left( \mu, \frac{W}{RTT_{\min}} \right) \) where \( \mu = 100 \text{ PKTS/s} \)

\( \frac{W}{RTT_{\min}} = \frac{8 \text{ PKTS}}{0.1 \text{ s}} = 80 \text{ PKTS/s} \)

Then \( \lambda = \min \left( 100, 80 \right) \text{ PKTS/s} = 80 \text{ PKTS/s} \)

Alternatively, since \( \mu \times RTT_{\min} = 100 \text{ PKTS/s} \times 0.1 \text{ s} = 10 \text{ PKTS/s} \),

\( \lambda = \frac{\mu}{RTT_{\min}} = \frac{80 \text{ PKTS/s}}{0.1 \text{ s}} = 800 \text{ PKTS/s} \)
2. Again, since $M \cdot RTT_{\text{min}} = 10 < 16$, 

$$\lambda = \mu = 100 \text{ PKTs/s}.$$ 

Or $\lambda = \min(100, \frac{16}{0.1}) = 100$ as above.

3. When $W > M \cdot RTT_{\text{min}}$, there is queuing delay in front of the bottleneck link, and $RTT$ exceeds $RTT_{\text{min}}$.

$$M \cdot RTT_{\text{min}} = (100 \text{ PKTs/s})(0.1) = 10 \text{ PKTS}$$

So the smallest $W$ for which $RTT$ exceeds $RTT_{\text{min}}$ is $11 \text{ PKTS}$.

4. When $W > 2^{16} = 256$, the sequence number field wraps around, and the protocol stops working correctly.

It: the sequence numbers are no longer unique!

So for $W = 256$ packets, protocol stops working correctly.

$$\frac{W}{RTT_{\text{min}}} = \frac{256 \text{ PKTS}}{0.1 \text{s}} = 2560 \text{ PKTS/s} = 2560 \times 1000 \text{ Bytes/s} = 256 \text{ MBytes/s}$$

Thus, the smallest value of the bottleneck link bandwidth for which the protocol stops working correctly is $256 \text{ MBytes/s}$.