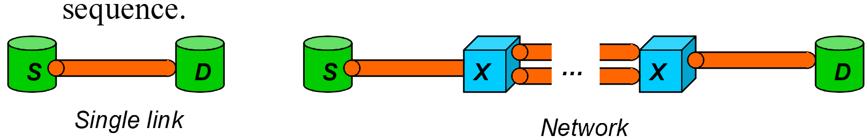
How do we evaluate protocols?

- Metrics:
  - Throughput
  - Delay (+ delay variance (jitter))
  - Probability of packet loss
  - Efficiency
  - Etc …
- Methodology:
  - Analytical
  - Experimental (Empirical)
  - Simulation (how this is different from Experimental techniques?)
- Why bother?
- Example: ARQ Protocols

*Automatic **R**etransmission re**Q**uest (ARQ)*

- The goal is to detect erroneous frames/packets and request retransmission.
- Note that ARQ techniques can operate on the link and the network layer. In principle, there is no difference, but in practice there is:

  On the data link layer, all frames follow the same "path." This is not so on the network layer. Why?
- Thus, on the data link layer, frames will not arrive out-of-sequence.



*Automatic **R**etransmission re**Q**uest (ARQ)*

- ARQ techniques have to <u>correctly</u> (properly sequenced, no duplicates, no loss) deliver frames/packets to the upper layer. ARQ schemes should also be also *efficient.*
- Two factors contribute to loss of efficiency:
  - control information overhead and
  - link idle times
- The assumption behind ARQ techniques is that an erroneous frame/packet will be discarded. Then, such a loss will be detected and a "replacement" copy of the packet resend.
- Note that the destination has no way of identifying which node an erroneous packet was destined to. Why?

*Automatic **R**etransmission re**Q**uest (con't)*

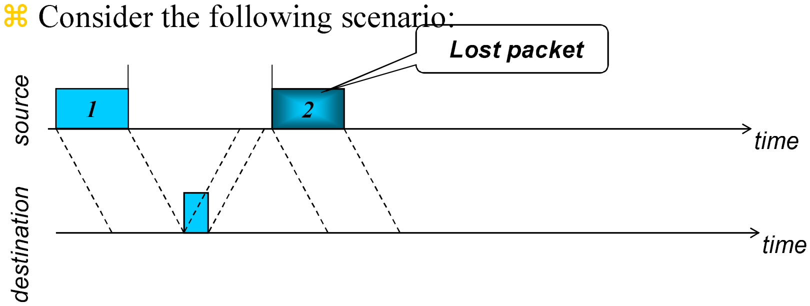
- Note that the destination has no way of identifying who an erroneous packet was destined to. Why?
- To achieve the ARQ goal, the ARQ techniques relay on three mechanisms:
  - Sequencing
  - Acknowledgements (ACK/NAK)
  - Timeout
- Each data packet and each control packet is labeled with a "unique" consecutive number. This allows to identify each packet as part of a sequence.

*Automatic **R**etransmission re**Q**uest (con't)*

- Once a data packet is received by a destination, the destination acknowledges the data packet by returning an short control packet, called *ACK.*
- If the destination can deduct that a packet was lost (how?), it can send a negative acknowledgement control packet (NAK), prompting the source to retransmit the lost data packet.
- A timer is associated with each packet that is transmitted by the source and set for a specific timer delay (timeout interval). When the timer expires, the packet is assumed loss and is retransmitted.

Automatic Retransmission reQuest (con't)

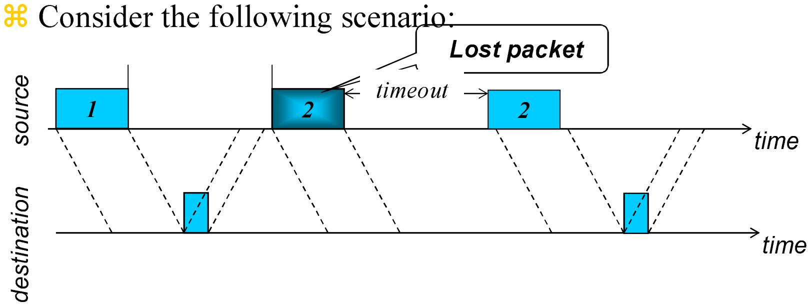
- Why do we need timeout?
- Consider the following scenario:



Packet #2 is never received. The process comes to a stop.
Solution: Timeout for each transmitted packet.

Automatic Retransmission reQuest (con't)

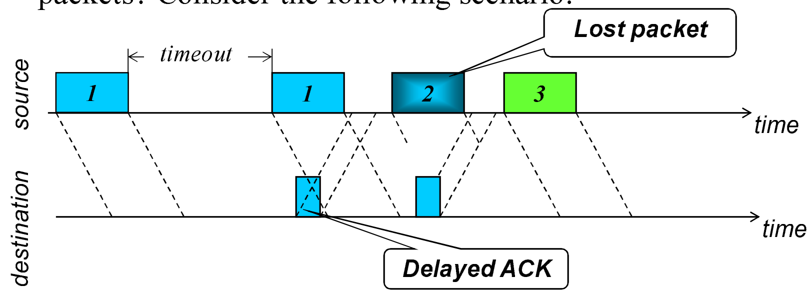
- Why do we need timeout?
- Consider the following scenario:



Packet #2 is never received. The process comes to a stop.
Solution: Timeout for each transmitted packet.

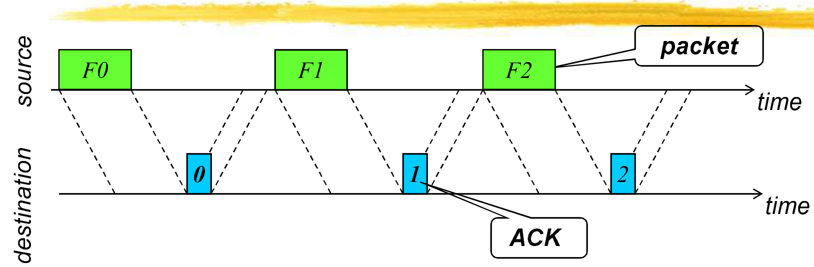
*Automatic **R**etransmission re**Q**uest (con't)*

- Why do we need sequence numbers on data and control packets? Consider the following scenario:



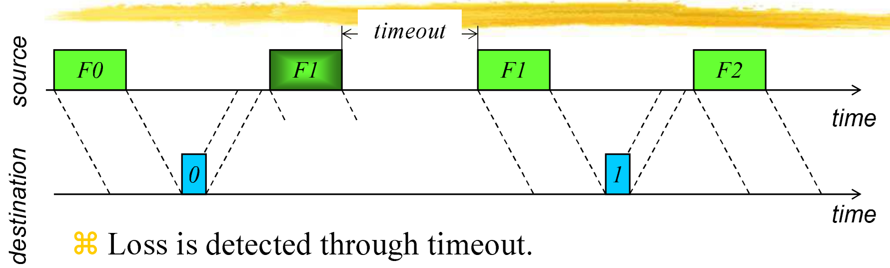
Packet #2 is never received ! Solution: ACKs have to be sequenced as well.

ARQ: Stop and Wait (S&W) protocol



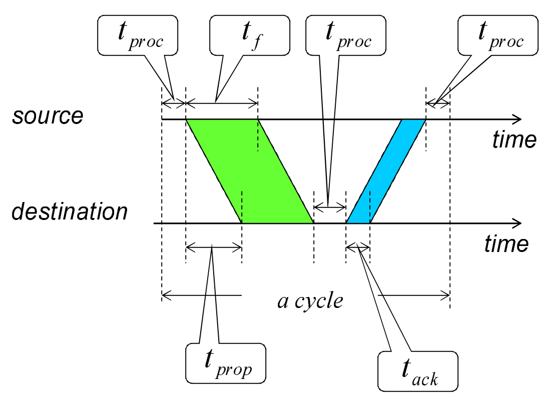
- The simplest ARQ protocol is the S&W.
- After sending a packet, the source waits for an acknowledgement (ACK) from the destination, at which time it sends the next packet.
- Only <u>one</u> packet is outstanding at any time.

ARQ: S&W protocol with errors



- Loss is detected through timeout.
- Since only one packet is outstanding at any time and ACKs are sequenced, there is no ambiguity of what packet needs to be retransmitted.
- Note that since only one packet is outstanding at each time, one bit is sufficient for sequencing in the S&W scheme.

ARQ: S&W utilization



ARQ: S&W utilization

⌘ Definitions:

- t_f - frame transmission time
- t_{prop} - propagation time
- t_{proc} - processing time
- t_{ack} - ACK transmission time

F - frame size [bits]

A - ACK size [bits]

C - channel data rate [bps]

⌘ A *cycle* contains one successful data packet transmission with a successful ACK:

$$\text{a } cycle \equiv t_0 = 2t_{prop} + 2t_{proc} + t_f + t_{ack} =$$

$$2(t_{prop} + t_{proc}) + F/C + A/C\,.$$

ARQ: S&W utilization

❑ During one cycle, only one frame is transmitted. So, the utilization is ($I = (t_{prop} + t_{proc})$; a is one-way propagation (+processing) delay):

❖ $U_{no-errors} = \frac{t_f}{t_0} = \frac{D/C}{(F+A)/C + 2(t_{prop}+t_{proc})} = \frac{D}{(F+A)+2C(t_{prop}+t_{proc})}$

❖ $U_{no-errors} = \frac{t_f}{t_0} = \frac{D/F}{(1+A/F)+2CI/F} \cong \frac{1}{1+2a}$ (assuming $D \cong F$, $A \cong 0$)

❖ To improve utilization, keep $a \ll 1$ ($c.f.$ MAC-layer protocols)

❖ Keeping $a \ll 1$ is often challenging (*e.g.*, satellite communications)

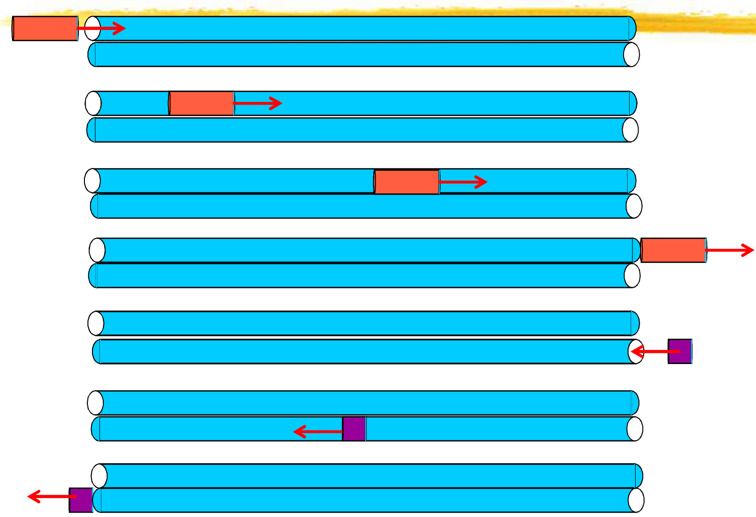
❖ Alternative considerations:

- ❖ Large D, Small $I \rightarrow$ Large U
- ❖ Also, Small H, Small $A \rightarrow$ Large U

ARQ: S&W utilization

- Example:
  - 1.5 [Mbps] link
  - $t_{prop} = 45\ [msec]$
  - F = 1 [KB}, t_f = 5.3 [msec]
  - $2a = 90/5.3 = 8.44$
  - $U \cong 1/(1 + 8.44) \cong 0.105 \ldots \cong 10.5\%$ utilization ☹
- The culprit is the fact that we cannot <u>fill up the pipe!</u>

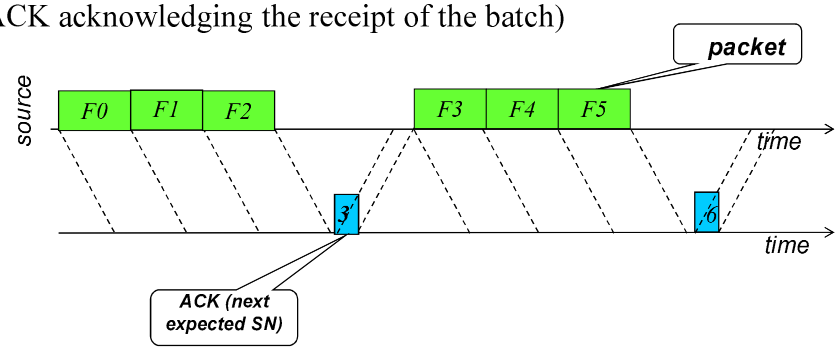
ARQ: S&W utilization



ARQ: Perfect utilization

Window-based Protocols

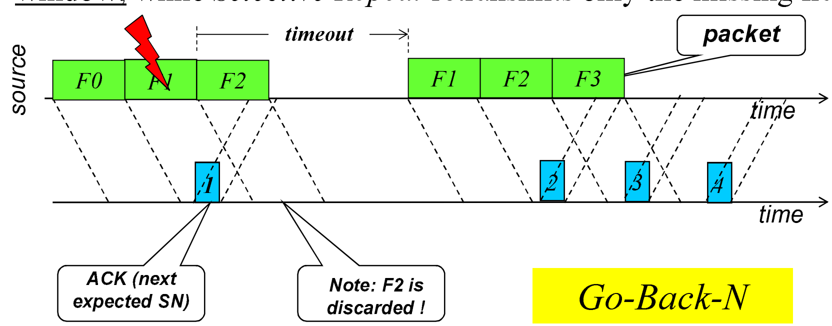
- The solution is to send more than one packet in a batch (with one ACK acknowledging the receipt of the batch)



- Two basic window-based protocols: *Go-Back-N* and *Selective-Repeat* (*"Selective ACK"* and *"Negative ACK"*)

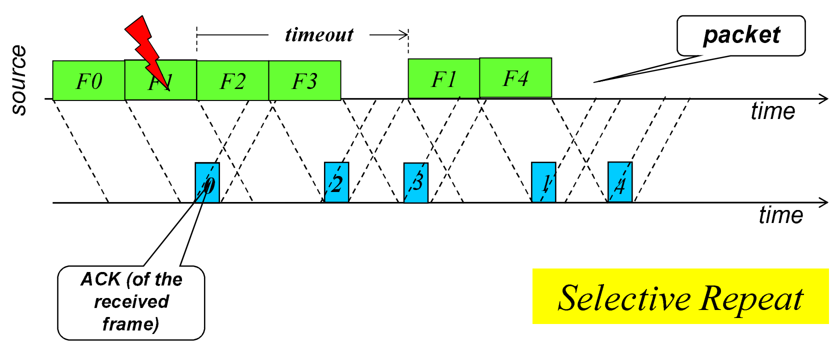
Window-based Protocols

- The basic difference between the two schemes is in how they handle errors.
- Upon detection of an error, *Go-Back-N* retransmits the whole window, while *Selective Repeat* retransmits only the missing frames.



Window-based Protocols

- The *Selective Repeat* scheme retransmits only the missing frames, so it has larger utilization than the *Go-Back-N* scheme.



ARQ: Window-based utilization

If there are no errors, Window-based protocols can achieve (almost) 100% utilization.

(1) Why "almost?

(2) What is the condition for (almost) 100% utilization? (under some assumptions: $W \geq 1 + 2a$)

(3) What happens in the presence of errors?