CprE 489
Lecture 11

Fall Semester 1997

Basic Simplex Protocol

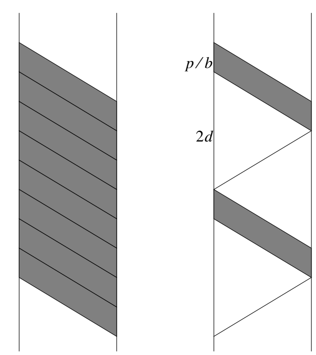
- Sender - repeat forever
  - get data from network layer
  - create frame
  - send frame to physical layer
- Receiver - repeat forever
  - wait for event {framearrival}
  - remove data from frame
  - put data to network layer
- Assume an errorless physical layer

- Problem: Receiver may be overwhelmed.

Stop-and-Wait Protocol

- Sender - repeat forever
  - get data from network layer
  - create frame
  - send frame to physical layer
  - wait for event {ackarrival}
- Receiver - repeat forever
  - wait for event {framearrival}
  - remove data from frame
  - put data to network layer
  - send ack frame to physical layer
- Problem solved: Receiver nor Sender can be overwhelmed.
- New Problem: Physical layer may be under-utilized

Under-Utilization



Simlex Protocol   Stop-and Wait Protocol

$$U = \frac{p/(p/b+2d)}{b} = \frac{1}{1+\frac{2db}{p}}$$

Example:
d = .001 sec.
b = 1 Mb/sec
p = 1,000 bits

U = 1/3

What about Errors

- If Simplex is modified to throw away bad frames then there will be gaps in data put to the network layer.
- If Stop-and-Wait is modified to throw away bad packets then either

  1. It hangs if no ack is sent

  2. There are gaps in the data put to the network layer if acks are always sent.

Retransmission

- Sender - repeat forever

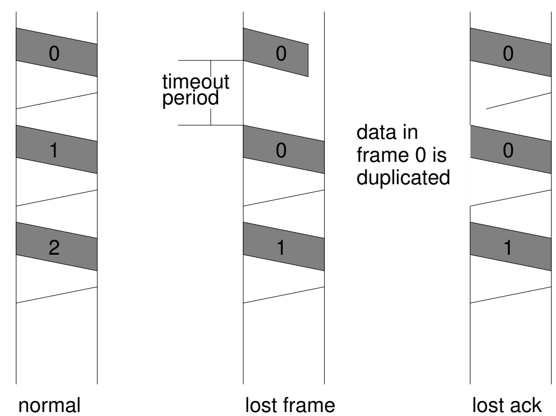
```
  get data from network layer
  create frame using error detection
  done := false
   repeat
     send frame to physical layer
     set timer
     wait for event {ackarrival, frameerror, timeout}
         if event = ackarrival then done := true  endif
   until done
endrepeat
```


Receiver

- Receiver - repeat forever

```
  wait for event {framearrival, frameerror}
    if event = framearrival then
      remove data from frame and put to network layer
      send ack to physical layer
    endif
endrepeat
```


Retransmission Executions



Positive Ack with Retransmission (PAR)

- Sender

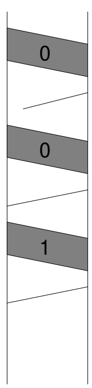
```
seq_no := 0
repeat forever
   get data from network layer
   create frame with seq_no field
   done := false
   repeat
      send frame to physical layer
      set timer
      wait for event {ackarrival, frameerror, timeout}
      if event = ackarrival then done := true  endif
   until done
   seq_no := 1 - seq_no
endrepeat
```


Receiver

- Receiver

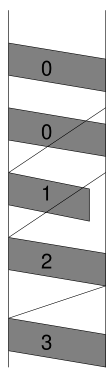
```
expected_seq_no = 0
repeat forever
   wait for event {framearrival, frameerror}
      if event = framearrival then
         if expected_seq_no = seq_no in frame  then
             remove data from frame and put to network layer
             expected_seq_no := 1 - expected_seq_no
         endif
      send ack to physical layer
      endif
endrepeat
```


PAR Execution 1 and Early Timeout Execution



data in this frame is ignored

lost ack



* put to network layer

This frame sent too soon

early timeout

Problems:
An early timeout can cause gaps in data put to network layer

Frame 1 appears to be acknowledged.

Frame 2 is ignored because it has an unexpected seq number.

Fixing PAR

- Put seq_no into the acknowledgements. Then the Sender would know exactly what is being acknowledged.