Selective-Reject Program Design

Daniel Jones

CPE 464

Dr. Hugh Smith

# Part One – Design Problems

1. Packet Flow Diagrams
   1. Normal Packet Flow

A close up of text on a black background

Description automatically generated

* + Clean termination after RR7 is received
  + Expected execution. Data packets can be sent before the RR’s are in as long as within window, but this examples behavior best.
  1. 1 packet lost, window size of 3

A close up of text on a black background

Description automatically generated

* Data3 is lost, SREJ is sent for Data4, server goes quiet and buffers rest of window.
* Once Data3 is resent, RR is sent that encompasses all good packets by server, window re-opens.
  1. Multiple packets lost in a window (Window size = 4, 8 total packets)

A close up of text on a white background

Description automatically generated

* 2 packets dropped, wait until next invalid packet to send selective rejects for missed packets.
* Server goes quiet but buffers good packets until lost packets are transmitted.
* Send lower RR for lowest lost packet then highest for buffered data.
  1. RR lost for packet 2 (Window size 3, 6 total packets)A screenshot of a cell phone

     Description automatically generated
* There is no lost functionality as long as the RR lost does not cause the window to close since the server only cares about the highest RR received.
* If window is closed without RR lost, resend the data packet necessary based on timeout from server.
  1. Multiple RR’s lost (Window size 3, 6 total packets)

A screenshot of a cell phone

Description automatically generated

* Same concept applied as D, same edge conditions. The server will continue sending packets as long as the window is open and recognize the latest RR.
  1. SREJ lost (Window Size = 3, Total Packets = 6)

A screenshot of a cell phone

Description automatically generated

* Data packet is lost and SREJ is lost, server will timeout from not receiving RR’s to move window.
* On timeout, server will send lower edge packet (and window size) over again.
  1. Entire window of data lost

A screenshot of text

Description automatically generated

* Same concept as F, dictate recovery by server timeout.
  1. Entire window of RR’s lost

A screenshot of a cell phone

Description automatically generated

* Good packets are kept by rcopy.
* Server will timeout after not receiving RR’s, so it will send back lower edge packet.
* However, the server will get an RR for the highest packet sent, and the window will move, and higher, non-duplicate packets will be sent.

1. Scenarios in which the window will close:

* If no packets are lost and transmission is too slow, the window will close, especially if it’s too small.
* The window can close if one or more of the data packets are lost. Based on the SREJ sent back, the lost data will be sent back, and an RR of the highest good packet will increase the window size to open the window and continue transmission.
* If a SREJ is lost, the window will close, and the server will timeout. The server will send the lowest packet in the window and will receive the SREJ that was lost (or at least try 10 times).
* If the entire window of RR’s is lost, then window will close. The server will timeout and resend the lowest packet and receive an RR for the last good packet.

1. A) You might receive a duplicate data frame number after a timeout and the server sends the lowest data packet in the window. Rcopy should ignore the packet and RR or SREJ the appropriate packet.

B) If the data frame is higher than expected but still in a window, that means the data packet has been lost. SREJ the packet, go quiet but keep accepting packets and holding RR’s/SREJ’s, and respond to the client once it gets the lost packet.

C) If the data frame is the one expected but you have already received frames with higher numbers, then you have SREJ’d that packet and now you can flush the buffer to disk as far as the highest RR.

1. A) Walk through the scenario: send frame 0, 1, 2 and then receive RR 1, receive RR 3, send frame 3 and 4. (window size = 3)

B) Walk through the scenario: send frame 0, 1, receive RR 1, send frames 2, 3, receive SREJ 2, resend 2, receive RR 4. (window size = 3)

1. Assume your program receives data frames 1,2,3,4,5 and then receives frame 7 and 8 and then frames 10, 11 and window size of 3.

A screenshot of a cell phone

Description automatically generated

When packet is 6 is lost, rcopy will send a SREJ in response to packet 7 and buffer until packet 6 is received. Then, rcopy will flush its buffer and get ready for a new window by sending the highest available RR and the server will send more data. Same procedure will occur when packet 9 is lost.

1. A)

A screenshot of a cell phone

Description automatically generated

Sender Info Per Packet: (1, 1, 6), (1, 2, 6), (1, 3, 6), (2, 4, 7), (2, 5, 7), (2, 6, 7), (3, 7, 8),

(6, 8, 11), (6, 8, 11), (9, 9, 14), (9, 10, 14)

B)

A screenshot of a cell phone

Description automatically generated

Sender Info Per Packet: (1, 1, 6), (1, 2, 6), (1, 4, 6), (1, 5, 6), (1, 1, 6), (6, 6, 11) -> (6, 9, 11)

C)

A screenshot of a cell phone

Description automatically generated

Sender Info Per Packet: (1, 1, 6), (1, 2, 6), (1, 4, 6), (1, 5, 6) , (1, 3, 6), (6, 6, 11), (6, 7, 11),

(6, 8, 11), (6, 6, 11)

1. A)

A close up of a logo

Description automatically generated

B)

A screenshot of a cell phone

Description automatically generated

C)

A screenshot of a cell phone

Description automatically generated

D)

A screenshot of a cell phone

Description automatically generated

E)

A screenshot of a cell phone

Description automatically generated

F)

A screenshot of a cell phone

Description automatically generated

G)

A screenshot of a cell phone

Description automatically generated

1. A)

A screenshot of a cell phone

Description automatically generated

B)

A screenshot of a cell phone

Description automatically generated

C)

A screenshot of a cell phone

Description automatically generated

**Part II – State Diagrams**

RCOPYA picture containing text, map

Description automatically generated

SERVER

A picture containing text, map

Description automatically generated