CSE123 Discussion 2

Oct 10th, 2017

Sliding Window Protocol
Sender

LAR - Sequence number of last acknowledgement received, defines lower extreme of window
LFS - Sequence number of last frame sent, defines upper extreme of window
Window is from [LAR+1, LFS], that is all frames that have been sent but not yet acked
SWS - Bound on sliding window size
Frame Sequence Number in Window

CASE 1: Usual Case
LAR <= LFS

CASE 2: Sequence Number Wrap Around
LAR > LFS

LAR <= LFS && seqNo > LAR && seqNo <= LFS
LAR > LFS && (seqNo > LAR || seqNo <= LFS)

On Sender with SWS = 4, [0,7] sequence numbers
Receiver

NFE - Next frame expected
LFR - Sequence number of largest consecutive frame received
LAF - Sequence number of largest acceptable frame
RWS - Max receive window size
LFR = NFE - 1
LAF = NFE + RWS - 1
Frame Sequence Number in Window

CASE 1: Usual Case
NFE + RWS - 1 >= NFE
NFE + RWS - 1 >= NFE && seqNo >= NFE && seqNo <= NFE + RWS - 1

Remember NFE is just LFR + 1 and LAF is just NFE + RWS - 1.

Green sequence numbers are in window and grey are outside.

CASE 2: Sequence Number Wrap Around
NFE + RWS - 1 < NFE
NFE + RWS - 1 < NFE && (seqNo >= NFE || seqNo <= NFE + RWS - 1)

On Receiver with RWS = 4, [0,7] sequence numbers
Frame Not in Window On Receiver

Send ack that acknowledges NFE - 1
This tells the sender that we have
   successfully received all frames up to NFE -
   1 = LFR.
Will happen when ack is lost and needs to be resent
Circular Send / Receive Window

Implement send and receive queue as circular array or list
Index in to sender’s send queue using sequence number % SWS
Index in to receiver’s receive queue using sequence number % RWS
Used in book’s code for SWP