Mark Habashy

Jeeho Ahn

selective repeat

sender sends 5 packets at a time with a timeout for each of them, if sender doesn't receive the ack for the packet within the timeout period it resends the packet

sequence number starts at 0 till 9

packet timeout is set to 1 second

window size for sender is set to 5

window size for receiver is set to 5

if receiver receives a packet within its window it sends the sender an ack packet for the packet it receives. if the packet did not lie within the window it discards it

IMPORTANT: data transfer is all done in one serversocket.accept() session (only one socket is opened used throughout the session)

to use:
python sender.py
python receiver.py

to test:

set the variable ignore_3_once to True. This makes the sender add the packet 3 to the time out dictionary but never sends it to mimic a lost packet.
after the timeout of one second it resends it.

go back n

IMPORTANT PER PROFESSOR ALDEER AND SLIDE 75 OF LECTURE 3 - 4 (A SCREENSHOT IS FOUND IN GO BACK N DIRECTORY BY THE NAME SLIDE 75):

The receiver only acks the highest-numbered frames received in sequence (THIS IS WHAT IS

IMPLEMENTED ON THE RECEIVERS SIDE)

sender sends 5 packets at a time. The receiver only acks the highest-numbered frames received in sequence.
If the received a packed it wasn't expecting it acks the highest-numbered packet received in sequence.
the sender sends the packet having sequence number = ack sequence + 1

sequence number starts at 0 till 9

window size for sender is set to 5

window size for receiver is set to 1

IMPORTANT: data transfer is all done in one serversocket.accept() session (only one socket is opened used throughout the session)

to use:
python sender.py
python receiver.py

to test:

set the variable ignore_3_once to True. This makes the sender add the packet 3 to the time out dictionary but never sends it to mimic a lost packet.
after the timeout of one second it resends it.

selective repeat

sender sends 5 packets at a time with a timeout for each of them, if sender doesn't receive the ack for the packet within the timeout period it resends the packet

sequence number starts at 0 till 9

packet timeout is set to 1 second

window size for sender is set to 5

window size for receiver is set to 5

if receiver receives a packet within its window it sends the sender an ack packet for the packet it receive. if the packet did not lie within the window it discards it

IMPORTANT: data transfer is all done in one serversocket.accept() session (only one socket is opened used throughout the session)

to use:
python sender.py
python receiver.py

to test:

set the variable ignore_3_once to True. This makes the sender add the packet 3 to the time out dictionary but never sends it to mimic a lost packet.
after the timeout of one second it resends it.