Name: _____

CS 325 Computer Networks Fall 2019

Exam #2

Closed book and notes. This exam should have five pages.

Problem 1: [30 Points] Answer the following questions as concisely as possible.

a) In TCP, what is a "Triple-ACK?" What does it indicate and how does TCP respond?

b) In the Selective Repeat sliding window protocol, if the receiver's window is a different size than the sender's, is it a problem? Does it lead to correctness issues? Efficiency issues? Other? Explain.

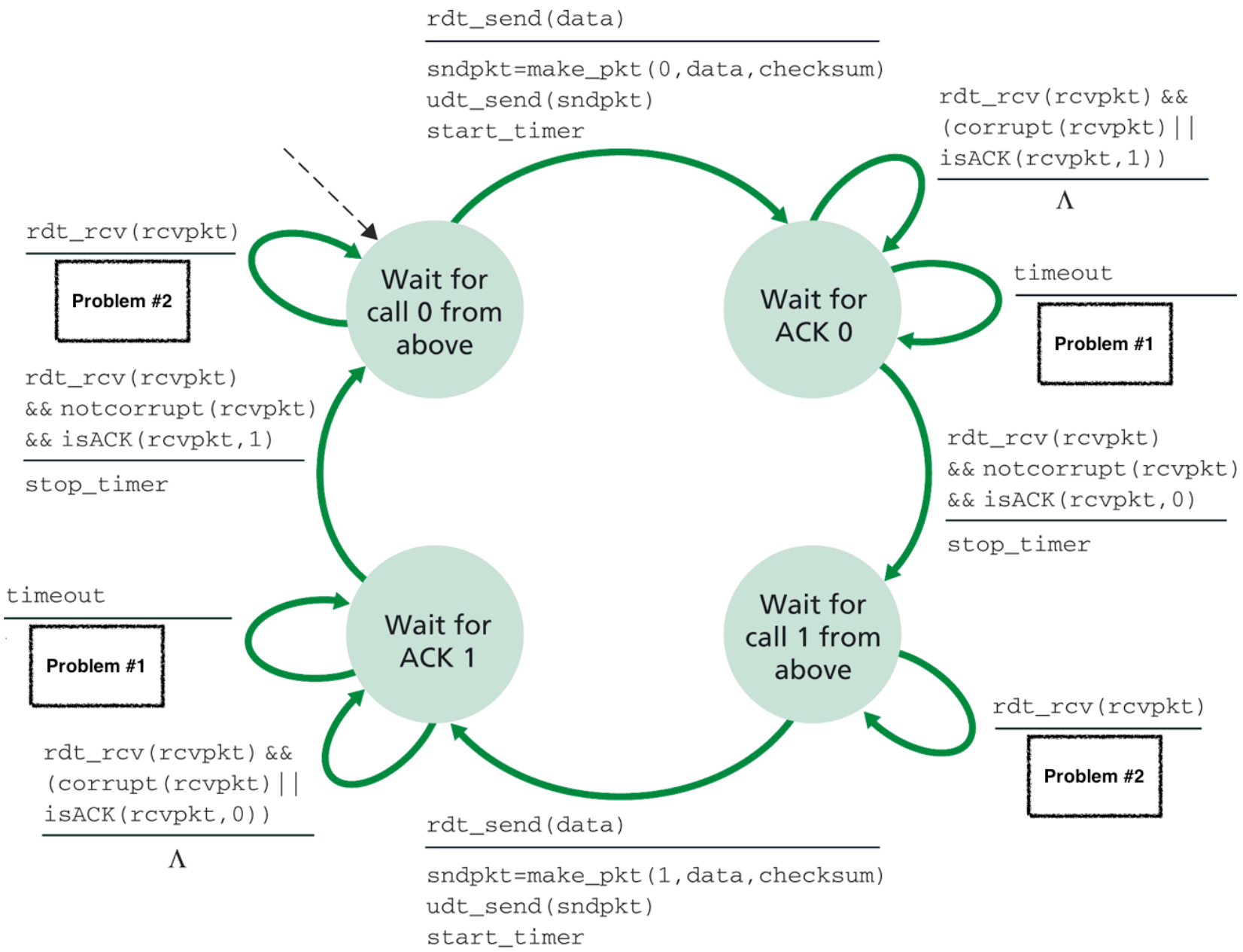
c) Wireless Ethernet goes to a lot of trouble to do reliable delivery of data, so why do we have to do it all over again in the Transport layer?

Problem 2: [30 points] The diagram on the next page specifies a sender's behavior in one of the reliable transport-layer protocols from the book, with portions of some of the transition details removed. Feel free to remove the diagram page from your exam as you answer the questions below.

a) In each of the "Wait for ACK" states there's a transition that handles timeout events. What belongs in the "Problem #1" boxes in those transitions? If it's different for each timeout transition, please show both. Use the same notation used on the rest of the diagram when listing the missing actions.

b) In each of the "Wait for call" states there's a transition that handles `rdt_rcv` events. What belongs in the "Problem #2" boxes in those transitions? If it's different for each transition, please show both. Use the proper notation when listing the missing actions.

c) When your great aunt looked at the diagram, she was shocked to see that it didn't use NAKs. "How can a protocol guarantee reliable delivery without NAKs?", she asks. Use the space below to provide your response:


rdt_send(data)
sndpkt=make_pkt(0,data,checksum)
udt_send(sndpkt)
start_timer
rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt)||
isACK(rcvpkt,1))
Λ
rdt_rcv(rcvpkt)
Problem #2
timeout
Problem #1
Wait for
call 0 from
above
Wait for
ACK 0
rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& isACK(rcvpkt,1)
stop_timer
rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& isACK(rcvpkt,0)
stop_timer
timeout
Problem #1
Wait for
ACK 1
Wait for
call 1 from
above
rdt_rcv(rcvpkt)
Problem #2
rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt)||
isACK(rcvpkt,0))
Λ
rdt_send(data)
sndpkt=make_pkt(1,data,checksum)
udt_send(sndpkt)
start_timer

Problem 3: [30 Points]

a) When we discussed TCP, we talked about its Additive Increase Multiplicative Decrease mechanism. What is it that TCP is adjusting via AIMD? Why?

b) What if all TCP implementations used an Additive Increase Additive Decrease approach instead? (That is, it increases by regularly adding a fixed amount, and decreases by subtracting a similar amount.) Apparently that won't work as well, or TCP would use AIAD instead. Describe some key disadvantages of using AIAD.

c) Is AIMD appropriate for UDP? Explain why or why not.

Problem 4: [10 Points]

In the 802.11 implementation project, what is the role of the `LinkLayer` class's `send()` method? Who calls it, and what does it do once called? (I only want to know what happens in `send`, not about actions that might subsequently take place in other parts of the code.)