Lektion 8 (31-03-2020)

## Review 7

**Suppose a process in Host C has a UDP socket with port number 6789. Suppose both Host A and Host B each send a UDP segment to Host C with destination port number 6789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?**

Ja den kan godt kende forskel, fordi at ”sourcen” bliver sendt med, så den kan se at de kommer fra to forskellige steder. I tilfældet med UDP så er det kun ”source” socketen som bliver sendt med, ved TCP så er det også IP’en som bliver sendt med.

## Review 8

**Suppose that a Web server runs in Host C on port 80. Suppose this Webserver uses persistent connections, and is currently receiving requests from two different Hosts, A and B. Are all of the requests being sent through the same socket at Host C? If they are being passed through different sockets, do both of the sockets have port 80? Discuss and explain.**

Ja de vil begge blive modtaget i den samme port, nemlig port 80. Den bruger ”source” informationen til at kende forskel på de datapakker som den modtager.

## Review 12

**Visit the Go-Back-N Java applet at the companion Web site.**

1. **Have the source send five packets, and then pause the animation before any of the five packets reach the destination. Then kill the first packet and resume the animation. Describe what happens.**

Fordi at man burger Go-Back-N så bliver alle de andre packer ”tabt”, fordi at den kun modtager pakkerne i en bestemt rækkefølge, hvilket vil sige at den skal have den første pakke før den anden pakke.

1. **Repeat the experiment, but now let the first packet reach the destination and kill the first acknowledgment. Describe again what happens.**

Den første pakke bliver stadig ”acknowledged” på grund af den måde Go-Back-N virker på, fordi at for at clienten sender en ”acknowledgement” på pakke 2, så ved man at den også har modtaget pakke 1, fordi at ellers ville den ikke godtage pakke 2.

1. **Finally, try sending six packets. What happens?**

Dette er ikke muligt, fordi at vinduet kun har en størrelse på 5, så man bliver nødt til at vente til der kommer en åbning i vinduet.

## Review 13

Repeat R12, but now with the Selective Repeat Java applet. How are Selective Repeat and Go-Back-N different?

1. **Have the source send five packets, and then pause the animation before any of the five packets reach the destination. Then kill the first packet and resume the animation. Describe what happens.**

Så gemmer clienten stadig de andre packets og venter på nummer 1 bliver gensendt, før den kan sende det videre til app laget. Efter at timeren er udløbet vil den første packet blive gensendt hvorefter en ”acknowledgement” vil blive sendt for den.

1. **Repeat the experiment, but now let the first packet reach the destination and kill the first acknowledgment. Describe again what happens.**

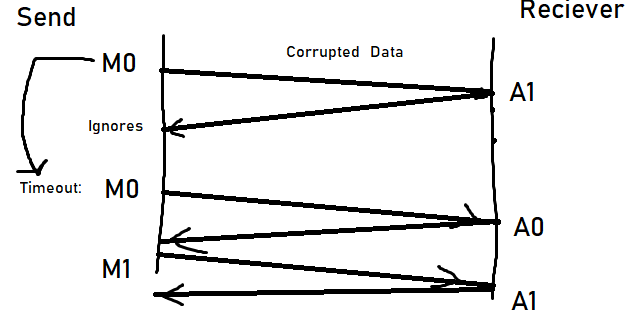
Her sker circa det samme som ved den ovenfor, efter et stykke tid vil pakken blive gensendt og en ny ACOK vil blive sendt og modtaget denne gang.

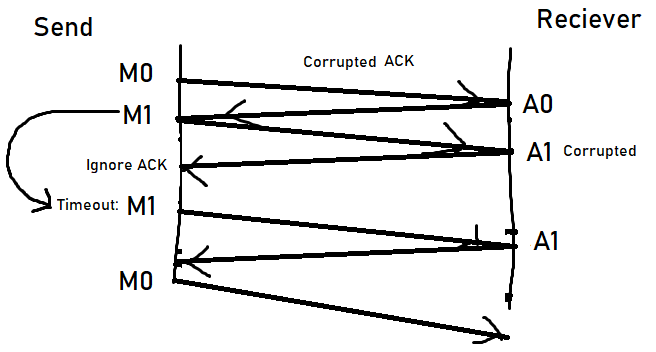
1. **Finally, try sending six packets. What happens?**

Dette er heller ikke muligt her, fordi at vinduet kun har en størrelse på 5.

## Problem 9

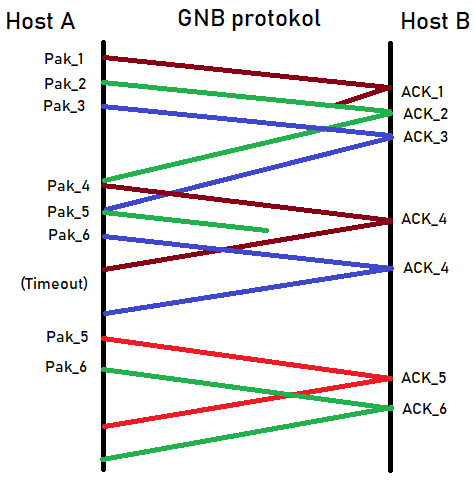
**Give a trace of the operation of protocol rdt3.0 when data packets and acknowledgment packets are garbled. Your trace should be similar to that used in Figure 3.16.**





## Problem 19

**Suppose Host A and Host B use a GBN protocol with window size *N* 5 3 and a long-enough range of sequence numbers. Assume Host A sends six application messages to Host B and that all messages are correctly received, except for the first acknowledgment and the fifth data segment. Draw a timing diagram (similar to Figure 3.22), showing the data segments and the acknowledgments sent along with the corresponding sequence and acknowledge numbers, respectively.**

GNB:  


### SR:

