Solutions Networking 1

R1,R19, R23, R24

R1

There is no difference. Throughout this text, the words “host” and “end system” are used interchangeably. End systems include PCs, workstations, Web servers, mail servers, PDAs, Internet-connected game consoles, etc

R19

19. a) 500 kbps
b) 64 seconds
c) 100kbps; 320 seconds

R23

The five layers in the Internet protocol stack are – from top to bottom – the application layer, the transport layer, the network layer, the link layer, and the physical layer. The principal responsibilities are outlined in Section 1.5.1.

R24

Encapsulation is the process of passing a packet from a higher layer to a lower layer. In the simplest case, it simply appends additional information (i.e., a *header*) to the source packet. In more complicated scenarios, the original packet can be split into multiple packets, each carrying its own header. Decapsulation is the opposite process of encapsulation. It extracts the header from a source packet from a lower layer and passes the payload to the higher layer. If lower layer packets are part of a sequence, the corresponding payloads are put together before they are passed to the higher layer. Each protocol in a layer of a protocol stack relies on the services of the lower layers but not on their information. Thus, the information a protocol needs to process a packet should entirely be contained in the header of that protocol. Encapsulation and decapsulation are flexible mechanisms to allow each protocol to operate independently from others while being able to interface with each other.

Øvelser

1) http://gaia.cs.umass.edu/kurose_ross/interactive/end-end-delay.php

(findes online på siden)

2) http://gaia.cs.umass.edu/kurose_ross/interactive/end-end-throughput.php

(findes online på siden)

3)

Problem 6

a) $d_{prop} = m / s$ seconds.

b) $d_{trans} = L / R$ seconds.

c) $d_{end-to-end} = (m / s + L / R)$ seconds.

d) The bit is just leaving Host A.

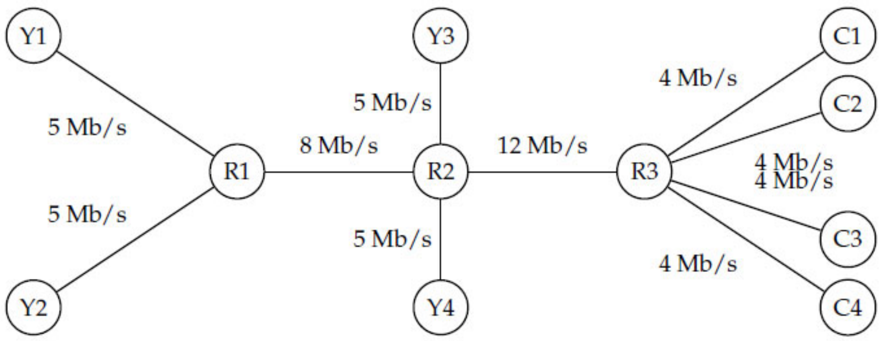
e) The first bit is in the link and has not reached Host B.

f) The first bit has reached Host B.

g) Want

$$m = \frac{L}{R} s = \frac{120}{56 \times 10^3}\left(2.5 \times 10^8\right) = 536 \text{ km}.$$

4)



The network shows four servers Y1 to Y4 which host video content and four clients C1 to C4 who are concurrently downloading the video content using FTP from each of these servers (C1 from Y1, C2 from Y2, C3 from Y3 and C4 from Y4). The network is connected using routers R1, R2 and R3 and the transmission rates of the individual links have been outlined in the figure. The end to end throughput experienced by the clients for their download is

☐ a) 2 Mb/s

☐ b) 3 Mb/s

☐ c) 4 Mb/s

☐ d) 5 Mb/s

Solution: B)

5)

P25:

a) BDProduct= (20e6 / 2.5e8)*5e6 bits= 400000 bits
b) BD product= 400000 bits
c) The bandwidth-delay product of a link is the maximum number of bits that can be in the link.
d) The width of a bit = length of link / bandwidth-delay product, so 1 bit is 50 meters long.
e) s/R

Message is 10^6 bits
Broken into 100 packets of 10.000 bits = 10^4 bits
Each link is 5 mpbs = 5*10^6 bps

Problem 31

a) Time to send message from source host to first packet switch $=\frac{\mathbf{10^6}}{\mathbf{5\times10^6}} = \mathbf{0.2}\text{sec}$ With store-and-forward switching, the total time to move message from source host to destination host $= \mathbf{0.2}\boldsymbol{sec} \times \mathbf{3}\boldsymbol{hops} = \mathbf{0.6}\boldsymbol{sec}$

b) Time to send 1st packet from source host to first packet switch = $\frac{1\times10^4}{2\times10^6}\text{sec} = 5\,m\,\text{sec}$. Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = $2\times 5m\,\text{sec} = 10\,m\,\text{sec}$

10^4 / 5*10^6= 2 ms

2*2 ms = 4ms

c) Time at which 1st packet is received at the destination host = $5\,m\,\text{sec}\times 3\,hops = 15\,m\,\text{sec}$. After this, every 5msec one packet will be received; thus time at which last (800th) packet is received = $15\,m\,\text{sec} + 799 * 5m\,\text{sec} = 4.01\,\text{sec}$. It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).

3*2 ms = 6ms

6ms+99*2ms= 204 ms = 0,2 s

d)

i. Without message segmentation, if bit errors are not tolerated, if there is a single bit error, the whole message has to be retransmitted (rather than a single packet).
ii. Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and suffer unfair delays.

e)

i. Packets have to be put in sequence at the destination.
ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

6) Et streaming firma skal have uploadet et ny datasæt på 40 terabytes til en server, der er placeret tæt hos forbrugerne, men et stykke væk fra firmaet. Deres Internet forbindelse til serveren tillader en gennemsnitlig upload hastighed på 100 Mbps. Hvor lang tid tager det? Sammenlig tid og pris med at sende en fysisk pakke med et speditionsfirma med næste-dags levering. Antag firmaet køber en dedikeret forbindelse til serveren, med 10 gange højere kapacitet. Hvor lang tid tager det så? Hvad bliver den gennemsnitlige udnyttelsesgrad af denne, under antagelse af at et nyt datasæt uploades en gang om måneden, og den daglige trafik (email, web-surfing, etc) udgør 20 Mbps i gennemsnit. Overvej de praktiske konsekvenser i scenariet.

40 terabytes = 40 * 10^{12} * 8 bits. So, if using the dedicated link, it will take 40 * 10^{12} * 8 / (100 *10^6) =3200000 seconds = 37 days.

1GB forbindelse 3.7 dage.

Udnyttelsesgrad (ex måned): 3.7 dage med 100% + 30 dage med 20Mbps/1Gbps % / 30dage) = 4.3/30% = 14%

Med fysisk forsendelse med næste-dag-levering, kan du garantere at data er klar næste dag til en pris (for international forsendelse) på 500Kr. Send så store data-mængder i en kasse med harddiske vha. et shipping firma!

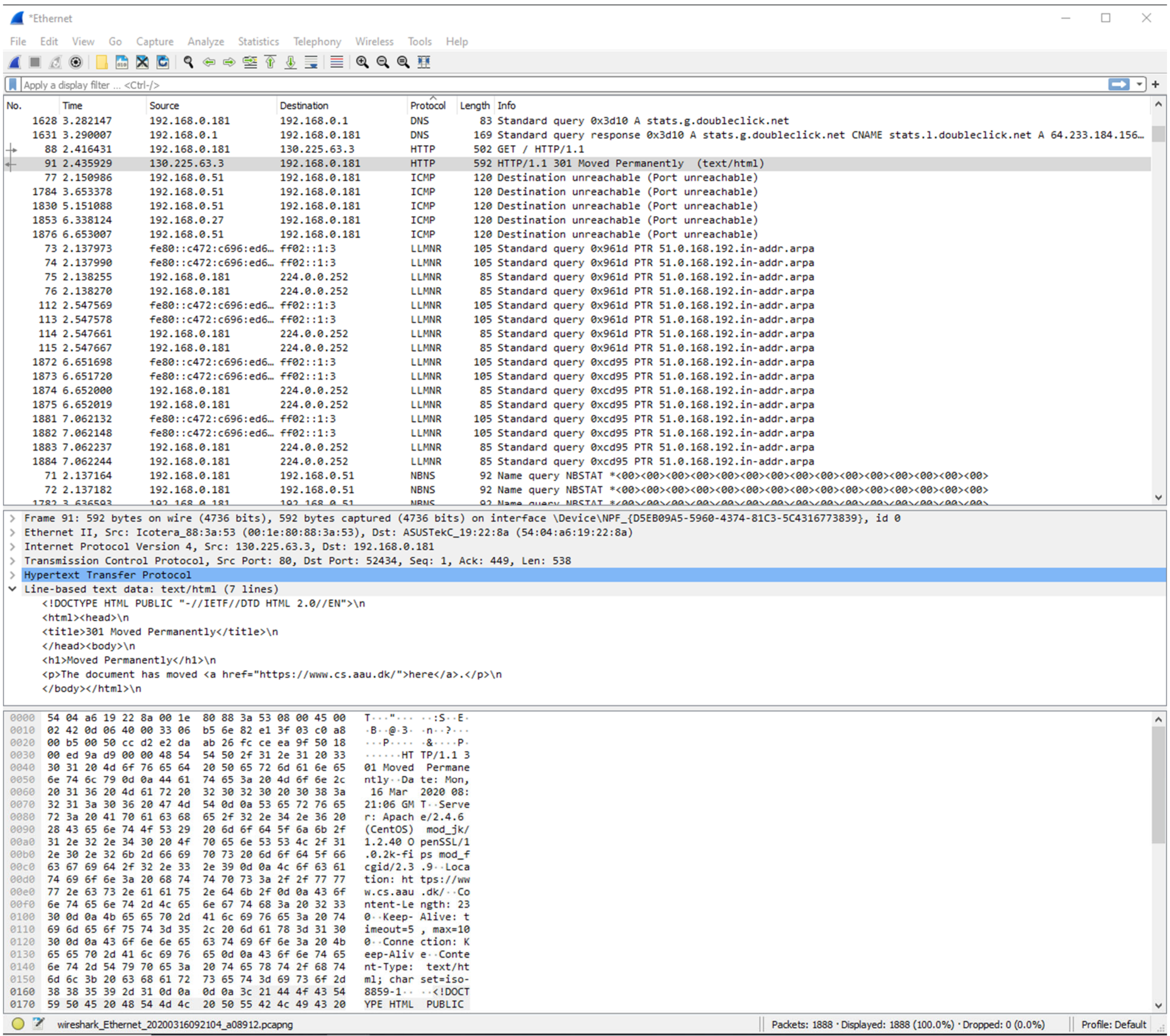
https://what-if.xkcd.com/31/

Praktiske

- P105 Wireshark Lab.
- Nedenfor har jeg opsnappet en kommunikation til http://www.cs.aau.dk (GET). Reponset på dette er en kode ”301” (Redirect) som fortæller at dokumentet er flyttet til et andet URL, som angiver at HTTPS protokollen skal anvendes. Browserens reaktion er at genindlæse dokumentet ved brug af den nye URL.

  I det midterste vindue er det muligt at udfolde og se headerne på alle niveauer i stakken; her: Ethernet, IP, TCP, http. Prøv at bemærke ”enveloping”/”encapsulation” (I skal ikke forstå indholdet af headerne endnu).

  Bemærk at der bliver sendt og modtaget ganske mange pakker selv i et kort scenarie, så det er normal nødvendigt at sætte filtre op, så man kun får vist de pakker der er interessante for det I kigger efter.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1628 3.282147 192.168.0.181 192.168.0.1 DNS 83 Standard query 0x3d10 A stats.g.doubleclick.net
1631 3.290007 192.168.0.1 192.168.0.181 DNS 169 Standard query response 0x3d10 A stats.g.doubleclick.net CNAME stats.l.doubleclick.net A 64.233.184.156…
88 2.416431 192.168.0.181 130.225.63.3 HTTP 502 GET / HTTP/1.1
91 2.435929 130.225.63.3 192.168.0.181 HTTP 592 HTTP/1.1 301 Moved Permanently (text/html)
77 2.150986 192.168.0.51 192.168.0.181 ICMP 120 Destination unreachable (Port unreachable)
1784 3.653378 192.168.0.51 192.168.0.181 ICMP 120 Destination unreachable (Port unreachable)
1830 5.151088 192.168.0.51 192.168.0.181 ICMP 120 Destination unreachable (Port unreachable)
1853 6.338124 192.168.0.27 192.168.0.181 ICMP 120 Destination unreachable (Port unreachable)
1876 6.653007 192.168.0.51 192.168.0.181 ICMP 120 Destination unreachable (Port unreachable)
73 2.137973 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
74 2.137990 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
75 2.138255 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
76 2.138270 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
112 2.547569 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
113 2.547578 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
114 2.547661 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
115 2.547667 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0x961d PTR 51.0.168.192.in-addr.arpa
1872 6.651698 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
1873 6.651720 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
1874 6.652000 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
1875 6.652019 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
1881 7.062132 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
1882 7.062148 fe80::c472:c696:ed6… ff02::1:3 LLMNR 105 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
1883 7.062237 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
1884 7.062244 192.168.0.181 224.0.0.252 LLMNR 85 Standard query 0xcd95 PTR 51.0.168.192.in-addr.arpa
71 2.137164 192.168.0.181 192.168.0.51 NBNS 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>
72 2.137182 192.168.0.181 192.168.0.51 NBNS 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>
Frame 91: 592 bytes on wire (4736 bits), 592 bytes captured (4736 bits) on interface \Device\NPF_{D5EB09A5-5960-4374-81C3-5C4316773839}, id 0
Ethernet II, Src: Icotera_88:3a:53 (00:1e:80:88:3a:53), Dst: ASUSTekC_19:22:8a (54:04:a6:19:22:8a)
Internet Protocol Version 4, Src: 130.225.63.3, Dst: 192.168.0.181
Transmission Control Protocol, Src Port: 80, Dst Port: 52434, Seq: 1, Ack: 449, Len: 538
Hypertext Transfer Protocol
Line-based text data: text/html (7 lines)
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
<html><head>\n
<title>301 Moved Permanently</title>\n
</head><body>\n
<h1>Moved Permanently</h1>\n
<p>The document has moved here.</p>\n
</body></html>\n
0000 54 04 a6 19 22 8a 00 1e 80 88 3a 53 08 00 45 00
0010 02 42 0d 06 40 00 33 06 b5 6e 82 e1 3f 03 c0 a8
0020 00 b5 00 50 cc d2 e2 da ab 26 fc ce ea 9f 50 18
0030 00 ed 9a d9 00 00 48 54 54 50 2f 31 2e 31 20 33 ······HT TP/1.1 3
0040 30 31 20 4d 6f 76 65 64 20 50 65 72 6d 61 6e 65 01 Moved Permane
0050 6e 74 6c 79 0d 0a 44 61 74 65 3a 20 4d 6f 6e 2c ntly··Da te: Mon,
0060 20 31 36 20 4d 61 72 20 32 30 32 30 20 30 38 3a 16 Mar 2020 08:
0070 32 31 3a 30 36 20 47 4d 54 0d 0a 53 65 72 76 65 21:06 GM T··Serve
0080 72 3a 20 41 70 61 63 68 65 2f 32 2e 34 2e 36 20 r: Apach e/2.4.6
0090 28 43 65 6e 74 4f 53 29 20 6d 6f 64 5f 6a 6b 2f (CentOS) mod_jk/
00a0 31 2e 32 2e 34 30 20 4f 70 65 6e 53 53 4c 2f 31 1.2.40 O penSSL/1
00b0 2e 30 2e 32 6b 2d 66 69 70 73 20 6d 6f 64 5f 66 .0.2k-fi ps mod_f
00c0 63 67 69 64 2f 32 2e 33 2e 39 0d 0a 4c 6f 63 61 cgid/2.3 .9··Loca
00d0 74 69 6f 6e 3a 20 68 74 74 70 73 3a 2f 2f 77 77 tion: ht tps://ww
00e0 77 2e 63 73 2e 61 61 75 2e 64 6b 2f 0d 0a 43 6f w.cs.aau .dk/··Co
00f0 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 32 33 ntent-Le ngth: 23
0100 30 0d 0a 4b 65 65 70 2d 41 6c 69 76 65 3a 20 74 0··Keep- Alive: t
0110 69 6d 65 6f 75 74 3d 35 2c 20 6d 61 78 3d 31 30 imeout=5 , max=10
0120 30 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 0··Conne ction: K
0130 65 65 70 2d 41 6c 69 76 65 0d 0a 43 6f 6e 74 65 eep-Aliv e··Conte
0140 6e 74 2d 54 79 70 65 3a 20 74 65 78 74 2f 68 74 nt-Type: text/ht
0150 6d 6c 3b 20 63 68 61 72 73 65 74 3d 69 73 6f 2d ml; char set=iso-
0160 38 38 35 39 2d 31 0d 0a 0d 0a 3c 21 44 4f 43 54 8859-1·· ··<!DOCT
0170 59 50 45 20 48 54 4d 4c 20 50 55 42 4c 49 43 20 YPE HTML PUBLIC
wireshark_Ethernet_20200316092104_a08912.pcapng
Packets: 1888 · Displayed: 1888 (100.0%) · Dropped: 0 (0.0%)
Profile: Default