

Operating Systems and Networks

Solution 6

Note: these solutions constitute material *supplemental* to the exercise sessions.

1 Traceroute: Theory

Consult the man page for traceroute (`man traceroute` from the Linux command line or on the web at <http://linux.die.net/man/8/traceroute>). Consider the following traceroute output:

```
[davs@pyxis]:~$ traceroute -n ethz.ch
traceroute to ethz.ch (129.132.128.139), 30 hops max, 60 byte packets
 1  192.33.93.1  0.822 ms  0.856 ms  0.893 ms
 2  10.10.1.81  1.222 ms  1.267 ms  1.265 ms
 3  10.1.17.242  1.252 ms  0.988 ms  1.117 ms
 4  192.33.92.185  1.815 ms  1.891 ms  1.778 ms
 5  * * *
 6  * * *
```

Figure 1: Traceroute output

- a) What does each of the three measurements next to each IP address on lines 1-4 mean?

Answer: Each packet is sent 3 times. Each of the measurements is the time measured for that packet.

- b) Give one possible reason why hops 5 and 6 show “* * *”

Answer: The hop did not reply with TIME EXCEEDED messages.

2 Data Transmission

A direct Swiss flight between Zürich and New York takes 8h54m. You get on this flight, and take with you a 4 terabyte hard drive. Assume 4 terabytes is 32 terabits ($3.2 \cdot 10^{13}$ bits). Once you land in New York:

- a) What was the data transmission rate (in Gigabits/s)?
- b) What was the propagation delay?
- c) How long would it take to transfer the same amount of data on a typical home fiber connection (assume a data transfer rate of 100 megabits/s)?

- d) Since the flight time is constant for any amount of data (1 byte or 100 terabytes), up to how much data does it make sense to transfer over the fiber connection?

Answer:

- a) Data: $3.2 \cdot 10^{13}$ bits. Time $3.2 \cdot 10^4$ seconds. Rate: 10^9 bits/s = 1 Gbps.
b) 32000 seconds
c) $3.2 \cdot 10^{13}$ bits / 10^8 bits/s = $3.2 \cdot 10^5$ s = 3.7 days
d) The most data you can send in 32 000 seconds is 3.2 terabits, or 0.4 terabytes, or 400 gigabytes. More data than this would take longer than 8h54min, so you get a better transmission rate by flying.

3 Bandwidth and delay

Suppose a 100 Mbps point-to-point link is being set up between Earth and a new lunar colony. The distance from the moon to Earth is 385,000 km and data travels over the link at the speed of light ($3 \cdot 10^8$ m/s)

- a) Calculate the minimum delay for the link.
b) Calculate the bandwidth-delay product for the link.
c) A camera on the lunar base takes pictures of Earth and saves them in digital format to disk. Suppose Mission Control on Earth wishes to download the most current image (25 MB). What is the minimum amount of time that will elapse between when the request for the data goes out and the transfer is finished?

Answer:

- a) Time for the data to travel from earth to moon (delay) is $385,000\text{km} / 3 \cdot 10^8\text{m/s} = 1.28333\text{s}$
b) $1.28333\text{s} \times 100 \text{ Mbps} = 128\,333\,000 \text{ bits} = 128 \text{ Mbits} = 16 \text{ MByte}$
c) The request will take 1.28333s to reach the moon. The time to put the picture on the link is $25\text{MB}/100\text{Mbps} = 2\text{s}$. It will take another 1.28333s to return to Earth. So the total time is $2 \times 1.28333\text{s} + 2\text{s} = 4.56666\text{s}$.

4 Bandwidth

How much bandwidth is needed to send computer screen images through an optical fiber? Assume the screen is 1920x1080 pixels, and each pixel carries 24 bits. The framerate is 60 images per second.

Answer: $1920 \times 1080 \times 24 \times 60 = 2,985,984,000$ or about 3Gbps.

5 Nyquist Theorem

We know that the Nyquist theory applies to copper wire. Does it also apply to optical fiber? Why? Why not?

Answer: The theorem also applies to optical fiber. It has nothing to do with the transmission medium.

6 Internet Checksum

A message 1001 1100 1010 0011 is transmitted using the Internet Checksum (using 4-bit words). What is the value of the checksum?

Answer: Adding the 4-bit words yields $1001 + 1100 + 1010 + 0011 = 100010$. The carry is added to the LSBs: 0100. The one's complement value is then 1011, which is the checksum.

And indeed, the check works out as expected. Adding the data and the checksum yields $1001 + 1100 + 1010 + 0011 + 1011 = 101101$, which gives 1111 (carry) and thus 0000 (negation).

7 Hamming Code

A 12-bit hamming code whose hexadecimal value is 0xE4F arrives at a receiver. What was the original value in hexadecimal? Assume that no more than 1 bit is in error.

Answer: The 12-bit hamming code in binary notation reads

1	1	1	0	0	1	0	0	1	1	1	1
b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}

The bits are split into parity bits and data bits as follows

1	1	1	0	0	1	0	0	1	1	1	1
p_1	p_2		p_3				p_4				
		d_1		d_2	d_3	d_4		d_5	d_6	d_7	d_8

We should have $p_1 = b_3 \oplus b_5 \oplus b_7 \oplus b_9 \oplus b_{11} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$, which is indeed the case.

Moreover, we should have $p_2 = b_3 \oplus b_6 \oplus b_7 \oplus b_{10} \oplus b_{11} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$. However, we see that $p_2 = 1$. Before we guess the error, let's go on first.

For p_3 , we should have $p_3 = b_5 \oplus b_6 \oplus b_7 \oplus b_{12} = 0 \oplus 1 \oplus 0 \oplus 1 = 0$, which is indeed the case.

We conclude that the value of the parity bit p_2 , hence bit b_2 has accidentally been flipped. After correction, we obtain

1	0	1	0	0	1	0	0	1	1	1	1
b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}

which corresponds to subtracting 0x400 from 0xE and yields thus 0xA4F.