CS433/533 Homework Assignment One
Bo Song

1.
Based on geometric distance between destination and Yale, I found a domain in south Africa that has a longest number of hops in the path, which is 24.
utl-lnx1.puk.ac.za 24 hops South Africa, National Research Foundation

6 ISP
Yale University
State of Connecticut
Ip Allocation for Geant Network
Ubuntunet Alliance for Research and Education Networking.
Uninet
Potchefstroom University for Christian Higher Education
Traceroute list:
1 anger.net.yale.edu (128.36.232.1) 3.636 ms 3.642 ms 3.641 ms
Yale University
2 10.1.2.81 (10.1.2.81) 0.390 ms 0.421 ms 0.464 ms
3 10.1.2.113 (10.1.2.113) 0.402 ms 0.449 ms 0.597 ms
4 Level3-10G-ASR.net.yale.internal (10.1.4.40) 1.759 ms 4.145 ms 4.143 ms
5 cen-10g-yale.net.yale.internal (10.1.3.102) 2.205 ms 1.790 ms 2.553 ms
6 * * *
7 enrt064hhh-9k-te0-3-0-5.net.cen.ct.gov (67.218.83.254) 2.950 ms 2.867 ms 2.985 ms
State of Connecticut
8 198.71.46.215 (198.71.46.215) 2.419 ms 2.500 ms 2.425 ms
Michigan, Ann Arbor, Internet
9 et-5-0-0.1180.rtr.newy32aoa.net.internet2.edu (198.71.46.214) 5.207 ms 5.164 ms 5.149 ms
10 198.71.45.237 (198.71.45.237) 76.614 ms 76.550 ms 76.467 ms
11 ae2.mx1.gen.ch.geant.net (62.40.98.153) 83.117 ms 83.106 ms 83.066 ms
Switzerland, geneve, ip allocation for geant Network
12 ae1.mx1.fra.de.geant.net (62.40.98.109) 91.528 ms 91.521 ms 91.479 ms
13 ubuntunet-gw.mx1.fra.de.geant.net (62.40.125.22) 98.086 ms 98.113 ms 97.904 ms
Netherlands, Amsterdam, Ip Allocation for geant network Infrastructure
14 196.32.210.177 (196.32.210.177) 111.021 ms 112.082 ms 112.039 ms
South Africa, Western Cape, Ubuntunet Alliance for research and education networking.
15 te-1-4-2018-mtz1-pe1.ubuntunet.net (196.32.209.117) 282.660 ms 282.674 ms 282.629 ms
16 xe0-0-2-700-dur1-pe2-n.tenet.ac.za (155.232.6.86) 285.863 ms 286.988 ms 287.012 ms
Western Cape, Uninet
17 te0-12-0-2-pta1-p1-n.tenet.ac.za (155.232.6.29) 330.471 ms 323.929 ms 322.393 ms

18 te0-1-0-1-jnb2-p1-n.tenet.ac.za (155.232.6.25) 336.656 ms 323.132 ms 332.443 ms
19 t8-2-jnb2-pe1-n.tenet.ac.za (155.232.7.158) 297.720 ms 297.971 ms 298.078 ms
20 155.232.14.149 (155.232.14.149) 300.850 ms 300.373 ms 300.975 ms
Limpopo, Giyani, Uninet
21 143.160.3.49 (143.160.3.49) 301.309 ms 301.366 ms 301.320 ms
Potchefstroom, Potchefstroom University for Christian Higher Education
22 143.160.3.74 (143.160.3.74) 301.666 ms 301.813 ms 301.767 ms
23 143.160.7.242 (143.160.7.242) 301.532 ms 309.918 ms 301.574 ms
24 utl-lnx1.puk.ac.za (143.160.32.1) 367.425 ms 369.060 ms 368.584 ms

1.
[P2] Determine the number of external phone lines that Yale will need in order to achieve a call blocking percentage of 1%. Assume that each person at Yale makes one external phone call per day, and each such phone call lasts on average 3 minutes, with the memoryless distribution. The number of people at Yale can be found at: http://www.yale.edu/about/facts.html

According to the webpage, currently Yale has 16722 students and faculties.

$$\mathrm{p_k} = \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k p_0$$

$$\mathrm{p_0} = \frac{1}{1 + \frac{1}{1!}\frac{\lambda}{\mu} + \frac{1}{2!}\left(\frac{\lambda}{\mu}\right)^2 + \cdots + \frac{1}{N!}\left(\frac{\lambda}{\mu}\right)^N}$$

Merge the two equations above:

$$\mathrm{p_k} = \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k \frac{1}{1 + \frac{1}{1!}\frac{\lambda}{\mu} + \frac{1}{2!}\left(\frac{\lambda}{\mu}\right)^2 + \cdots + \frac{1}{N!}\left(\frac{\lambda}{\mu}\right)^N}$$

$$\lambda = \frac{16722}{24 * 60}$$

$$\mu = \frac{1}{3}$$

$$\mathrm{N} = 16722$$

Let $\mathrm{p_N} = 1\%$
When N == 46, blocking percentage is 1.19%
When N == 47, blocking percentage is 0.8%
Python code

```
import math
n = 47
l = 16722.0/24/60
u = 1.0/3
sum = 0.0
for x in xrange(0,n+1):
    sum += (l / u) ** x / math.factorial(x)
res = (l / u) ** n / math.factorial(n) / sum
print res
```

[P3] Suppose that you are designing a Web server for your startup. You have acquired a single machine with a quad-core processor. Assume that CPU is the bottleneck. You anticipate that Web requests arrive (memoryless) at a rate of 15 requests/second, and benchmarking shows that it takes a core on average 200 ms to serve a Web request. What is the average service time that each Web request experiences? If it is a dual-core processor, what happens? You need to draw the state diagram when working on this problem.

Service time equals to:

$$\text{Service Time} = \mathrm{S}\frac{1}{1-\rho}$$

$$\rho = \frac{\lambda}{\mu}, \mathrm{S} = \frac{1}{\mu}$$

In quad-core processor, Service time = 1 / 5 * 1 / (1 – (15/4) / 5) = 4 / 5

In duo-core processor, S = 1 / 5 * 1 / (1 – (15/2) / 5) = Infinite big.

[P4] This elementary problem explore propagation delay and transmission delay, two central concepts in data networking. Consider two hosts, A and B, connected by a single link of rate R bps. Suppose that the two hosts are separated by m meters, and suppose the propagation speed along the link is s meters/sec. Host A is to send a packet of size L bits to Host B.

Express the propagation delay, dprop, in terms of m and s.

Determine the transmission time of the packet, dtrans, in terms of L and R.

Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.

Suppose Host A begins to transmit the packet at time t = 0. At time t = dtrans, where is the last bit of the packet?

Suppose dprop is greater than dtrans. At time t = dtrans, where is the first bit of the packet?

Suppose dprop is less than dtrans. At time t = dtrans, where is the first bit of the packet?

Suppose s = 2.5 x 108, L = 120 bits, and R = 56 kbps. Find the distance m so that dprop equals dtrans.

1. Dprop = m / s
2. Dtrans = L / R
3. End-to-end delay = Dprop + Dtrans = m / s + L / R
4. Last bit just leave the source.
5. First bit of the packet is L / R * s away from source
6. First bit of the packet is at the destination
7. M / (2.5 * 10^8) = 120 / (56 * 1000)
   M = 5.36*10^5

[P5] Suppose two hosts, A and B, are separated by 20,000 kilometers and are connected by a direct link of R = 2 Mbps. Suppose the propagation speed over the link is 2.5 · 108 meters/sec.

Calculate the bandwidth-delay product, R · dprop.

Consider sending a file of 800,000 bits from Host A to Host B. Suppose the file is sent continuously as one large message. What is the maximum number of bits that will be in the link at any given time?

Provide an interpretation of the bandwidth-delay product.

What is the width (in meters) of a bit in the link? Is it longer than a football field?

Derive a general expression for the width of a bit in terms of the propagation speed s, the transmission rate R, and the length of the link m.

R * Dprop = 2 * 10^6 * (20000 * 10 ^ 3 / (2.5 * 10^8)) = 1.6 * 10^5

$$\#\text{bit on link} = \min\left(\frac{20000 * 10^3}{2.5 * 10^8} * 2 * 10^6, 8 * 10^5\right) = 1.6 * 10^5$$

Interpretation:

The maximum number of bit that would be in the link.

Width of a bit:

$$\text{Width of a bit} = \frac{\text{length}}{\#\text{bit on link}} = \frac{2*10^7}{1.6*10^5} = 125\text{m}$$

Reference to this webpage hypertextbook.com/facts/2001/NinTam.shtml

The rectangular field is 160 feet (48.5 m) wide and 360 feet (109.1 m) in length (53.3 yards, 120 yards respectively), including the two 10 yard (9.2 m) end zones. If you know the rules of the game, then you should know how hard it is to gain just one yard.

So the width of a bit is longer than a football field.

$$\text{Width of a bit} = \frac{\text{s}}{\text{R}}$$

[P6] In this problem, we consider sending real-time voice from Host A to Host B over a packet-switched network (VoIP). Host A converts analog voice to a digital 64 kbps bit stream on the fly. Host A then groups the bits into 56-byte packets. There is one link between Hosts A and B; its transmission rate is 2 Mbps and its propagation delay is 10 msec. As soon as Host A gathers a packet, it sends it to Host B. As soon as Host B receives an entire packet, it converts the packet's bits to an analog signal. How much time elapses from the time a bit is created (from the original analog signal at Host A) until the bit is decoded (as part of the analog signal at Host B)?

$$\text{t} = \frac{56 * 8}{2 * 10^6} + 0.01 = 0.010224s$$

[P7] Suppose you would like to urgently deliver 40 terabytes data from Boston to Los Angeles. You have available a 100 Mbps dedicated link for data transfer. Would you prefer to transmit the data via this link or instead use FedEx over-night delivery? Explain.

Time of using link:

$$T_{\text{link}} = \frac{40 * 8 * 10^{12}}{100 * 10^{6}} = 3.2 * 10^{6}\text{s}$$

Time of using FedEx:

$$T_{\text{FedEx}} = 24 * 60 * 60 = 8.64 * 10^{4}s$$

$$T_{\text{FedEx}} < T_{\text{link}}$$

Use FedEx

[P8] Consider an application that transmits data at a steady rate (for example, the sender generates an N-bit unit of data every k time units, where k is small and fixed). Also, when such an application starts, it will continue running for a relatively long period of time. Answer the following questions, briefly justifying your answer:

Would a packet-switched network or a circuit-switched network be more appropriate for this application? Why?

Suppose that a packet-switched network is used and the only traffic in this network comes from such applications as described above. Furthermore, assume that the sum of the application data rates is less than the capacities of each and every link. Is some form of congestion control needed? Why?

1. Circuit-switched network is more appropriate. Firstly, the application transmits data at a steady rate. Secondly, the application runs in a relatively long time. Circuit-switched network fits its requirements better.
2. Yes. According to the description of the question, the sender may generate data at a momentary peak rate, which may overflow the buffer and exceed the capacity of link. So congestion control is needed.

[P9] Suppose users share a 3 Mbps link. Also suppose each user requires 150 kbps when transmitting, but each user transmits only 10 percent of the time. (You can refer to the discussion of packet switching versus circuit switching in Section 1.3.2. of the textbook, if you want.)

When circuit switching is used, how many users can be supported?

For the remainder of this problem, suppose packet switching is used. You can use either our queueing analysis in class or direct binomial distribution analysis. Find the probability that a given user is transmitting.

Suppose there are 120 users. Find the probability that at any given time, exactly n users are transmitting simultaneously.

Find the probability that there are 21 or more users transmitting simultaneously.

Circuit switching can support $\frac{3*10^{6}}{150*10^{3}} = 20 \text{ users simutinously.}$

For a given user, the probability that a given user is transmitting: 10%

$$P_{\text{n}} = C_{120}^{\text{n}} 0.1^{n} 0.9^{120-n}$$

$$P_{\text{n}}(n > 20) = 1 - P_{n}(n \leq 20) = 1 - \sum_{\text{n}=0}^{20} C_{120}^{n} 0.1^{n} 0.9^{120-n} = 0.008$$

Python code

```
import operator as op
def ncr(n, r):
        r = min(r, n-r)
        if r == 0: return 1
        numer = reduce(op.mul, xrange(n, n-r, -1))
        denom = reduce(op.mul, xrange(1, r+1))
        return numer//denom

sum = 0.0
for x in xrange(0,21):
        sum += ncr(120, x) * (0.1 ** x) * 0.9 ** (120 - x)
sum = 1 - sum
print sum
```

[P10] In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as message segmentation. The figure below illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is 8x106 bits long that is to be sent from source to destination in the figure below. Suppose each link in the figure is 2 Mbps. Ignore propagation, queuing, and processing delays.

Consider sending the message from source to destination without message segmentation. How long does it take to move the message from the source host to the first packet switch? Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?

Now suppose that the message is segmented into 800 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?

How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.

In addition to reducing delay, what are reasons to use message segmentation?

Discuss the drawbacks of message segmentation.

1. $\mathrm{T_{unit}} = \frac{8*10^6}{2*10^6} = 4s,\ \mathrm{T_{total}} = T_{unit} * 3 = 12s$

2. It takes $\mathrm{T_{unit}} = \frac{10^4}{2*10^6} = 0.005s$ to move the first packet from source host to the first switch.

   At T=0.005s, the first packet is sent from the first switch to the second switch and the second packet is sent from the source host to the first switch. At T=0.01s will the second packet be fully received at the first switch.

3. It takes $\mathrm{T_{total}} = \frac{8*10^6}{2*10^6} + T_{unit} * 2 = 4.01s$ to send the whole file if message segmentation is used. Sending package with segmentation is much more efficient than that without segmentation.
4. If package corrupts during transmission, transmit with message segment need only resend the small package rather than the whole big file, which increase the performance of transmission. In the path of transmission, there may be some devices that do not support large package transfer.
5. Message segmentation increases the overhead of transmission since each package should contain a header. It also increases the structure complexity of routers, since each router should segment/reassemble packages.

[P11] Consider sending a large file of F bits from Host A to Host B. There are three links (and two switches) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of S bits each and adds 80 bits of header to each segment, forming packets of L = 80 + S bits. Each link has a transmission rate of R bps. Find the value of S that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay.

$$\mathrm{T_{delay}} = \frac{80 + S}{R} \cdot (ceiling\left(\frac{F}{S}\right) + 2)$$

$$\mathrm{S} = \sqrt{40\mathrm{F}}$$

[P12] Both Skype and Google Talk offer services that allow you to make a phone call from a PC to an ordinary phone. This means that the voice call must pass through both the Internet and through a telephone network. Discuss how this might be done.

There must be a gateway in the path that can convert packet data to circuit data and vise versa. Part of the path uses circuit switch so a connection should be established over circuit switch before communication.