P1

From XMU, traceroute.

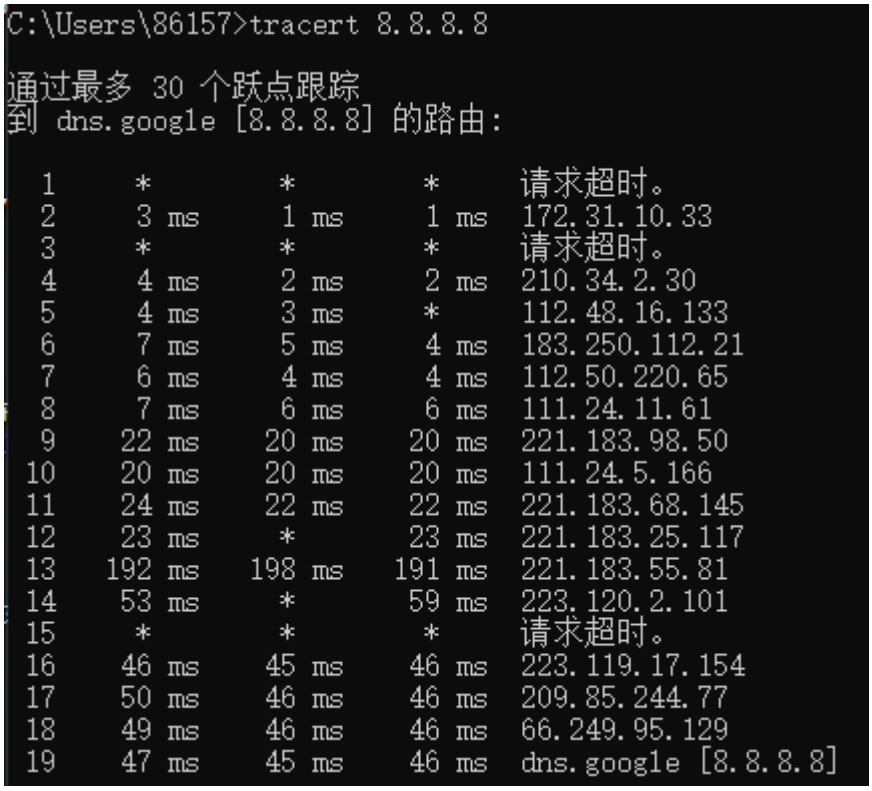
a.

In Ubuntu try to traceroute www.bilibili.com, but fail.

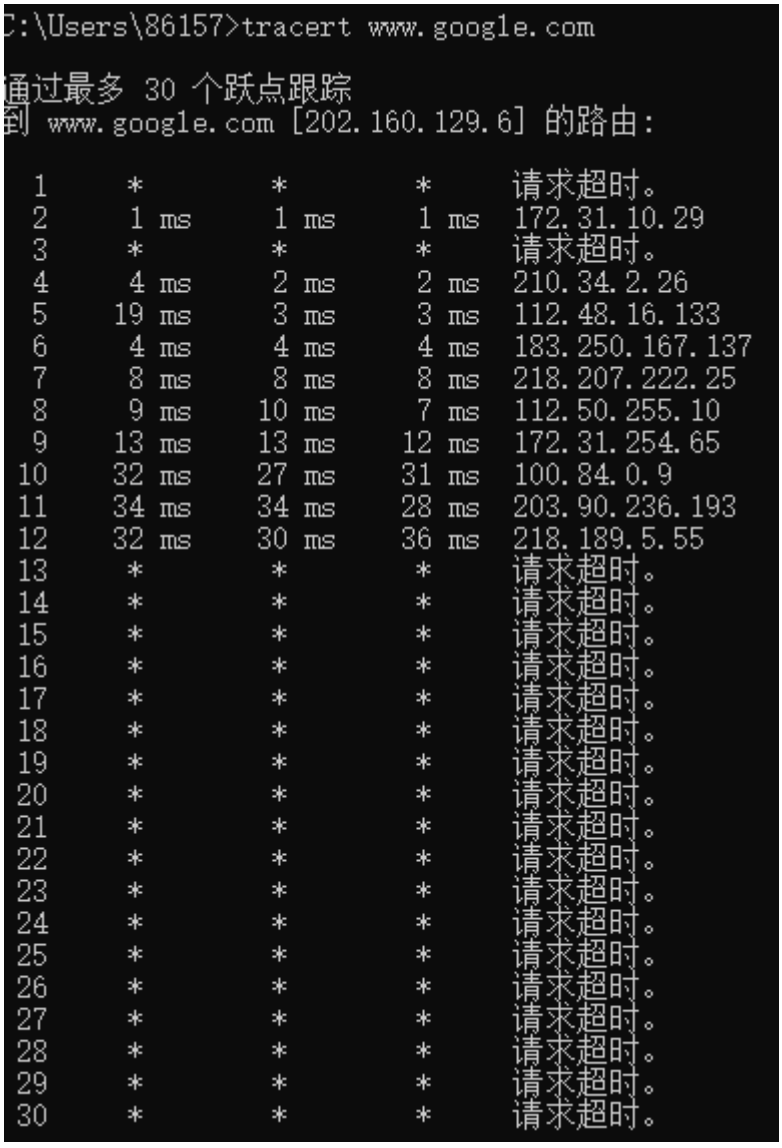
```
hadoop@ubuntu:~$ traceroute www.bilibili.com
traceroute to www.bilibili.com (120.240.49.148), 30 hops max, 60 byte packets
 1  _gateway (192.168.72.2)  0.334 ms  0.186 ms  0.129 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
```

In Windows, the use of tracert is the same as traceroute in Linux.

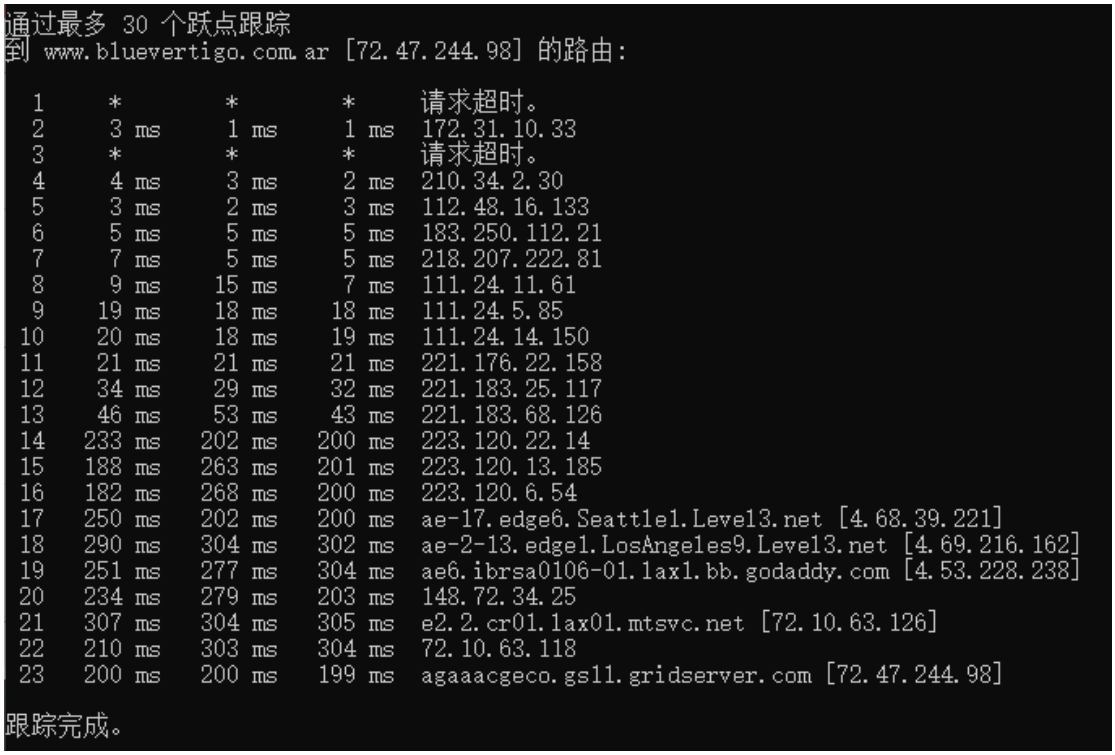
After trying tracert various IP, I found that foreign IP takes more hops than domestic IP commonly. Therefore, I found a large number of hops of a route is Google's public DNS 8.8.8.8, which takes 18 hops to get to the destination. The hops are as follows.



Also, when I tried to tracert www.google.com, I found it impossible to get the destination.

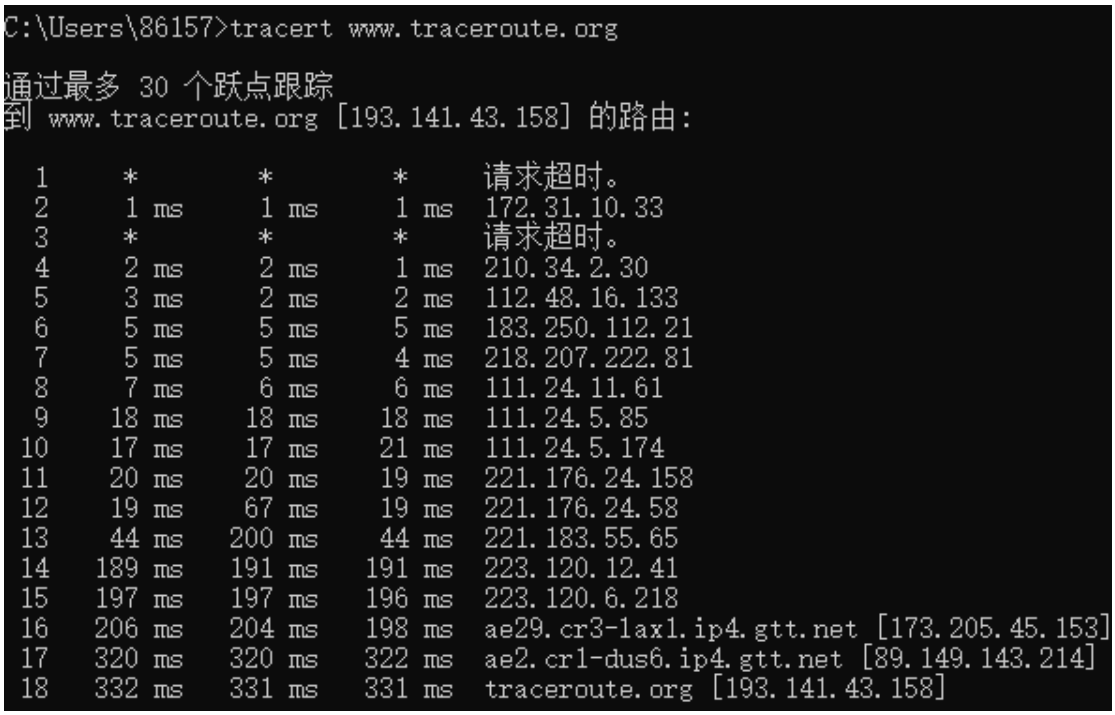


At last, I baidu that which country is the farthest from China. So I found a website of Argentina, which takes 22 hops to get to the destination. The hops are as follows.



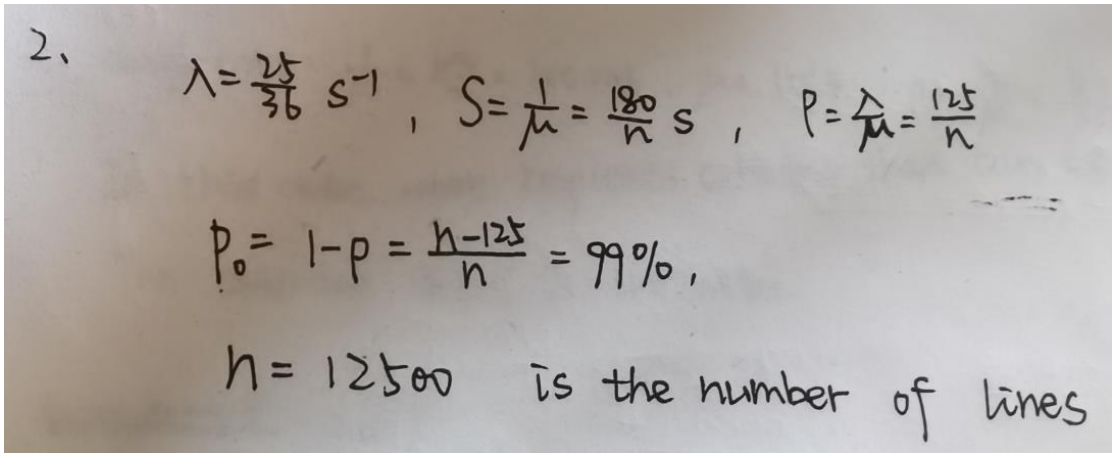
b.

Similarly, I tried foreign IP and finally got the largest of ISPs. But I can’t find a good way to show the ISP concisely. The ISPs are as follows:



福建省厦门市教育网、中国移动、美国 GTT、爱尔兰、德国。

P2


2.
$\lambda = \frac{25}{36}\ s^{-1}$, $S = \frac{1}{\mu} = \frac{180}{n}\ s$, $\rho = \frac{\lambda}{\mu} = \frac{125}{n}$
$P_0 = 1 - \rho = \frac{n-125}{n} = 99\%$,
$n = 12500$ is the number of lines

P3

3. $\lambda = 15/s$, $\frac{1}{\mu} = \frac{200}{4} = 50ms$, $\mu = 20/s$

$\rho = \frac{\lambda}{\mu} = \frac{3}{4}$, $S = 50ms$,

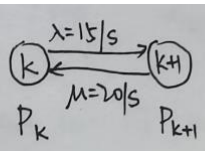
the average service time is $d = S \cdot \frac{1}{1-\rho} = 150m/s$

dual-core: $\frac{1}{\mu} = \frac{200}{2} = 100ms$, $\mu = 10/s$, $\rho = \frac{\lambda}{\mu} = \frac{3}{2} > 1$

In this case, more requests arriving than can be serviced

The average delay is infinite.

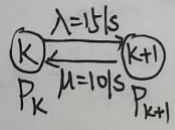
quad-core:



at equilibrium: $P_k \cdot \lambda = P_{k+1} \cdot \mu$, $\rho = \frac{\lambda}{\mu} = \frac{3}{4}$

~~when~~ $0 \le k \le n-1 \Rightarrow P_k < P_{k+1}$ and delay is small

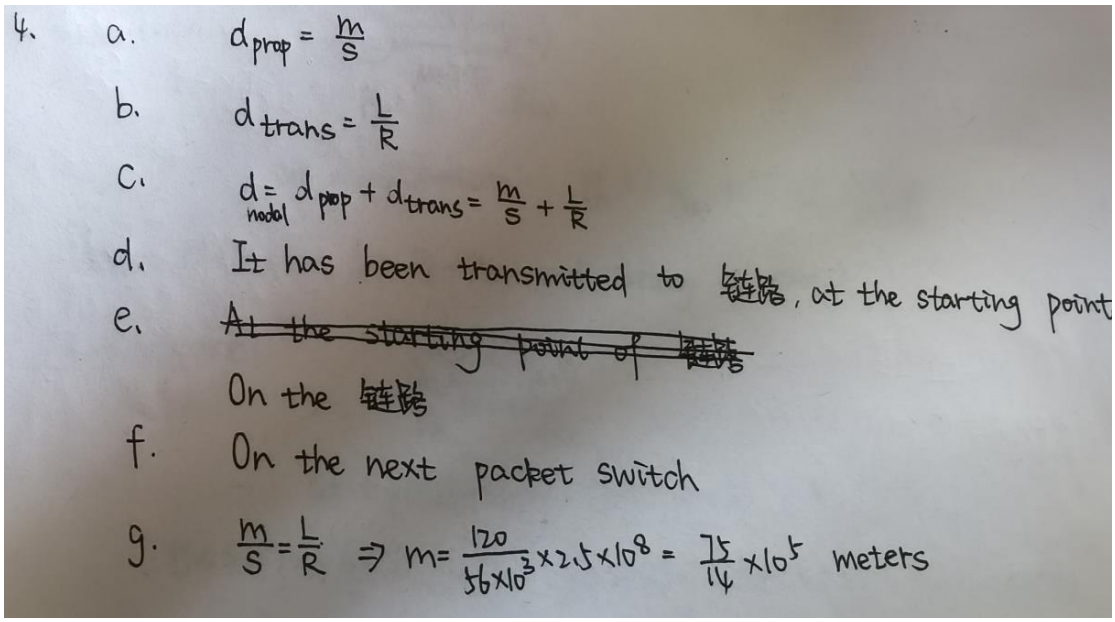
dual-core:



at equilibrium: $P_k \lambda = P_{k+1} \mu$, $\rho = \frac{\lambda}{\mu} = \frac{3}{2}$

$0 \le k \le n-1 \Rightarrow P_{k+1} > P_k$, queueing ~~ten~~ delay tends to be infinite

P4


4.
a. $d_{prop} = \frac{m}{s}$
b. $d_{trans} = \frac{L}{R}$
c. $d_{nodal} = d_{prop} + d_{trans} = \frac{m}{s} + \frac{L}{R}$
d. It has been transmitted to 链路, at the starting point
e. ~~At the starting point of 链路~~
On the 链路
f. On the next packet switch
g. $\frac{m}{s} = \frac{L}{R} \Rightarrow m = \frac{120}{56\times10^3} \times 2.5\times10^8 = \frac{75}{14}\times10^5$ meters

P5-P7

5. a. $R \cdot d_{prop} = 2\times10^6 \times \frac{2\times10^7}{2.5\times10^8} = 1.6\times10^5$ bits

b. $d_{prop} = 8\times10^{-2}$ s. $d_{trans} = \frac{8\times10^5}{2\times10^6} = 4\times10^{-1}$ s

$d_{trans} > d_{prop}$, Therefore the maximum number of bits is $R \cdot d_{prop} = 1.6\times10^5$ bits

c. 链路上可容纳的最大比特数

d. $\frac{2\times10^7}{1.6\times10^5} = 125$ m. Yes, ~~maybe (I don't actually know)~~

e. $\frac{m}{R \cdot \frac{m}{s}} = \frac{s}{R}$

6. $d_{proc} = \frac{56\times8}{64\times10^3}$ s $= 7$ ms $\quad d_{trans} = \frac{56\times8}{2\times10^6}$ s $= 2.24\times10^{-1}$ ms

$d_{modal} = 7 + 0.224 + 10 = 17.224$ ms

7. I prefer to use FedEx.

Dedicated link: ~~$d_{trans} = \frac{4\times10^{13}}{1\times10^8} = 4\times8\times10^5$ s~~

$d_{trans} = \frac{4\times10^{13}\times8}{1\times10^8}$ s $= 3.2\times10^6$ s ≈ 37 days

Though it is dedicated, it takes so long time to transfer so large data. Nobody can assure the electricity supply.

So FedEx is better.

P8-P9

8. a. Circuit-switched network.

Reasons: The transmitting rate is steady, and the application will continue running for a long period of time. Reservation and a single partition bandwidth are suitable for steady and a long period. Also, when too many clients use the application simultaneously, the rate may fluctuate.

b. No. Because the sum of rates is still less than the capacities of every link. The link is large enough so the queue wouldn't be longer than the maximum, so there is no congestion to control.

9. (a) the number of users is $\frac{3\times10^6}{1.5\times10^5} = 20$

(b) 10%

(c) $C_{120}^{n}\cdot(0.1)^{n}\cdot(0.9)^{120-n}$

(d) $\sum_{n=21}^{120} C_{120}^{n}(0.1)^{n}(0.9)^{120-n}$

P10

10. a. $d_{trans} = \frac{L}{R} = \frac{8\times10^6}{2\times10^6} s = 4s$

from source host to destination host : $3d_{trans} = 12s$

b. $\Delta t = \frac{L}{R} = \frac{1\times10^4}{2\times10^6} = 5\times10^{-3} s = 5ms$

At $2\Delta t = 10ms$

c. Time of the first packet arrives destination is $3\Delta t = 15ms$

The 800 th packet starts time: $0+(800-1)\times\Delta t = 3995ms$

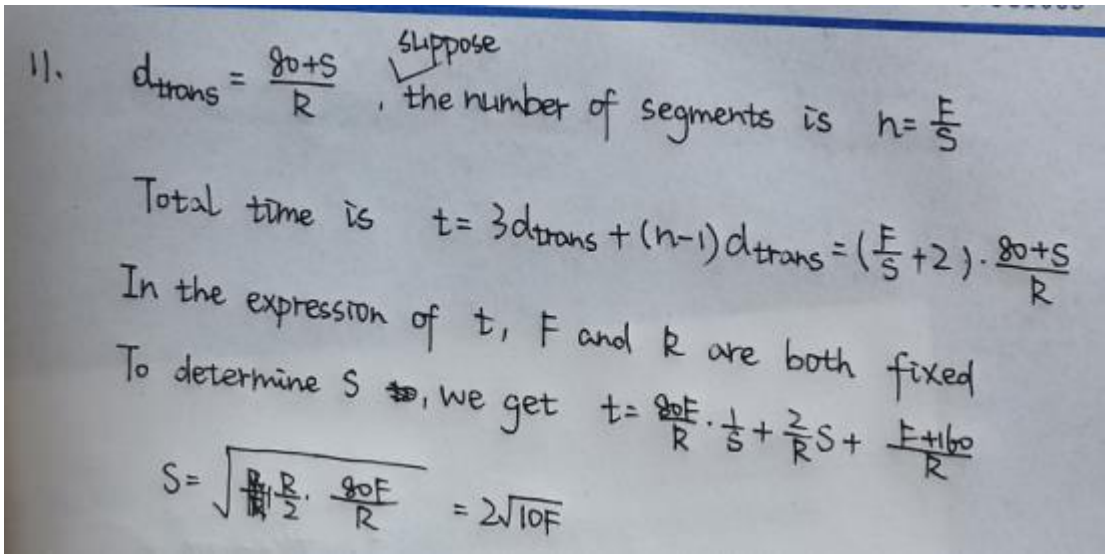
Total time: 4010 ms

By taking message segmentation, it gets much faster than (a). Message segmentation makes it efficient to transmit data in packet-switched packets.

d. If multiple and huge messages are transmitted without message segmentation simultaneously, they will queue up and the time of which will be incredible.

e. The link and packet switch have to tolerate the additional cost of per packet header.

P11



P12

In my opinion, a PC transmits data by using packet switching network while an ordinary phone use circuit switching network. Therefore, after Skype receives voice data from us through packet switching network, this application rearranges packets and transforms them into a certain data form, which is suitable for circuit switching network so that our original voice can be sent to our target receiver through circuit switching network.