

Lab2: Measuring Network Performance - Solutions

COM-208: Computer Networks

The goal of this lab is to understand **the meaning of throughput and delay**, as well as **the nature of different delay components**. For this, you will use `ping` (which you discovered in the last lab) to **experimentally estimate** throughput and delay between your computer and various targets around the world. You will see that there are quantities that we cannot directly measure, yet we can reason about them if we understand what they are.

Setup

Download from Moodle `runping.sh` and `plot.sh`. If they are not already executable, make them, e.g., by running `chmod u+x runping.sh` and `chmod u+x plot.sh`. If you are doing the lab on your own computer, make sure you have the `gnuplot` application installed.

`runping.sh` is a shell script that takes as an argument a target (DNS name or IP address) and pings the target many times with different packet sizes. For example, `./runping.sh www.epfl.ch` executes

```
ping -s 22 -c 50 -i 1 www.epfl.ch > www.epfl.ch-p50
...
ping -s 1472 -c 50 -i 1 www.epfl.ch > www.epfl.ch-p1500
```

The first line pings `www.epfl.ch` 50 times, once per second, using 50-byte packets, and writes the output to a file called `www.epfl.ch-p50`. The last line pings `www.epfl.ch` 50 times, once per second, using 1500-byte packets, and writes the output to a file called `www.epfl.ch-p1500`.

`plot.sh` takes as an argument a list of file names (that were produced by `runping.sh`) and produces three new files: (1) `destination_delay.png` shows delay as a function of time, for different packet sizes (different colors correspond to different packet sizes).

(2) `destination_scatter.png` shows delay as a function of packet size as a scatter plot.
(3) `destination_avg.txt` contains the average (2nd column) and minimum (3rd column) delay values for each packet size (1st column). For example, `./plot.sh www.epfl.ch-p*` produces the above three files for the target `www.epfl.ch`.

Measure and process

Run `runping.sh` with target `www.epfl.ch`, then with target `www.yahoo.com`, and then with target `www.gov.kg`. Given that you are sending one ping every second, these will take several seconds to complete.

When they finish, process the results by running:

```
./plot.sh www.epfl.ch-p*
./plot.sh www.yahoo.com-p*
./plot.sh www.gov.kg-p*
```

Understand your tool

We said earlier that `ping -s 22...` pings the target with 50-byte packets, while `ping -s 1472...` pings the target with 1500-byte packets.

- If the command specifies `-s 22` (or `-s 1472`), why might the resulting pings use 50-byte (or 1500-byte) packets, and not 22-byte (or 1472-byte) packets? To answer, start by reading what the `-s` parameter exactly means in the `ping` manual.

The 28 extra bytes come from headers: each ping message carries an ICMP header of 8 bytes and is encapsulated in an IP header of 20 bytes.

- What exactly does it mean that the target is pinged with “50-byte packets” or with “1500-byte packets”? Which part of the ping packet is 50 bytes? The application-layer message (header + data)? The entire packet (with transport-layer and network-layer and link-layer headers)? You can use Wireshark to answer, though that’s not the only way.

The application-layer message plus the transport-layer header plus the network-layer header.

Think about delay

A packet experiences different kinds of delay:

- The **transmission delay** experienced by a **packet on a link** is the amount of time it takes to push the bits of the packet onto the link, and it is equal to the packet length divided by the link transmission rate.
- The **propagation delay of a link** is the amount of time it takes for one bit to go from one end of the link to the other, and it is equal to the link length divided by the link propagation speed.
- The **queuing delay** experienced by a **packet at a switch** is the amount of time the packet sits inside a queue at the switch, waiting for other packets to be processed and transmitted. Queuing delay depends on the **other traffic** that traverses the same links and shares the same queues as the packet.
- The **processing delay** experienced by a **packet at a switch** is the amount of time it takes for the switch to process the packet (after it removes it from the queue and before starting to transmit it). Processing delay depends on the switch's processing capabilities and is typically independent of packet size, at least in most of the scenarios we will discuss in this class.

Look at the figures you have produced for each target and answer the following questions:

- Does delay change significantly over time? Whether you answer yes or no, explain why you think this is happening. E.g., if you answered that it does not change, consider each delay component and explain why it doesn't change over time. If you answered that it does change, which delay component(s) are the ones that change?

The following plots (`destination_delay.png`) show delay over time, for different packet sizes (each curve corresponds to a different size):

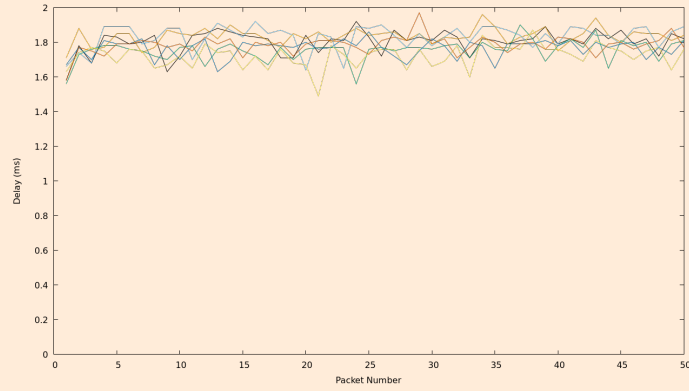


Figure 1: www.epfl.ch

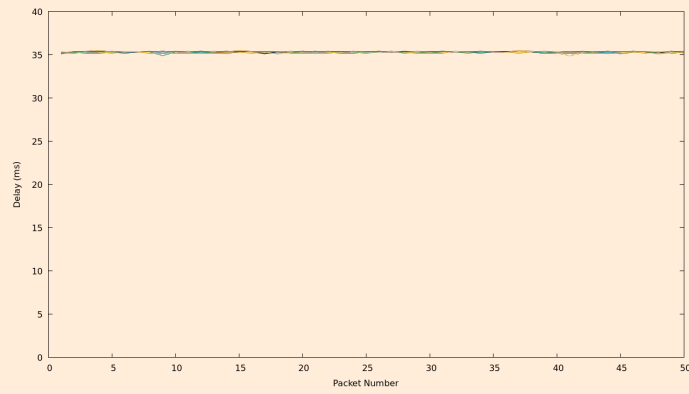


Figure 2: www.yahoo.com

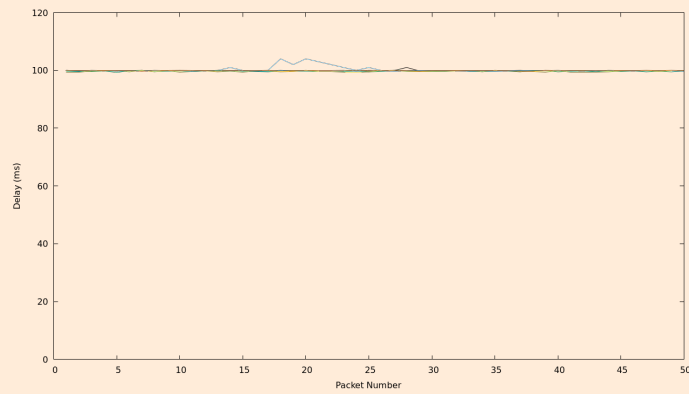


Figure 3: www.gov.kg

Delay does change over time, by hundreds of microseconds. This is most visible in the `www.epfl.ch` graph, because the total delay to this particular target is relatively small (less than 2ms).

When the path between two end-systems remains the same (i.e., consists of the same network links and packet switches), the only delay component that changes significantly over time is the queuing delay, which is affected by cross traffic. So, the change of delay over time that we observe in the graphs could be due to changing queuing delays.

Another possibility is that the path between your computer and each ping target changes over time, in which case it is not only the queuing delay, but the other delay components as well that change. Network-path changes, however, do not typically happen frequently enough to affect an experiment that lasts a few minutes. One exception is load-balancing, where there exist multiple network paths between two end-systems, and each packet is randomly assigned to one of them. However, if this happened between your computer and a ping target, you would see it in the delay pattern – packet delays would be clustered around a few distinct values, one for each network path.

Yet another possibility is that it is not (only) the network conditions or paths that change, but the processing delay *at the end-systems*: when a ping message arrives at a target, the target takes some time to process it and generate a response; similarly, when a ping response arrives at your computer, your computer takes some time to process it and register the time of its arrival. This processing delay may change over time depending on the other processes running on the target and on your computer.

- Does delay change significantly with the size of the ping packets? Do bigger packets experience more or less delay?
- Is the delay-packet size dependence the same for all targets? Whether you answer yes or no, explain why you think this is happening.

The following plots (`destination_scatter.png`) show delay as a function of packet size:

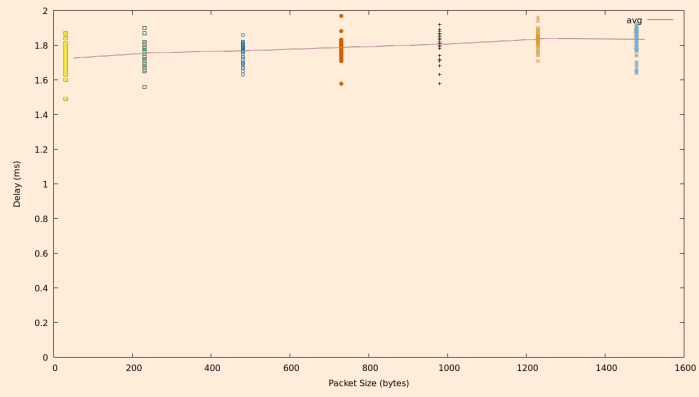


Figure 4: www.epfl.ch

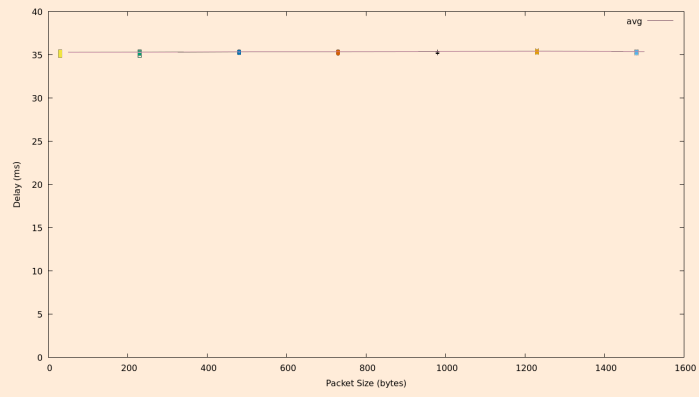


Figure 5: www.yahoo.com

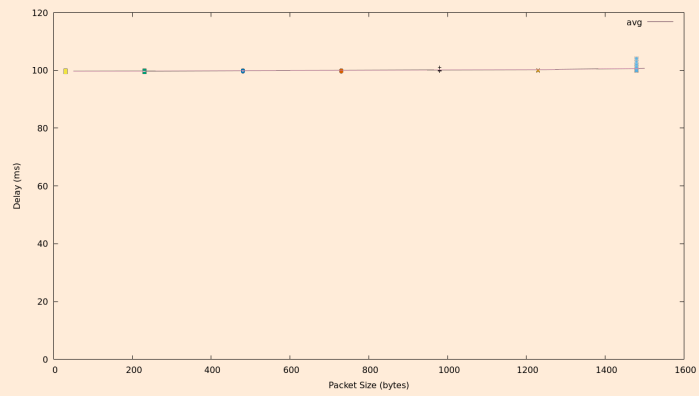


Figure 6: www.gov.kg

In the first plot, delay clearly increases with packet size. This is due to the transmission delay component, which increases with packet size.

In the other two plots, this trend is not clearly visible. The reason is that the propagation delay (which is independent of packet size) dominates the transmission delay, because the corresponding targets are much further away than the first target.

The above explains why the trend is not *clearly visible* (i.e., why delay does not increase with packet size *significantly*), but it does not explain why, in some cases, the delay experienced by a larger packet may actually be *smaller* than the delay experienced by a smaller packet. The most likely explanation for such an event is that the smaller packet encountered higher queuing delays inside the network, or higher processing delays at the end-systems, and the difference was large enough to counteract its smaller transmission delays.

- Suppose that, for a given target, there is no clear dependence pattern between delay and packet size, i.e., packet size does not affect delay significantly. What does that say about the path from your computer to that target?

That transmission delay on that path is dominated by propagation delay and/or queuing delay.

Estimate delay components

Now we will explore the kind of “rough” estimations that network engineers often need to do.

Ping tells you the overall delay experienced by different packets of different sizes, but it does not (directly) tell you anything about individual delay components, like transmission delay or queuing delay. This is not surprising, because ping operates at the end-systems, it has no access inside the network to see, e.g., how long it takes to transmit each packet on each link, or how long each packet has to wait in a queue at each switch.

Still, if we know the overall delay experienced by *many* different packets and of *many* different sizes, then we *can* infer information about individual delay components.

Study the content of the file `www.yahoo.com_avg.txt`. Estimate the following components of the delay between your computer and `www.yahoo.com`:

- The propagation+processing delay.
- The transmission delay.
- The average queuing delay.

You don't have enough information to do any precise estimation. Do the best you can given the information you have. Think about the particular nature of each delay component: Should it be the same for all packets, independently of their sizes and when each packet was sent? Should it depend on any particular way on packet size?

```
$ cat www.yahoo.com_avg.txt
```

```
50 35.279 34.940
250 35.293 34.991
500 35.358 35.224
750 35.361 35.193
1000 35.382 35.190
1250 35.404 35.221
1500 35.369 35.179
```

Propagation+processing delay: A reasonable approximation given the information we have is the **minimum delay** encountered by any packet sent to the given target, i.e., 34.940ms, in this example. Assuming that the path from your computer to the target and back does not change over time, and that propagation and processing delays do not change over time, we can be certain that the true propagation+processing delay is not larger than our approximation. It is a bit smaller, because even the smallest packet experiences non-zero transmission delays, and it may have experienced non-zero queuing delays; however, by picking the minimum delay encountered by any packet, we minimize this error.

Transmission delay: Consider **two packet sizes**, e.g., 50bytes and 1500bytes. First, we identify the minimum delay experienced by each packet size, i.e., 34.940ms and 35.179ms, respectively, in this example. Assume, for a moment, that the 50byte packet and the 1500byte packet that experienced these particular delays experienced the *same* propagation, processing, and queuing delays; so, if we subtract 34.940ms from 35.179ms, **all delay components cancel out except for transmission delay**, and we are essentially left with the transmission delay that would have been encountered by a 1450byte packet. So, we can approximate the transmission delay of a 1450byte packet by $35.179 - 34.940 = 0.239$ ms. How good is this approximation? The only unreasonable assumption we made is that the two packets experienced the same queuing delays. This may

not be true, but by picking the minimum delay experienced by each packet size, we minimize the error due to queuing delay.

Average queuing delay: A reasonable approximation given the information we have is to compute the **difference between the average and minimum delay** for each packet size, then compute the **average** of this difference **over all packet sizes**. In this example: $1/7 * ((35.279-34.940) + (35.293-34.991) + \dots + (35.369-35.179)) = 0.215\text{ms}$. By subtracting the minimum from the average delay for each packet size, we cancel out propagation, processing, and transmission delays. By averaging over all packet sizes, we get as close as possible to the true average queuing delay.

Now study the content of the file `www.gov.kg_avg.txt` and estimate the same delay components between your computer and `www.gov.kg`. Observe the differences in the results relative to `www.yahoo.com`. What might explain these differences?

```
$ cat www.gov.kg_avg.txt
```

```
50 99.699 99.372
250 99.761 99.354
500 99.920 99.601
750 100.003 99.676
1000 100.163 99.826
1250 100.274 99.927
1500 100.730 100.040
```

Using the same approximations:

- Propagation+processing delay: 99.354ms
- Transmission delay: $100.040 - 99.372 = 0.668\text{ms}$
- Average queuing delay: $1/7 * ((99.699-99.372) + (99.761-99.354) + \dots + (100.730-100.040)) = 0.393\text{ms}$

The most notable difference is in the propagation+processing delay. This is not surprising given that this website belongs to the government of Kyrgyzstan, so we expect it to be significantly further away from us than the Yahoo server that responded to our ping.

Estimate bottleneck throughput

The **bottleneck** link between a source and a destination is the link where data flows at the slowest rate. Estimate the **throughput** achieved by the bottleneck link between your computer and `www.yahoo.com`, *while transmitting a ping message*. Do the same estimation for target `www.gov.kg`.

Be careful: we are not asking for the *average* throughput achieved by the ping application (which is the total amount of ping-message data transferred from source to destination divided by the total transfer time). Rather, we are asking for the *instantaneous* throughput achieved *at the bottleneck link* while transmitting a ping message. To answer, you need to consider one particular delay component that you estimated above.

A link may act as a bottleneck link due to its transmission rate or due to queuing delays at the corresponding packet switch. Unfortunately, our measurements do not enable us to distinguish between these two scenarios.

The best we can do given the information we have, is assume that the bottleneck link is the one with the slowest transmission rate, and that that transmission rate is significantly lower than the transmission rate of any other link on the path – hence the transmission delay experienced by each packet on that link dominates the overall transmission delay experienced by the packet.

Under these assumptions, we can estimate the throughput of the bottleneck link as a packet size divided by our transmission-delay estimate for that packet size.

For `www.yahoo.com`:

- Packet size = $(1500 - 50) = 1450\text{bytes} = 11600\text{bits}$
- Transmission-delay estimate = 0.239ms
- Link throughput = $11600\text{bits} / 0.000239\text{s} = 48.54\text{mbps}$.

For `www.gov.kg`: $11600\text{bits} / 0.000668\text{s} = 17.37\text{mbps}$.