Homework 1: Getting Connected

Due: Thursday, October 3 @ 11:59 pm EDT

Overview and instructions

This homework has 4 problems. Most problems involve writing a short response, or filling out some tables with your answers. You can write your responses in your own document or add annotations to this one.

Note on collaboration

You are welcome (and encouraged!) to collaborate with your peers, but the solutions you write down must be **your own work** (ie, written by you). You are responsible for independently understanding all work that you submit—after discussing a problem as a group, you should ensure that you are able to produce your own answers independently to ensure that you understand the problem. For more information, please see the course Collaboration Policy.

In your submission, we ask that you include a brief *collaboration statement* describing how you collaborated with others on each problem—see the next section for details.

How to submit

You will submit your work in PDF form on Gradesope. Your PDF should conform to the following requirements:

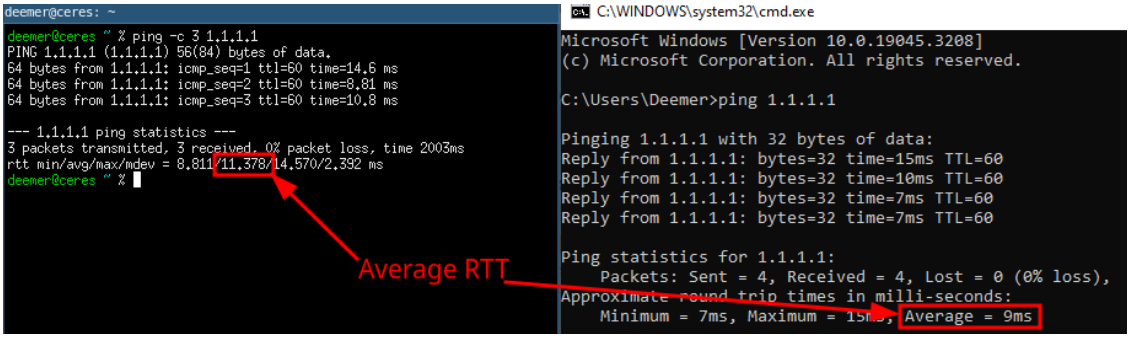
- Please **do not** include any identifying information (name, CS username, Banner ID, etc.) in your PDF, since all homeworks are graded anonymously
- Each problem (where "problem" is one of the Problems 1–4) should start on a separate page. When you submit on Gradescope, you will be asked to mark which pages correspond to which problem
- At the start of each problem, write a brief *collaboration statement* that lists the names and CS usernames of anyone you collaborated with and what ideas you discussed together
- If you consulted any outside resources while answering any question, you should cite them with your answer

1 Measuring latency

The Internet is pretty fast—but how fast is it, really? In this problem, we'll measure network latency using the `ping` command to gain a sense of how much latency the network adds to our communications and compare it to the theoretical lower bound: the speed of light. `ping` is a fundamental tool that is generally part of every host and router that uses IP: it sends a simple packet to a target host to see if it will respond back, and then measures the round-trip time (RTT). From any terminal on your host machine, you can ping another host like this:

- MacOS/Linux/Course container: `ping -c 3 <IP address>`
- Windows (Command prompt or Powershell): `ping <IP address>`

This particular invocation will send 3 ping packets, wait briefly for a response, and then show the round-trip time for each packet, and the average. On each system, it should look like this:



```
deemer@ceres: ~
deemer@ceres ~ % ping -c 3 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=60 time=14.6 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=60 time=8.81 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=60 time=10.8 ms

--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 8.811/11.378/14.570/2.392 ms
deemer@ceres ~ %
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Deemer>ping 1.1.1.1

Pinging 1.1.1.1 with 32 bytes of data:
Reply from 1.1.1.1: bytes=32 time=15ms TTL=60
Reply from 1.1.1.1: bytes=32 time=10ms TTL=60
Reply from 1.1.1.1: bytes=32 time=7ms TTL=60
Reply from 1.1.1.1: bytes=32 time=7ms TTL=60

Ping statistics for 1.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 15ms, Average = 9ms
```

Figure 1: Ping command on Linux/MacOS (left) and Windows (right)

Part a

The table on the next page lists four IP addresses and their approximate locations around the world. For each IP, fill in the table (or a similar table) by doing the following:

i. Find your distance to the location listed and fill it in on the table. An easy way to do this is just to search, eg, "distance from Providence, RI [your location] to Santa Barbara, CA [target IP]" on Google. An approximate value is fine—there is no need to be precise.

ii. Using the approximate distance you found, compute the **minimum round-trip time** based on the speed of light (≈3×10^8 m/s), and fill it in on the table. Write your answer in milliseconds.
Remember: round-trip time is the time to reach the destination *and come back*.

iii. Measure your average RTT to the target IP using `ping` and fill it in as the "measured RTT" on the table.

iv. Fill in the difference between your minimum and measured RTT, and write it down in milliseconds

Note on units: Write down your round-trip times and the difference in milliseconds, which is the standard way to report latencies for Internet traffic. You can report distance in either miles or kilometers—just make sure that you convert it appropriately when computing the minimum RTT (we've given you the speed of light in **meters** per second).

IP Address	Distance from you	Minimum RTT	Measured RTT	Difference
128.148.32.12 (Providence, RI)[1]				
128.111.1.1 (Santa Barbara, CA)				
129.67.1.190 (Oxford, UK)				
133.11.0.1 (Tokyo, Japan)				

[1] If you live in Providence, just measure the distance from where you are to the CIT. If you're in the CIT right now, just pick some small, nonzero value.

Answers will vary, but should show increasing RTTs with distance. (2 pts each: +1 for computed RTT, +1 for measured RTT)

(Optional) After looking at your measurements, what do you notice? Any thoughts or comments?

Part b

(4 pts) Your friend shows you their same measurements for this problem and every single one is at least 300ms higher than yours. **Why might this happen?** In 1–2 sentences, speculate on at least one possible difference that might cause this behavior. (We're not looking for one specific answer, just your reasoning about what might affect latency like this.)

Answer should mention some kind of *bottleneck*, probably close to the end host (ie, "your friend") that adds latency in all cases. Examples could include (but are not limited to):

- A really bad wifi link that adds a lot of latency
- Misbehaving ISP or other in-network device that is slowing down traffic
- Extremely high CPU or network load on the host that prevents it from receiving packets

Grading rubric: +2 pts for suggesting reasonable cause; +2 points for reasonably-correct explanation of how it would affect latency (can be brief)

2 Thinking about links

Consider the following questions about bandwidth and links based on what we learned in Lectures 2–4. These problems are open-ended—there are multiple possible correct answers. When grading, we will be looking for you to use sound reasoning to justify your answers based on your understanding of the principles we learned in class.

a. Say you download a big file from a website at 10MB/s. Next, you start two downloads at the same time and they each download at 5MB/s. **Why does this happen?** Explain in 1–2 sentences.

   Your connection has a fixed capacity, which limits the maximum possible throughput for downloading the file (in this case, 10MB/s). When starting two downloads, this capacity is shared among both connections (better yet, it's shared *fairly*, which is pretty nice).

   +2 for noting that the link has some kind of fixed capacity, +2 for explaining that the capacity is shared, partial credit as applicable.

b. Wifi has evolved significantly over time—modern versions like IEEE802.11ax advertise download speeds over 2Gbps. This surpasses the speeds for consumer-grade (non-datacenter) Ethernet cables and hardware, which typically operate at 1Gbps. **What are two reasons someone might still consider using using Ethernet over Wifi?** Explain your reasoning in around 1–2 sentences.

   There are several potential reasons (all of which are valid):

   - Ethernet is much less susceptable to packet loss than Wifi, so it should be able to reach its intended throughput much more consistently
   - Wifi's throughput is dependent on range—Ethernet may offer better performance at longer distances
   - (Other reasonable answers may also be accepted)

   +2 for each reason. For full credit, must compare Wifi/Ethernet in some way (ie, just writing "packet loss" with no explanation or connection is not sufficient). Partial credit as applicable.

3 IP forwarding practice

Note: We will have covered all the material for this problem after Lecture 7 on Thursday, September 28.

Suppose some router R has the following entries in its forwarding table:

Destination Network	Out Port/Next Hop
128.8.0.0/16	IF0
160.0.138.0/24	IF1
194.41.28.0/22	IF2
0.0.0.0/0	160.0.138.1

The table below lists some IP header fields for 4 packets labeled P1–P4. Based on this header information, fill in the last column ("Destination") to indicate where router R forward the packet—your answer should be one of the router's interfaces (IF0, IF1, IF2), or "drop", if the router would drop the packet.

When you submit your answer, feel free to show your work or write a brief justification for your decision.

#	Source IP	Dest. IP	TTL	Destination
P1	1.0.4.20	128.8.30.10	64	**IF0 (match on row 1)**
P2	128.8.12.1	160.0.138.56	0	**Drop (TTL is zero)**
P3	128.8.168.5	68.198.201.90	32	**IF1 (matches default route)**
P4	1.0.4.1	194.30.2.120	32	**IF1 (matches default route)**

3pts each, graded as follows:

- 3/3: Correctly identifies interface
- 2/3: Good reasoning for match, but missing important detail (TTL, longest prefix match, default)
- 1/3: Many details missing
- 0/3: Missing/no credit

4 Tutorial: sniffing broadcast traffic with Wireshark

This problem is a short "tutorial" on sniffing broadcast traffic, which is "background" traffic used to by certain services to discover information about the network. We'll discuss some of these services more in lecture 8 on Tuesday, October 3.

To complete the tutorial, follow the instructions here:
`https://hackmd.io/@csci1680/tutorial-broadcast-sniffing`

At the end of the tutorial, you'll be asked to write down some observations and include a screenshot of the traffic you found. To receive credit for this problem, just include these items in your submission.

Note: Tutorials are meant to be short problems where you gain some hands-on practice with some networking tools or protocols that we don't otherwise have a chance to cover in our assignments. These are mostly for your benefit, are graded on completion, and should not take more than 30 minutes to complete.

If you find yourself getting stuck on part of a tutorial, please come to hours or post on Ed to let us know. We can help resolve issues and clarify instructions. Tutorials are a new part of the course, and we hope they are fun and interesting, without adding too much work!