Problem Set #1

CS 352 Internet Technology

Released: September 16, 2022; Due: September 30, 2022 at 8 pm Eastern Time

Instructions

Please read and follow these instructions carefully.

1. You must work on this problem set **individually.**
2. Please complete your answers to the questions below and upload your responses to Canvas as **a single PDF file.**
3. Your answers (preferably typed up rather than handwritten) must be clear, legible, and concise. If we cannot understand your answer with reasonable effort, you will get zero credit.
4. If you leave a question out or clearly state "I don't know", you will receive 25% of the points for that question.
5. We care not just about your final answer, but also how you approach the questions. In general, if you show us your reasoning for your answers, we will do our best to provide partial credit even if your final answer is incorrect. However, if you provide no reasoning, and your answer turns out to be incorrect, you will typically receive zero points. So, please explain yourself.
6. You are free to discuss the problem set on Piazza or through other means with your peers and the instructors. You may refer to the course materials, textbook, and resources on the Internet for a deeper understanding of topics. However, you cannot lift solutions from other students or from the web. Do not post these problems to question-answering services like Chegg. All written solutions must be your own. We run sophisticated software to detect plagiarism and carefully monitor student answers.
7. There is a due date/time but no "time limit" on problem sets. That is, you may take as long as you need to work on problem sets, as long as you submit them on time. Further, you may make an unlimited number of submissions on Canvas.
8. As a response to the last question of this problem set, please specify who you collaborated with, and also include all the resources you consulted to answer questions in this problem set, including URLs of pages you visited on the Internet. Also specify which question and aspect you got help with. Please be as thorough and complete as possible here. It is mandatory to answer this question.
9. If you have any questions or clarifications on the problem set, please post them on Piazza or contact the course staff. We are here to help.

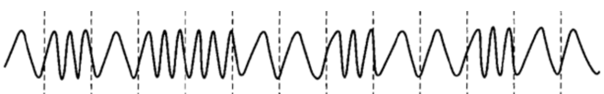
Questions

There are 11 questions in this problem set totalling to 50 points.

(1) Layering (2 points) Why is Internet software/hardware arranged in layers?

That is because the Internet software/ hardware is a very difficult task. Layering provides modularity.
Layering simplifies understanding, testing, maintaining. When people find a mistake, it is easy to improve or replace protocol at one layer without affecting others. Also, each layer is well-defined function and interfaces to layers above and below it.

(2) Modulation type (2 points). A digital signal contains the sequence of bits `0101100100100`. What modulation is being used in the physical signal (for example, voltage plotted against time) below? Please explain how you arrived at your conclusion.



Frequency modulation, because we can see in the image that the curve between the first gap is low frequency, and in the second gap, that is a high frequency. Also, the same image was in the lecture slide 02, page 6

(3) Switching types (2 + 2 = 4 points). For each question below, your answer must be one of "circuit", "message", or "packet", along with a justification.

(a) Suppose you'd like to design a network with routers that need to support application conversations that transfer a lot of data, but only between a few pairs of endpoints. Which kind of switching would you use for the network's routers? Please justify your answer (2 points).

circuit, since there are only few pairs of endpoints, we should reserve the resources. Also, since we have lots of data, it is safe to reserve the resource

(b) Suppose, instead, your network must support applications that move a small number of application-level messages between a large number of pairs of endpoints. Each application-level message may be very large. Further, even if the entire message isn't received yet, it is helpful for the applications to transfer smaller portions of the messages as early as possible to the receiving endpoints. What kind of switching would you build into the network's routers? Please justify your answer (2 points).

I think we should use packet switching, because when a message you sent is very large, the packet switching will separate this data into different packets, and each packet will have its own header. In this case, we can still receive portion of our data while we wait for the rest of data.

(4) Bandwidth and delay (2 + 2 + 1 + 1 = 6 points).

(a) Is it possible for a link or network path to have high bandwidth and high propagation delay? Why or why not? Justify.

Yes, because bandwidth and propagation delay are independent. Bandwidth means how many data it can transmit per unit time. And propagation means time to move one bit across. Clearly, those two things are not relative.

(b) Is it possible for a link or network path to have low bandwidth and low propagation delay? Why or why not? Justify.

As I state in the previous question, it is possible for a link or network path to have low bandwidth and low propagation delay, because bandwidth and propagation delay are independent.

(c) If you need to transfer a large file very quickly, which kind of link would you rather prefer, one with a higher or lower bandwidth?

I would choose higher bandwidth, because transfer a large file very quickly
require us to put data into the link very fast

(d) If you need to support a highly interactive (real-time) voice call, which kind of link would you rather prefer, one with a higher or lower propagation delay?

I would choose lower propagation delay, because a highly interactive
call requires us to move one bit across very fast

(5) Total delay (3 + 3 = 6 points). An endpoint sends a packet of size 4000 Bytes (1 Byte = 8 bits) over a link with bandwidth 2 Mbit/s (1 M = 10^6). The link has a propagation delay of 10 milliseconds (1 milli = 10^{-3}).

(a) After how long since the sender pushed the first bit of the packet into the link does the final bit of the packet arrive at the other end of the link? Explain your answer. (3 points)

bandwidth: 2Mbit/s = 2 * 10^6 bit/s
size: 4000byte = 4000 * 8 bits
transmission delay: (4000*8) bits / 2 * 10^6 bit/s = 16 * 10^-3 s
propagation delay: 10 milliseconds = 10 * 10^-3 s
total delay: 16 * 10^-3 s + 10 * 10^-3 = 26 * 10^-3 s

(b) At the other end of the link is a router which pushes the packet through another link that is identical to the last one (described in question above). The packet encouters a small queue with queueing delay 5 milliseconds before entering this second link. The router uses store and forward packet switching. What is the **total** time it takes to move the packet end to end, i.e., first bit entering the first link to last bit exiting the second link? Explain your answer (3 points).

queueing delay = 5 milliseconds = 5 * 10^-3 second

From previous question, we know that transmission delay: 16 * 10^-3 s, and propagation delay is 10 * 10^-3 s. However, since we add another identical link, so the transmission delay is now (16 * 10^-3) * 2 s, and propagation delay is (10 * 10^-3) * 2 s.
The total time is transmission delay + propagation delay + queueing delay = 0.057 s

(6) App-layer connection (6 points).

(a) What are the components of the 4-tuple denoting an application-layer connection?

We need to denote A, B as two endpoints
<IPA, PortA, IPB, PortB>

(b) Why could you not just use the 2-tuple (IP_A, IP_B) to denote an application-layer connection?

If we just use 2-tuple, we cannot distinguish different application, such as email, web.
Because IPA and IPB are two host addresses. Without the port, we cannot access different address at the same time.

(c) Could you think of an example where IP_B and $port_B$ have the same values across many application-layer connections?

One example is that, when B is a web server, and there are lots of web clients. In this case, B have to provide web service to those web clients

(7) Client and server (2 points). Explain one difference between a “client” and a “server” in an application.

Server is always online, and it should always wait for the request.
Client sends its request to server, and server will handle this request and respond back to client

(8) Domain Name System (2 * 6 = 12 points).

(a) Why do we need the domain name system (DNS)? What problem does it solve?

The IP address is difficult to remember, because normally it contains 12 digits.
Therefore, we can just remember the domain name, instead of the whole IP address. And the DNS will help us map the domain name and respond back with its IP address

(b) Explain how *hierarchy* helps DNS meet the scaling needs of serving Internet users.

The hierarchy partition different domain names to TLD, authoritative name server, DNS server. In this case, TLD will only need to be responsible for part of mapping.

(c) Explain how *replication* helps DNS meet the scaling needs of serving Internet users.

Replication will balance DNS servers query load
Sometimes, we will have millions of users online, Therefore, if we only have one server, it will overcrowd. Thus, we have replication to scale out at each level of hierarchy, so that it will prevent a server overwhelmed.

(d) Explain how *caching* helps DNS meet the scaling needs of serving Internet users.

When you search the domain name on local server that is stored in the cache, the local server will jump directly to the IP address without doing the iterative or recursive query. In this case, even if a DNS server breaks down, the caching will still respond the IP address you are looking for. Also, caching can reduce the other DNS servers query load

(e) What is the purpose of the *identification* field on a DNS protocol message?

Identification is a part of the message header. It contains 16 bits for query. It will be copied into the reply message, so that the client can match the received query

(f) What is an authoritative DNS server?

Authoritative DNS server is the final step of mapping the domain name to IP address. The authoritative DNS server helps provide publicly accessible DNS records that can map the domain name to IP address.

(9) Finding authority (3 points). Can you list a set of authoritative DNS servers for `cs.princeton.edu`? Explain how you arrived at the answer.

cs.princeton.edu.	21600	IN	NS	ns6.dnsmadeeasy.com.
cs.princeton.edu.	21600	IN	NS	ns5.dnsmadeeasy.com.
cs.princeton.edu.	21600	IN	NS	ns7.dnsmadeeasy.com.
cs.princeton.edu.	21600	IN	NS	dns2.cs.princeton.edu.
cs.princeton.edu.	21600	IN	NS	dns1.cs.princeton.edu.

I need to type dig -t NS cs.princeton.edu in the command prompt. NS means the DNS records type is NS, and it returns the hostname of authoritative name server.

(10) Partitioning, caching, and DNS query load (7 points). Suppose there are N domain names available in the Internet, equally subdivided over T top-level domains (such as `.com`). Suppose there are C local DNS resolvers, each issuing $N*k$ queries, i.e., k queries to each of the N domains, where $k > 1$. Assume that each of the N domains has a unique authoritative name server with an entry in the corresponding top-level DNS server. Further assume that queries are iterative from the point of view of the local DNS resolver.

Now answer the questions below. Explain your reasoning clearly to receive partial credit.

(a) Suppose there is no caching on any of the DNS servers. How many queries does the root DNS server service? (1 point)

There will be C*N*k queries, because we are using iterative so every queue needs to go from the root server.

(b) Suppose there is no caching on any of the DNS servers. How many queries does each top-level DNS server service? (2 points)

(C*N*k)/T, because TLD only need to take care of his part of mapping. In this case, TLD will only have queries from the root DNS server. Since there are T TLD servers, the answer is (C*N*k)/T

(c) Suppose all local DNS resolvers cache the DNS responses they receive indefinitely. Now how many queries does the root DNS server service? (2 points)

C * N, because since we have caching, we can map the IP address directly. Since there are N domain names, the answer is C * N

(d) Suppose all local DNS resolvers cache the DNS responses they receive indefinitely. How many queries does each top-level DNS server service? (2 points)

(C * N)/T, because the caching is involved, and root DNS server have C * N queries, and there are T TLD servers, so the answer is (C * N)/T

(11) Collaboration and References (mandatory). Who did you collaborate with on this problem set? What resources and references did you consult? Please also specify on what questions and aspects of the problem set you got help on. If you did not consult any resources other than the lecture slides and textbook, just say "no collaboration".

PPT from the lectures, recitation, TA from section 02
I also discussed problem 8 with my partner: jj740