Varun Narravula
ID: 923-287-037

Prof. Qun Wang
CSC-645 Computer Networks

Homework 2

1. DNS Servers

Describe the three kinds of DNS servers and their functionalities in the hierarchy of DNS systems.

Root DNS servers are DNS servers that get queried by recursive resolvers (i.e. Cloudflare 1.1.1.1) for a given domain. They will refer to TLD DNS servers for the TLD of the domain that was requested. The TLD DNS servers will eventually pass back the real IP address after it does its own referral, described next.

Top-Level Domain (TLD) DNS servers are intermediate DNS servers that receive referrals from root DNS servers. They are segmented by TLD (i.e. `.com`, `.edu`, `.net`, etc.), and they will refer to the authoritative DNS servers for the domain that was requested. Once they get the destination IP address from the authoritative DNS server, they will send the address back to the root DNS server.

Authoritative DNS servers are DNS servers that contain the true resource records for a given domain. They will respond to queries for that domain, usually from TLD DNS servers, and return the IP address for that domain. As such, they do not refer to any other DNS servers for their lookups, and are the last in this chain of DNS resource lookups.

Suppose the host cs.sfsu.edu wants to know the IP address of `gaia.cs.umass.edu`. Also suppose that the local DNS server of SFSU is called `dns.sfsu.edu`, and an authoritative DNS server for gaia.cs.umass.edu is called `dns.umass.edu`. Describe the interaction of the host and DNS servers (including the local DNS server).

The local DNS server for cs.sfsu.edu will send a query to the root DNS server for the IP address of `gaia.cs.umass.edu`. The root DNS server will then refer to the TLD DNS server for `.edu`, which will refer to the authoritative DNS server for `gaia.cs.umass.edu`, which is `dns.umass.edu`. Finally, the authoritative DNS server at dns.umass.edu will return the IP address for `gaia.cs.umass.edu`. This gets passed back up the chain to the root DNS server, which sends it back to the local DNS server as the response for the DNS query.

A recursive resolver (i.e. Cloudflare 1.1.1.1) can be in-between the local and root DNS servers, and can serve as a large-scale DNS cache. If the address is already in the cache, it will be returned directly to the local DNS server by the recursive resolver.

2. Reliable Data Transfer Protocols

In the reliable data transfer protocols, why do we need to introduce sequence numbers?

Reliable data transfer protocols like TCP need sequence numbers mostly to ensure that packets are received in the correct order. Also, sequence numbers can be used to detect packet loss, duplicate packets, out-of-order packets and other such issues. They can be also used to ensure proper buffering techniques so that packets can be delivered in such a way that the server isn't overwhelmed.

Why do we need to introduce ACK/NAK?

ACKs and NAKs are used to ensure that packets are received in the correct order, and that no packets are lost. NAKs would indicate explicit error states, such as a packet being lost in some way, while ACKs can indicate that the packet was successfully received and/or what packets are expected next.

Why do we need to introduce timers?

Protocols like TCP do not have explicit error states with NAKs, and instead rely on timeouts to detect packet loss, especially in the case of TCP Tahoe.

3. Go Back N and Selective Repeat

Visit the online animations for Go Back N and Selective Repeat.

a.

Have the source send five packets, and then pause the animation before any of the five packets reach the destination. Then kill the first packet and resume the animation. Describe what happens and why.

When the first packet (we'll call it P1) is lost, the receiver cannot ACK any subsequently received packets, since the Go Back N protocol requires that packets be received in order. Therefore, the receiver will discard all packets after P1. The sender will wait for a timeout, and then it will re-transmit all packets starting from P1 again up until P5.

b.

Repeat the experiment, but now let the first packet reach the destination and kill the first acknowledgment. Describe again what happens and why.

When the sender fails to receive an acknowledgment for P1 (we'll call it ACK1), it will assume the packet was lost, and will re-transmit starting from P1 again. When the receiver gets the re-transmitted P1, it will still discard all packets starting from P1, even if it has received P2, P3, and so on, since the receiver doesn't support out-of-order buffering. Then, the receiver will wait for the re-transmitted subsequent packets after P1 to be re-transmitted, and this continues.

c.

Repeat a and b with the Selective Repeat and answer the above questions again.

In the first scenario, when P1 is lost, the receiver will not discard all packets like it does in Go Back N, since this protocol supports out-of-order transmission. Instead, after P2-P5 are ACKed, and not P1, the sender will notice that it has not received ACK1, and will re-transmit P1 again. Once it receives an ACK1, all packets will have been transmitted successfully, since P2-P5 are already buffered after being ACKed.

In the second scenario, when ACK1 is lost, the receiver will simply re-transmit P1 after it notices it has not received an ACK1. Similar to the first scenario, after P1 is re-transmitted by the sender, the receiver will send an ACK1 again, and since the receiver has already buffered P2-P5, transmission is complete.

4. Sequence and Acknowledgement Numbers

Host A and B are communicating over a TCP connection, and Host B has already received all bytes up through byte 126 from A. Suppose Host A then sends two segments to Host B back to back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127. Host B sends an acknowledgment whenever it receives a segment from Host A.

a.

In the second segment sent from Host A to Host B, what is the sequence number?

207; this is the sequence number starting from 127 and adding 80 for the second.

b.

If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

Host B will ACK the start of the next segment it expects, which is again 207.

c.

If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

Host B has not received the first segment from 127-206 first, so the ACK number will stay at 127 until the first segment that starts from 127 is received. This is because TCP is in-order.

d.

Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number; for each acknowledgment that you add, provide the acknowledgment number.

```
Time   |      A → B (SEQ)      |      B → A (ACK)
------------------------------------------------------
T0     | [SEQ 127, 80 bytes]   |
T1     | [SEQ 207, 40 bytes]   |
T2     |                       | [ACK 207] (Lost)
T3     |                       | [ACK 247] (Delayed)
```