INST346 HW02

The Application Layer

September 27, 2017

1 Delay, Loss, and Throughput

1. Consider sending a packet from a source host to a destination host over a fixed route. List the delay components in the end-to-end delay. Which of these delays are constant and which are variable?

   Solution Delay is composed of four parts: 1) processing, 2) propagation, 3) transmission, 4) queuing. Components 1-3 are fixed/constant, and queuing delay is variable.

2. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 500 kbps, R2 = 2 Mbps, and R3 = 1 Mbps.

   a) Assuming no other traffic in the network, what is the throughput for the file transfer?

   Solution Throughput without other traffic is the minimum of all link rates, or $\min(R_1, R_2, R_3) = R_1 = 500$ kbps.

   b) Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?

   Solution $\frac{\text{4,000,000 bytes}\cdot\text{(8 bits / byte)}}{\text{500 kbps}} = \frac{\text{32,000,000 bits}}{\text{500,000 bps}} = 64$ seconds.

   c) Repeat (a) and (b), but now with R2 reduced to 100 kbps.

   Solution New throughput is $R_2 = 100$ kbps. Therefore, new time for transfer is $\frac{\text{32,000,000 bits}}{\text{100,000 bps}} = 320$ seconds.

3. Suppose you need to deliver a 40-terabyte data set from Boston to Los Angeles. You use FedEx overnight, and the delivery person drops off the drive containing this data exactly 24 hours after you package and send the drive.

   a) What is the average throughput of this data transfer?

   Solution

$$\text{Average} = \frac{40\text{ TB}}{24\text{ hours}} = \frac{40 \cdot 1,024^4 \cdot 8\text{ bits}}{86,400\text{ seconds}} \approx \frac{351,844\text{ Gigabits}}{86,400\text{ seconds}} \approx 4.07\text{ Gbps}$$

   b) What is the median instantaneous throughput of this data transfer (For simplicity, we will specify "instantaneous" as the transfer speed at a particular second)?

   Solution 0 bps. 86,399 seconds at 0bps, and 1 second at 352 Tbps.

2 Network Security

1. Describe how a botnet can be created, and how it can be used for a DDoS attack.

   Solution Botnets are often created from many independent machines, most of which have been exploited in some way. Exploits, which can allow for remote code execution, can be digital (e.g., malware) or social in nature (e.g., convincing a user to run a malicious application or document), and once an attacker has remote code execution privileges on a machine, he/she can direct that machine to download other malware, visit a website, or use some service. With *many* of these exploited machines working in concert, a botnet operator can coordinate large-scale denial-of-service attacks distributed over many networks. As a result, network security professionals are often unable to inhibit the DDoS attacks since they cannot shut off all links to the Internet (as opposed to a centralized DoS attack, in which a security professional could call the source ISP and ask them to shut off access for the attackers).

2. Suppose Alice and Bob are sending packets to each other over a computer network. Suppose Sybil positions herself in the network so that she can capture all the packets sent by Alice and send whatever she wants to Bob; she can also capture all the packets sent by Bob and send whatever she wants to Alice.

   a) List some of the malicious things Sybil can do from this position.

      Solution Sybil can pretend to be Bob to Alice (and vice-versa) and partially or completely modify the message(s) being sent from Bob to Alice, what's called a "man-in-the-middle" attack. For example, Sybil could intercept credit card data or other confidential information being shared between Bob and Alice without either partying knowing. She can also easily modify messages by changing the phrase "Alice, I owe you $100" to "Alice, I owe you $10,000". Furthermore, Sybil can drop packets sent by Bob to Alice (and vise-versa), even if the packets from Bob to Alice are encrypted.

   b) List some techniques that could be employed to counter the threats you listed above (if any).

      Solution Assuming a secure key infrastructure (a big assumption), encryption could be used to counter man-in-the-middle attacks, since encryption can guarantee confidentiality and integrity. Sybil's ability to drop packets, however, is only counter-able by having redundant links between Bob and Alice. This inability to address potential malicious packet loss is part of the reason why "network neutrality" is an issue.

3. Recall that TCP can be enhanced with SSL to provide process-to-process security services, including encryption. Does SSL operate at the transport layer or the application layer? Justify your answer.

   Solution In some sense, SSL operates between these layers, so either answer is appropriate as long as it is justified.

   At a fundamental, operating system-implementation level, SSL operates at the application layer. The application developer must include SSL as a library when designing the application and uses SSL as a wrapper around the OS's exposed socket interface. That is, SSL is **not** implemented by the OS in the transport layer, so it can be said to be an application-layer protocol. (I would consider this answer to be the most correct)

From the standpoint of an application developer, a development platform may provide SSL-type support, and SSL can appear to exist below the application layer as a result. That is, the developer interacts with SSL-enabled sockets in a manner similar to the standard socket interface. Therefore, in another sense, SSL can be seen as an extension on the existing transport-layer services.

3 Client-Server and P2P Models

1. List two different applications that are naturally suitable for P2P architectures and explain why P2P would be better than client-server.

   Solution Some common examples include peer video conferencing (where P2P is useful to reduce network traffic if two users are closer to each other than they are some third-party server), low-cost distributed file sharing (P2P distribution times can be lower, and infrastructure can be cheaper), fault-tolerant systems like root-level DNS (P2P is better-suited for disaster recovery), and other use cases where users may still want to communicate with their neighbors even if the full network is segmented or broken (e.g., country-level services when a trans-Atlantic cable is cut, or local apartment-wide mesh networks that can exist without a broader Internet connection).

2. List two different applications that are naturally suitable for client-server architectures and explain why.

   Solution Centralized, proprietary services are generally well-suited for client-server because the company can exhibit full control over the server hardware and distribution network. E-commerce sites are similarly well-suited for client-server because the site needs to have confidence in the order-fulfillment pipeline (i.e., they need to control the server), and the server may be directly connected to the warehousing system (e.g., Amazon). Other examples include high-performance computing environments in which a single high-performance cluster is shared among researchers, as these applications need specialized hardware.

3. State whether you agree or disagree with the following statements, and justify your answer.

   a) Sharing a file via BitTorrent gives the publisher confidence that the file will be available regardless of disaster or censorship.

      Solution I would agree with this statement. While BT is not a foolproof solution for disaster tolerance or censorship, BT does provide a measure of support for both of these things. For fault tolerance, even if the network is cleaved in two, individuals in either subnetwork can access each other and continue to share even if they no longer have access to the original server. In case of censorship, it is more difficult for a government to censor millions of peers than it is to censor a single company's servers.

   b) Files shared via BitTorrent are more secure against unauthorized access than files shared via HTTP.

Solution I would disagree with this statement. BT has limited support for authentication, so a bad actor could easily gain access to the swarm by publishing a fake or modified torrent file and share bad data. Furthermore, with enough resources, a bad actor could find collisions in BT's hashing mechanism and share bad content even in an established swarm. In an HTTP-based client-server environment, it's more difficult for a bad actor to interact with the HTTP client unless the bad actor exists between the client and server.

4 Transport Services and Network Communication

1. Suppose you wanted to do a transaction from a remote client to a server as fast as possible. Would you use UDP or TCP? Why?

   Solution You would use UDP because it does not have the connection handshake overhead, has a smaller packet header, and disregards congestion control.

2. Suppose you wanted to do a transaction from a remote client to a server and it was critical to ensure your transaction was processed and processed correctly. Would you use UDP or TCP? Why?

   Solution You would use TCP because it guarantees error-free delivery. That is, if the data can be transmitted to the destination, TCP can ensure it is eventually received correctly.

3. Describe an application that requires no data loss and that is also highly time-sensitive?

   Solution Examples of such applications can be found in medical and military realms. For example, remote medical surgery would likely require no data loss in visual and sensory information and remote control, and time sensitivity would be highly important, so the doctor can respond to changes in the patient's condition rapidly. Remote drone control, dog fighting, or remote radar guidance might also be good examples.

5 HTTP, DNS, and the Web

1. Explain why you think HTTP uses TCP instead of UDP.

   Solution For use cases like web pages, we want to ensure *all* data is transmitted and received successfully, so our web pages don't have missing chunks of content. For instance, visiting your online banking site and only seeing part of your balance could be quite distressing. While HTTP could have been developed on top of UDP and still ensure reliable transfer, HTTP would have to handle all the reliable transmission itself, thereby complicating the protocol. Furthermore, web pages are not time-critical, so the built-in congestion control present in TCP is useful since it provides fair and equitable bandwidth allocation.

2. Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects? Why?

Solution In its simplest form, web caching can reduce delay in receiving objects by eliminating the need to perform any network traffic at all. If you are constantly downloading the Google logo, your computer could store a **local** copy of this logo in its local cache, and any time you request the particular URL that matches the Google logo, the browser just pulls the logo from the cache instead of requesting it from a remote server.

Large organizations also use *caching servers* to aggregate often-accessed data across the whole organization. In this way, anyone in the organization who requests Google's logo could get the response from the local *caching server*, which will be much closer and generally have a much higher-speed connection to the client.

Web caching *cannot* reduce the delay for *all* objects though. Consider the case of breaking news, the Facebook stream, or Twitter's feed. Much of this content would be dynamic and therefore impossible to cache. Therefore, while you may reduce delay for Facebook's logo or Twitter's logo, you will still need to connect to Facebook/Twitter's remote server to get your most recent feed data.

3. Obtain the HTTP/1.1 specification (RFC2616). Answer the following questions:

   a) Explain the mechanism used for signaling between the client and server to indicate that a persistent connection is being closed. Can the client, the server, or both signal the close of a connection?

   Solution Persistent connections are discussed in section 8 of RFC 2616. Sections 8.1.2 and 8.1.2.1 of the RFC indicate that either the client or the server can indicate to the other that it is going to close the persistent connection. It does so by including the connection-token "close" in the Connection-header field of the HTTP request/reply.

   b) Can a client open three or more simultaneous connections with a given server?

   Solution Indeed so. While the RFC states clients should *not* maintain more than 2 such connections, no web server software I know of enforces this limit. In fact, IE 8 opens 6 simultaneous, persistent connections. More information is available here: `http://goo.gl/sV3xx9`

4. Answer the following questions regarding DNS:

   a) What is a *whois* database?

   Solution For a given input of domain name (such as ccn.com), IP address or network administrator name, a *whois* database can be used to locate the corresponding registrar, whois server, DNS server, and so on.

   b) Use the ARIN whois database to determine the IP address ranges used by the University of Maryland.

   Solution According to ARIN, the University of Maryland has six IPv4 ranges and four IPv6 ranges:

   65.127.220.0 - 65.127.221.255

   128.8.0.0 - 128.8.255.255

   129.2.0.0 - 129.2.255.255

192.54.94.0 - 192.54.103.255

199.7.91.0 - 199.7.91.255

206.196.160.0 - 206.196.191.255

2001:468:C00:: - 2001:468:CFF:FFFF:FFFF:FFFF:FFFF:FFFF

2001:500:2D:: - 2001:500:2D:FFFF:FFFF:FFFF:FFFF:FFFF

2605:880:: - 2605:880:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

2620:50:: - 2620:50:F:FFFF:FFFF:FFFF:FFFF:FFFF

c) How can an attack leverage open DNS resolvers to conduct DDoS attacks?

Solution An attacker can use these open DNS resolvers to perform amplification attacks. These attacks involve sending specially crafted small queries with spoofed source addresses to the resolvers. These small queries result in large responses from the DNS servers, and since the source address was spoofed, the DNS server sends this large response to the target.

With enough of these open DNS servers all sending large responses to a target, one can marshal on the order of gigabits per second of sustained attack.

5. Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

Solution We can periodically take a snapshot of the DNS caches in the local DNS servers. The web server that appears most frequently in the DNS caches is the most popular server. This is because if more users are interested in a Web server, then DNS requests for that server are more frequently sent by users. Thus, that Web server will appear in the DNS caches more frequently.

For a complete measurement study, see: Craig E. Wills, Mikhail Mikhailov, Hao Shang "Inferring Relative Popularity of Internet Applications by Actively Querying DNS Caches", in IMC'03, October 27-29, 2003, Miami Beach, Florida, USA

6. Assume you are developing a video streaming service to compete with Netflix. You want to make your service cheaper by reducing your infrastructure costs (e.g., dollars spent on bandwidth, file transfers, and building out data centers). Describe a content distribution strategy that may help you achieve these goals.

Solution If your customers were willing to provide local storage and some bandwidth, you could use a P2P delivery strategy that could substantially reduce your infrastructure costs. By sharing the data storage and communication burden among your clients, you would need fewer content servers and fewer high-speed links. You would have a lot of coordination to manage among your customers, but you could do it with something similar to BitTorrent.

6 Socket Programming

1. A UDP server needs only one socket for accepting data, whereas TCP servers need two sockets. Why?

Solution UDP does not use welcoming sockets since it does not maintain an open stream of communication between hosts. TCP instead uses this welcoming socket to coordinate access and creation of these streams between hosts.

2. If a TCP server were to support n simultaneous connections, each from a different client, how many sockets would the TCP server need?

Solution Based on the answer above, if a TCP server were to support n simultaneous connections, it would need $n + 1$ sockets, one for each host plus the welcoming socket.