CS158A Project 3: Final Draft

Group Bits Per Pixel: Vinson La, Clayton Chan, Mike To

Project Description:

In this project, two primary programs are used – a server and client program. The server and client program were coded in C using sockets. The goal of the project is to analyze the transmission of messages across the server and client programs under different conditions.

The two protocols being examined are UDP and TCP. The first condition being examined is message size. The message sizes tested are 1, 1024, 4096, 8192, 16384, 32678, and 64000/65536 bytes. The second condition examined is round-trip time in seconds. The third condition is throughput in bytes per second. Each protocol is measured for each condition individually. Then, the protocols run together simultaneously for each condition.

Lastly, TCP has two different options set. With the options enabled, the round-trip time and throughput are recorded.

Performance:

UDP: Round-Trip Time vs. Message Size

1 Client and Server on same LAN

TCP: Round-Trip Time vs. Message Size 2 and TCP: Throughput vs. Message Size 2

1 Client and Server located on different LAN, TCP options disabled

TCP: Round-Trip Time vs. Message Size 2 and TCP: Throughput vs. Message Size 2

1 Client and Server located on different LAN, TCP options enabled

TCP: Round-Trip Time vs. Message Size 3 and TCP: Throughput vs. Message Size 3

2 Clients (one on same LAN, one on different LAN) and Server

UDP/TCP: Round-Trip Time vs. Message Size

Discussion of Results:

The “UDP: Round-Trip Time vs. Message Size 1” graph shows that the average round-trip time grows nearly exponentially as the message increases by a factor of 2. Since these values were done on the same LAN, these values represent the lowest average round-trip time values possible.

The “UDP: Throughput vs. Message Size 1” graph shows significant inconsistencies. For message size 4096 bytes and 16384 bytes, the throughput is nearly identical while message size 8192 has nearly twice the throughput of those message sizes. One possible cause of this is lost packets. By having more retransmitted messages, throughput would increase.

Unlike UDP, TCP shows a near linear growth in RTT. Also, there were more random spikes in delay occurring throughput the TCP graphs. In TCP RTT 1 graph, there are no random spikes in delay. However, in TCP RTT 2 graph, 1 byte messages have almost as much RTT delay as 4096 byte messages. TCP RTT 3 graph, 1024 byte messages have as much RTT delay as 8192 byte messages. One possible cause of this is congestion. The server must have been overloaded with the segments and took slightly longer to read all the segments. One notable phenomenon occurring is when one TCP client is sending, the throughput from message size 8192 to 16384 nearly doubles as shown in TCP throughput 1 and TCP throughput 2 graphs.

TCP RTT 1 graph and TCP throughput 1 graph show data for TCP options off while TCP RTT 2 graph and TCP throughput 2 graph show data for TCP options on. With options on, RTT ranged from 0.18 to 0.95 seconds while options off ranged from 0.09 to 0.93 seconds. With options on throughput ranged from 11380 to 137972 bytes per second, while options off ranged from 13656 to 140940 bytes per second (ignoring the negligible throughput for 1 byte message sizes). With options enabled, the TCP protocol must have sent more header data which led to a slight increase in RTT. Since more total data was sent with the options enabled, it led to an increase in throughput.

TCP RTT 3 graph and TCP throughput 3 graph show data for multiple clients sending to a single server. Client 1 is located on a different LAN and Client 2 is located on the same LAN as the server. For Client 1, the values for RTT are only slightly higher than those on TCP RTT 1 graph except for message size of 1024 bytes in which the RTT doubled when two clients were sending data to the server. Even when the TCP server receives large message sizes, it is seen that there was no drastic increase in RTT occurs for the client located on a different network. Thus, the server must be able to handle connections with no significant performance drop when there are more than 2 clients.

UDP/TCP transmission both for RTT and throughput start off roughly the same. TCP, however, have longer round-trip time and have less throughput. Although the TCP and UDP client are on the same LAN as the server, UDP is faster though inconsistent. Since TCP's header carries more data than UDP's header, it has significantly higher RTT.

List of References:

Our textbook, Chapter 1, Sec. 1.4

Instructor provided: sockets\_m.pdf

http://rites.uic.edu/~solworth/sockets.pdf