Lab 3: File Servers, Traffic Generators, and Monitoring

Duc Viet Le
CS536

October 15, 2016

Problem 1.
For this problem, I use an audio file (i.e.) with file size of 3183357 bytes ($\approx$ 3.1 MB). I use both check sum and Linux `diff` to check for the authenticity of the downloaded files. With TCP, we did not have any corrupted files.
Below are the results of time and through put for different package sizes:

Package size	Time	Reliable Through Put
1 byte	5265 ms	4,836,939 bps
10 bytes	590 ms	43,152,693 bps
50 bytes	276 ms	92,095,696 bps
500 bytes	274 ms	92,382,231 bps
1000 bytes	273 ms	93,115,342 bps
1500 bytes	272 ms	92,695,123 bps
2500 bytes	273 ms	93,255,654 bps
10000 bytes	275 ms	92,895,832 bps

Below is plotted graph of the datas:

While the small package size may limit reliable through put, much bigger package size did not have much impact on throughput. As the package size increases, the reliable throughput and transmission time did not change. Thus, to improve the server performance, I may choose package size to be power of two because it may be helpful for computers and kernels, not too big so that we can fit inside processor (i.e L2 cache) ...

Problem 2.
For payload-size = 1000 bytes, packet-count = 1000. We obtain the following result with different packet-spacing:
`Traffic_rcv` was run at *sslab*01. `Traffic_snd` was run at my home.

	Sender			Receiver		
Packet Spacing	Time	BPS	PPS	Time	BPS	BBS
50ms			19.947 pps	50.132s	BPS	BBS

Problem 3.