Luis Emilio Alcántara Guzmán A01027304
Ana Paola Minchaca García A01026744
Quantitative Methods and Simulation

Final Project

I. Goal. To identify, describe and implement a probabilistic TCP/IP congestion control algorithm.

We can start by explaining what TCP is, TCP stands for Transmission Control Protocol, a communication standard that enables application programs and computing devices to exchange messages over a network. It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks. A typical TCP flow evolves through three phases: connection, transfer and close. Although at the midst of this connection there comes a time where incoming traffic exceeds outgoing bandwidth, this is called a congestive collapse and it generally occurs at choke points in the network. When a network is in this condition, it settles into a stable state where traffic demand is high but little useful throughput is available and achieved, during which packet delay and loss occur and quality of service is extremely poor. So, to avoid this, TCP uses a multi-faceted congestion-control approach. For each connection, TCP maintains a *congestion window* or the *cwnd,* limiting the total number of unacknowledged packets that may be in transit end-to-end. The *congestion window* is maintained by the sender and is a means of stopping a link between the sender and the receiver from becoming overloaded with too much traffic, this window is calculated by estimating how much congestion there is on the link. When a connection is set up, the *congestion window*, a value maintained independently at each host, is set to a small multiple of the maximum segment size (MSS) allowed on that connection.

Another part of the TCP congestion control strategy is slow start, this is used in conjunction with other algorithms to avoid sending more data than the network is capable of forwarding, that is, to avoid congestion. Slow start begins initially with a congestion window size of 1, 2, 4 or 10 MSS. The value for the congestion window size can be increased by 1 MSS with each acknowledgement received, effectively doubling the window size each RTT. The transmission rate will be increased by the slow-start algorithm until either a packet loss is detected, or the receiver's advertised window (rwnd) is the limiting factor. There is also another factor which determines whether we continue with the normal procedure or if there is a need to use an alternate congestion control algorithm, this factor is called the slow start threshold. When the congestion window is below the threshold then normal TCP congestion

control procedures are used. This means that the alternate congestion control algorithm will be invoked only when a flow's congestion window surpasses the threshold.

But, what are exactly these congestion control algorithms? Congestion control algorithms are being developed with a focus on optimizing different metrics and making trade-offs between the various metrics depending on the desired working environment and use cases. There is a wide range of algorithms available to apply congestion control and these vary on the type and amount of feedback they received from the network, the aspect of performance and the fairness criterion it used. Some known algorithms which were mentioned in the Study of Proposed Internet Congestion Control Mechanisms are the following:
Binary Increase Congestion control (BIC), Compound TCP (CTCP), Fast Active Queue Management (AQM) Scalable TCP (FAST), High-Speed TCP (HSTCP), Hamilton TCP (H-TCP) and Scalable TCP. Our main focus being the FAST algorithm which will be explained below.

TCP Fast Open

TCP Fast Open is used for traffic congestion when it has a large congestion window, when talking about fast TCP we know that it aims to achieve an equilibrium congestion window that does not change during the life of a flow, whereas other algorithms use an oscillating congestion window.

The main goal of TCP Fast Open (TFO) is improving the performance of short transfers such as the retrieval of web objects. A brief description of TCP is that it enables data to be exchanged safely during TCP's initial handshake. The core of TFO contains a security cookie that allows the server to authenticate the client that started the protocol.

This cookie contains an encrypted data string that is used in the server to validate the ownership of the IP of the client. When the server gets that generated cookie, it encrypts the IP obtained from the client and generates a cookie of length up to 16 bytes. The encryption and decryption used for such a process is rather fast compared to regular processing of packets. After a certain time those cookies generated in the server are revoked, this for security purposes and given that the client's IP will change eventually.

Luis Emilio Alcántara Guzmán A01027304
Ana Paola Minchaca García A01026744
Quantitative Methods and Simulation

The server maintains a counter of total pending TFO connection requests. When the number of pending TFO connections exceeds a certain threshold , it falls back on regular three way handshake. That mechanism makes this algorithm really safe in case of an attack, and doesn't affect the service itself.

The advantage of TCP Fast Open is that it will help many webpages to reduce its loading time; this will not only benefit the owner of the website, but also the end-user since it will make it faster for its request to be processed. TCP Fast Open would decrease HTTP transaction network latency by 15% and whole-page load time over 10% on average, and in some cases up to 40%.

Latency and page load time is important even for small websites, this is directly related to user satisfaction with a website, and how many users will end up using someone's web page. Even when the loading time is slightly better it will show an impact in how the user experiences feels. And for the owners of such websites will benefit from the revenue generated from views and ads.