Measuring Internet Censorship in Iran from Multiple Vantage Points

Neeki Hushyar
UMass Amherst
nhushyar@cs.umass.edu

Angela Upreti
UMass Amherst
aupreti@cs.umass.edu

Keywords

TCP variants; Comparison; SACK; Reno; New Reno, Vegas; Tahoe

Web traffic runs over TCP and given the scale at which the internet has grown and how much we rely on the web, it is important that the underlying transport protocol is able to operate at a reasonably low latency and provide high throughput. Network congestion causes packet loss which increases latency and hence, reduces throughput. Using the right TCP variant saves both money and time. Amazon found that every 100ms latency cost then 1 percent in sales [1]. Higher latency/wait-time in web applications discourages clients from revisiting the cite. This is specially true for video applications such as youtube. Clients usually close the page if the latency is high. Fairness in a the underlying transport layer is equally important. One flow should not hog most of the network bandwidth and throttle other flows. Otherwise, different customers paying the same amount for internet service would get widely different performance. Metrics such as throughput, average and end-to-end latency, packet drop rate, fairness are different across different tcp variants and the congestion control algorithms they use.

Congestion control algorithm for TCP was proposed by Van Jacobson in response to 1986's 'congestion collapse'. The original TCP proposed by Van Jacobson contained elements such as dynamic window sizing, slow start, additive increase and multiplicative deacrease (AIMD). Since then, several variants of TCP have been proposed. TCP Reno adds fast recovery and fast retransmit to TCP Tahoe. TCP Reno's fast recovery and fast retransmit suffer when multiple packets are dropped from the same window. In presence of multiple packet loss, the TCP pipe often gets drained. NewReno has to wait for retransmit timeout to be able to send data again. Consequently, multiple packet loss leads to low throughput. TCP NewReno was introduced to solve this. When in fast recovery, TCP NewReno interprets a single partial Ack as a signal to retransmit another packet rather. NewReno does not wait for 3 duplicate ACKs before retransmitting. TCP SACK is another TCP variant that solves the multiple packet drop problem in Reno. SACK allows the receiver to acknowledge any non sequential packets it has received. NewReno without SACK is unable to tell exactly which packets are missing. While NewReno can retransmit only one lost packet per RTT, TCP SACK can transmit as many lost packets as the congestion window allows. Variant Vegas uses RTT as an estimate of congestion rather than using packet loss.

We conduct three different simulation studies in this paper:

1. Compare performances of Tahoe, Reno, NewReno, and Vegas under congestion.
2. Compare fairness between different TCP variants.
3. Study influence of queuing disciplines (1)RED and (2) Drop Tail on TCP Sack and TCP Reno.

1. ACKNOWLEDGMENTS

2. REFERENCES

[1] R. Kohavi and R. Longbotham. Online Experiments: Lessons Learned. *Computer*, 40(9):103–105, Sept. 2007.

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4