NF-TCP: Network Friendly TCP

Mayutan Arumaithurai*, Fabian Glaser*, Xiaoming Fu*, K. K. Ramakrishnan†
*Institute of Computer Science, University of Goettingen, Germany,
Email: {arumaithurai, glaser, fu}@cs.uni-goettingen.de
†AT&T Labs-Research, U.S.A., Email: kkrama@research.att.com

I. INTRODUCTION

Peer to Peer (P2P) file sharing applications form a significant part of the Internet traffic. According to a study from Ipoque[1], P2P accounted for between 43% and 70% of the total traffic in Germany, Northern and Southern Africa, South America, Middle East, as well as Eastern and Southern Europe in the recent past. To further optimize throughput, P2P applications open multiple connections. Moreover TCP is inherently designed to be fair among all flows, irrespective of their needs. This results in unfair and potentially poor service for standard delay-sensitive flows that use a single TCP connection.

Based on technological limitations and user expectations, the flows can be broadly classified into those originating from delay-sensitive applications and those from delay-insensitive applications. Delay-sensitive applications include interactive applications, mission critical applications and also web-traffic. Delay-insensitive applications include P2P file sharing, data center backups, software updates, download and play applications such as video on demand and also streaming applications with huge buffers. Users would prefer to have, and Internet Service Providers (ISPs) be able to provide, differential treatment to delay-sensitive and delay-insensitive flows to enhance user experience and reduce network congestion. To meet this challenge, there is a need for a protocol that can be used by delay-insensitive applications. This protocol should be able to dynamically adapt to network conditions and be submissive (not necessarily compete for equal share) to standard TCP flows during congestion. Such a protocol would not only enable delay-insensitive applications to use the network during non-congestion periods, but also tolerate applications that use multiple connections that exploit non-congested paths. We use the term network friendly protocol to describe such a protocol.

Recently, the IETF LEDBAT working group [1] has attempted to develop a network friendly TCP that does not require the support of intermediate nodes. The LEDBAT effort currently comprises a oneway delay-based congestion control mechanism. Earlier, Kuzmanovic and Knightly [2] also developed a TCP Low Priority (TCP-LP) proposal, using one-way packet delay for congestion indication and TCP-transparent congestion avoidance.

We propose Network Friendly TCP (NF-TCP) that is designed to utilize network feedback on time-scales of a Round Trip Time (RTT). The protocol uses feedback similar to Explicit congestion Notification (ECN) [3] with slight modifications to generate congestion notification from intermediate routers earlier. To opportunistically utilize bandwidth during non-congestion periods, NF-TCP is also designed with a probe based bandwidth-estimation module. NF-TCP differs from other (recent) network friendly approaches in 1) the aggressive manner in which NF-TCP exploits available bandwidth by intertwining an available bandwidth measurement component with congestion control; 2) the usage of congestion marking (ECN) instead of delay measurement to identify incipient congestion; 3) exploiting the measurement of available bandwidth estimation, but still be robust to estimation errors by having the congestion avoidance mechanism then adapt its load until the onset of congestion.

[1] http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009

II. NF-TCP CONGESTION CONTROL FRAMEWORK

NF-TCP is designed in a modular fashion so that the measurement-based available bandwidth estimation informs the congestion control but the latter is not strictly dependent on the timeliness or accuracy of the estimate.

NF-TCP begins with a slow-start phase similar to that of standard TCP. However, it seeks to take advantage of an associated available bandwidth process that uses a probe-based approach to obtain a rough estimate of the available bandwidth. It uses this rough estimate as a safe guideline to increase its sending rate rapidly up to that measured value. However, the increase is guided by any explicit congestion marks (ECN) that are received so as to quickly respond to the onset of congestion as well as any inaccuracies in the estimate of the available bandwidth. On detecting congestion, it enters into a submissive phase, wherein it aggressively gives up its share of bandwidth to standard TCP flows and competes for equal share if and only if all the other existing flows are NF-TCP flows. To have the network provide an earlier indication of the onset of congestion for NF-TCP, ECN-capable routers use a lower threshold to begin marking NF-TCP flows than what would be used for marking (or dropping) standard TCP flows.

A. Ability to effectively detect network congestion as early as possible

Early detection of congestion assists NF-TCP in quickly yielding to standard TCP. The aim is to ensure that NF-TCP packets do not contribute to queue build-up/congestion that results in higher latencies for delay-sensitive traffic. Therefore

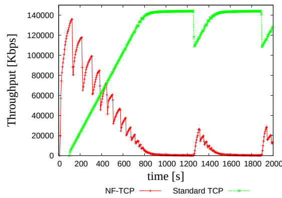

Fig. 1. Instantaneous throughput: one NF-TCP flow and one standard TCP flow. The standard TCP flow (in green) starts 90 seconds later.

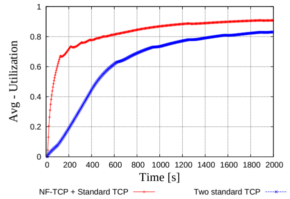

Fig. 2. Average utilization of bottleneck link: One NF-TCP flow coexisting with one TCP flow (red); two coexisting standard TCP flows (blue).

to detect the onset of congestion as early as reasonable, while ensuring that we do not respond to truly short-term transients, we set the marking threshold values lower than that for standard TCP using RED. Further, we use ECN as the feedback to the source, to enable a more rapid reaction to the onset of congestion. The queue thresholds are based on the same mechanisms as for a RED queue [4], except that the MinThreshold is set much lower and the MaxThreshold is set to about half of the buffer size. This mechanism ensures that the NF-TCP flow detects congestion at the very onset of queue build-up and backs off. The backing off results in yielding capacity to standard TCP flows. This modified RED queue behaves like a drop-tail queue for non-ECN enabled TCP flows. The receiver feeds back the received ECN marks via acknowledgements. Further study concerning the optimal choice of the various queue parameters such as MinThreshold, MaxThreshold, marking probability, etc. are ongoing.

B. Ability to saturate available bandwidth as fast as possible in the absence of other TCP flows

To optimize the network throughput and be opportunistic in using available bandwidth, NF-TCP is designed to be aggressive during non-congestion periods. Since NF-TCP is backed by an efficient congestion detection scheme and early dropping, it can afford to be aggressive as long as it is below the estimated available bandwidth.

The NF-TCP measures available bandwidth based on a probing mechanism similar to that of PathChirp [5] and RAPID [6]. The NF-TCP protocol is designed to dynamically adjust the measurement probe rate to be able to get an approximate measure of the available bandwidth as quickly as possible, while limiting the overhead it introduces in the network.

III. Simulation Results and Discussions

The simulation setup consists of a standard dumbbell scenario with a bottleneck link capacity of 150 Mbits/sec and an RTT of 100 ms. The router at the bottleneck link has a finite buffer capacity set equal to the Bandwidth Delay Product (BDP) of 2000 packets. FTP is the simulated application. The receiver generates a SACK for every received data packet. We assume that the receiver's advertised window is always large enough to ensure that the sending rate is equal to the congestion window size ($cwnd$). For convenience, the window size and the queue size are measured in terms of number of packets, and the packet size is 1000 bytes including the header size. The initial $ssthresh$ for Reno/SACK is set to 100 packets (100 Kbytes). Our ns-2 [7] implementation of NF-TCP is derived by modifying TCP Reno.

We evaluate the performance of NF-TCP coexisting with a standard TCP Reno/SACK flow in Figure 1. We can observe from Figure 1 that 1) the standard TCP flow obtains a higher share of the bottleneck link when co-existing with the NF-TCP flow compared to the case of co-existing with another TCP flow; 2) the NF-TCP flow with the support of bandwidth estimation is able to saturate the bottleneck bandwidth faster than TCP when no congestion is experienced; 3) the NF-TCP flow is backing off as soon as there is incipient congestion.

Figure 2 shows the corresponding overall utilization of the bottleneck bandwidth compared to the case of two standard TCPs coexisting. We can observe that the link is better utilized by a combination of NF-TCP and TCP traffic. This is due to aggressive nature of NF-TCP to use the available bandwidth as long as no ECN marks are received.

IV. Summary

We present the preliminary design of a network friendly TCP (NF-TCP), that allows delay-insensitive applications to be submissive to standard TCP flows during congestion periods. Differing from LEDBAT, NF-TCP is designed to be more aggressive in saturating the bandwidth during non-congestion periods by taking advantage of information obtained from an adaptive measurement of available bandwidth. However, it is conservative and robust in reacting to congestion by utilizing network-based congestion marking (ECN) as its core congestion control mechanism.

Our initial simulation results show the basic congestion control behaviour of NF-TCP. More experiments are being performed to evaluate its performance under various scenarios and study further potential improvements.

References

[1] S. Shalunov, "Low Extra Delay Background Transport (LEDBAT)," Internet Engineering Task Force, Internet-Draft draft-ietf-ledbat-congestion-00.txt, Oct. 2009, work in progress. [Online]. Available: http://tools.ietf.org/html/draft-ietf-ledbat-congestion

[2] A. Kuzmanovic and E. Knightly, "TCP-LP: a distributed algorithm for low priority data transfer," in *Proc. INFOCOM 2003*, vol. 3, 2003, pp. 1691–1701 vol.3.

[3] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168, Sep. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3168.txt

[4] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Jan. 1993.

[5] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "PathChirp: Efficient Available Bandwidth Estimation for Network Paths," in *Proc. Passive and Active Measurement Workshop*, 2003.

[6] V. Konda and J. Kaur, "RAPID: Shrinking the Congestion-Control Timescale," in *Proc. INFOCOM 2009*, 19–25 April 2009, pp. 1–9.

[7] "The network simulator – ns-2," http://www.isi.edu/nsnam/ns/.