ByPass: A Unified Transport Protocol for the Internet

Abhishek Chanda, Samuel C. Nelson, Gautam Bhanage, Dipankar Raychaudhuri
WINLAB, Rutgers University
achanda,snelson,gautamb,ray@winlab.rutgers.edu

1. INTRODUCTION

Over the years, the Internet has moved from being a "wired links only" system to a hybrid system with both wired and wireless links. TCP performs poorly over wireless links since it assumes all frame drops are due to congestion, this results in dropping the transmission window leading to degradation of throughput. Splitting a TCP connection enables local retransmissions and prevents throttling at the sender. Several new transport protocols have been proposed which breaks the "end-to-end" argument that TCP relies on. This alternate paradigm can be called the "hop-by-hop" paradigm where an "hop" can either be a physical hop (as in case of HOP [1]) or a logical hop (as in case of Split TCP [2]). Thus, we have two classes of transport protocols: protocols based on the "end-to-end" paradigm which perform well over wired links but not over wireless links and protocols based on the "hop-by-hop" paradigm perform well over wireless links but incur unnecessary overhead over wired links.

Problem formulation: These factors pose an unique challenge to an application developer, *how to select an optimal transport protocol for my application?*. This is a hard problem since it is impossible to determine the network topology and state while writing the application. Clearly, the protocol stack should select the optimal transport mechanism based on current network conditions. Then the application developer can always work with a network agonistic transport API. We argue that our proposed transport protocol called *ByPass* can swicth modes based on the current network condition and hence, can acheive optimal throughput and latency over all possible network topologies.

2. BYPASS FRAMEWORK

ByPass has two major features: (1) abstraction of the underlying transport mechanism from application developer (2) automatic adaptation of transport mechanism based on link conditions. Thus, *ByPass* is a self adjusting hop-by-hop transport protocol that selects hop end points dynamically based on factors like network conditions and path length. It selects hop end points on a packet-to-packet basis, allowing it to adjust rapidly to fluctuating network conditions.

Here, s is a metric which would determine whether the

if $currentnode == destination$ **then**
  $process\ packet\ and\ pass\ on\ to\ higher\ layers$
else
  $g \leftarrow cumulative\ bypass\ metric$
  $h \leftarrow number\ of\ hops\ bypassed\ so\ far$
  $nextHop \leftarrow get\ next\ hop\ from\ routing\ table$
  $s \leftarrow get\ stability\ of\ nextHop\ from\ routing\ table$
  $g \leftarrow g + f(s, h)$
  if $g \geq threshold$ **then**
    $end\ bypass,\ inform\ previous\ hops$
  else
    $initiate\ bypass\ request\ to\ nextHop$
  end if
end if

link to the next node is stable or not. $f(s, h)$ is a function that aggregates this metric with number of hops already bypassed to arrive at a decision metric g.

3. ONGOING WORK

Some major challenges which needs to be addressed are: (1) Explore ways to dynamically determine $threshold$ and study various choices for $f(s, h)$ (2) A mechnism to discover longest run of links which would minimize packet loss probability based on a link metric (s). How should such a service work? (3) How would *ByPass* handle factors like flow and congestion control? (4) How would it handle disconnections due to mobility? Our ongoing work aims to solve these challenges. Preliminary results indicate an improvement of throughput over existing schemes, however, we omit those results due to space constraint.

4. REFERENCES

[1] M. Li, D. Agrawal, D. Ganesan, and A. Venkataramani. Block-switched networks: a new paradigm for wireless transport. In *USENIX*, 2009.

[2] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi. Split tcp for mobile ad hoc networks. In *GLOBECOM*, 2002.