

MultiWAN: WAN Aggregation for Developing Regions

Kristin Stephens, Shaddi Hasan, and Yahel Ben-David

UC Berkeley
{ksteph, shaddi, yahel}@cs.berkeley.edu

1. INTRODUCTION

For rural ISPs and organizations, purchasing high-bandwidth, high-quality Internet connections is expensive, if such connections are even available. Subscribing to multiple low-capacity connections and load balancing flows across them presents an attractive alternative. However, this limits the burst rate of each flow to the capacity of the connection to which it has been bound. For example, a single file transfer could not take advantage of unused capacity on other connections. This increases flow completion times, an important metric of user-perceived performance.

Ideally, an organization could aggregate multiple WAN connections together to obtain the burst characteristics of a single high-bandwidth connection by load balancing *packets*, not flows. However, each WAN connection may have divergent link properties as a result of using different Internet routes and link technologies (*e.g.*, VSAT vs. DSL). The resulting variation in end-to-end delay leads to re-ordering of packets within a single flow, ruining TCP performance, which accounts for the bulk of Internet traffic.

2. OUR APPROACH

We propose MultiWAN, a system for packet-level load balancing to allow full utilization of multiple WAN connections without impacting TCP performance. MultiWAN emulates a single high-bandwidth link while performing no worse than flow-based load balancing. We focus on web traffic performance, which is typically asymmetric; thus, load balancing ingress traffic is as important as load balancing egress traffic.

MultiWAN consists of a pair of middleboxes, a *local endpoint* at the border gateway of a rural ISP or organization and a *remote endpoint* located at a place with a high-capacity WAN connection, such as a cloud provider's data center. We establish a VPN tunnel between the endpoints on each WAN connection, and use flowlet-switching [1] to load balance individual flows across these tunnels. To ensure downstream traffic is routed back through the tunnel, the remote endpoint performs NAT. MultiWAN requires no changes or configuration of end-hosts, and operates purely as a transparent middlebox. In addition, MultiWAN's tunnel architecture enables optimizations that are otherwise impractical, such as compression

and encryption between the local and remote MultiWAN endpoints.

MultiWAN must determine both the available bandwidth and latency of each WAN connection to allocate traffic appropriately. We achieve both in real time using in-band measurements by marking each packet sent through both MultiWAN endpoints with a timestamp. To determine whether congestion is occurring on the path used by a WAN connection, the receiving endpoint compares inter-arrival spacing with the gap between timestamps on each packet [2]. If the former is greater than the latter, MultiWAN assumes that WAN connection is congested, and shifts traffic off of it.

To calculate the difference in one-way-latency between each WAN connection without requiring synchronized clocks we also use per-packet timestamps. By taking the difference of a packet's timestamp and arrival time, we obtain the one-way-latency of a WAN connection's Internet path plus any drift in the sender's and receiver's clocks. Taking the difference of these values for two connections gives us the difference in one-way-latency between those two connections and eliminates clock drift between the two endpoints.

These measurements are reported from the receiving endpoint to the sending endpoint, and new flowlets are allocated preferentially to uncongested WAN connections. The latency information is used to group WAN connections into *latency classes*. Flowlet switching across paths with large latency differences yields poor burst performance since the minimum size of flowlets must be at least the difference in latency between paths, so MultiWAN performs flowlet switching within a latency class, and flow-level switching among latency classes. New flows are allocated to the lowest latency class, and flows are bumped to higher latency classes as the low latency class becomes congested. Flows move between latency classes in order of age, so long-lived flows are eventually placed in the higher latency classes, allowing short flows, which account for the bulk of web traffic, to use the low latency links and achieve shorter flow completion times.

We have implemented MultiWAN in our network emulation testbed and are evaluating its performance with simulated traffic derived from a trace from a rural Indian ISP. Initial results suggest MultiWAN achieves our design goals and adapts appropriately to changing network conditions. We hope to continue evaluating MultiWAN using a remote endpoint located in a public cloud environment and deploy the system with real users in the developing world.

3. REFERENCES

- [1] S. Kandula, D. Katabi, S. Sinha, and A. Berger. Dynamic load balancing without packet reordering. *SIGCOMM Comput. Commun. Rev.*, 37:51–62, March 2007.
- [2] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC '03, pages 39–44, New York, NY, USA, 2003. ACM.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DEV'12, March 11–12, 2012, Atlanta, GA.

Copyright 2012 ACM 978-1-4503-1262-2/12/03 ...\$10.00.