A Multipath Flow Routing Approach for Increasing Throughput in the Internet

Brian L. Mark and Shidong Zhang
Dept. of Electrical and Computer Engineering
George Mason University
Fairfax, VA 22030

***Abstract*—In the current Internet, hotspots arise because traffic tends to be routed along shortest hop paths sharing common links, leading to congestion on these links and underutilization on others. Current routing protocols are not able to exploit the path redundancy that exists in the Internet. To improve overall throughput and network utilization, we propose a multipath flow routing overlay system, coupled with network sensors that provide real-time information about overlay links, to distribute traffic more efficiently over the network. In particular, we focus on improving the throughput performance of large file transfers over TCP/IP-based internets. Files are dynamically split into multiple flows, which are then distributed over the network by the multipath flow routing overlay. We have implemented such a routing overlay using the Click modular software router and studied its performance on the Emulab network emulation testbed. We present performance results, which show that the multipath routing overlay can dramatically increase TCP throughput under a variety of network scenarios.**[1]

***Index Terms*— Internet, TCP/IP, Routing, Multipath Routing, Network Performance, Congestion Control, Network Sensing.**

I. Introduction

Congestion hotspots arise in the Internet in part due to the inability of the routing protocols to exploit path redundancy in the network. The current Internet routing protocols generally route packets between a given pair of source and destination endpoints along the shortest hop path. Depending on the traffic load, congestion can arise on the common links shared by various shortest path routes, while other links which could be used to alleviate the congestion are underutilized.

In multipath routing, routers maintain a multipath set of two or more paths to each destination subnet. In packet striping, packets are distributed in a (weighted) round-robin sequence over the paths in the multipath set. This approach works well when the paths are nearly equal in terms of bandwidth, latency, and packet loss rate. When the paths have asymmetric impairments, TCP throughput tends to be severely degraded due to out-of-order packet delivery at the receiver. Packet-level striping is an option in the ECMP (Equal Cost MultiPath) protocol [1] and is used in multilink PPP and IMA (Inverse Multiplexing for ATM).

The OMP (Optimized MultiPath) protocol [2] distributes traffic over a multipath set using a hash function (e.g., CRC-16) applied to the source and destination addresses in the packet header. The hash space is partitioned into subspaces corresponding to the paths in the multipath set. The hash-based approach does not suffer from packet reordering, but provides limited control over the distribution of traffic.

In this paper, we propose a multipath flow routing approach to improve network throughput, specifically for TCP-based file transfers. Our approach consists of two main components: 1) a multipath flow routing overlay and 2) an application-layer program to split a file into multiple flows. The multipath flow routing overlay is based on a software flow router called SAFIRE.[2] The routing overlay consists of SAFIRE nodes distributed in the network and interconnected by UDP tunnels. SAFIRE performs flow routing [3], in the sense that it identifies individual TCP flows, and distributes the flows over a multipath set according to a multipath traffic distribution policy based on real-time overlay link metrics provided by a network sensing infrastructure called the CHART Sensing Infrastructure (CSI) [4]. The file-splitting program dynamically partitions a given file into smaller pieces and transfers the pieces using multiple flows. The routing overlay then assigns the flows to the paths in a multipath set to maximize the file transfer throughput.

We evaluate the performance of the proposed multipath flow routing framework empirically on the Emulab [5] network emulation testbed using a simple network topology over a range of transmission path characteristics. We also compare the throughput performance of multipath flow routing with packet striping. We conclude that multipath flow routing provides nearly optimum TCP throughput performance over a given multipath set.

The remainder of the paper is organized as follows. Section II describes the multipath flow routing overlay in more detail. Section III discusses how file splitting is performed. In Section IV, the Emulab testbed environment and the network topology used to perform the throughput experiments are described. Section V provides a performance evaluation of the multipath flow routing approach in terms of TCP throughput. Finally, the paper is concluded in Section VI.

[1]This work was supported in part by DARPA Contract N66001-05-9-8904 (Internet Control Plane) and in part by the U.S. National Science Foundation under Grant No. ACI-0133390.

[2]The name *SAFIRE* is derived from Software Adaptive Flow-Intelligent RoutEr.

II. Multipath Flow Routing Overlay

The multipath flow routing overlay is formed as an overlay of SAFIRE nodes in a network, together with the CSI network sensing infrastructure.

A. SAFIRE Data Plane

SAFIRE implements flow state-aware routing in that packets are routed on the basis of per-flow state [3]. The data plane of SAFIRE consists of a multipath IP routing table, a flow table, and a path table. The routing table is similar to a conventional IP lookup table, except that each destination subnet entry contains pointers to multiple path pointers, up to a certain maximum number. An incoming packet destined to a particular subnet may have the possibility of being forwarded to one of several paths, each of which provides a distinct path to the same destination. A path pointer is an index into the path table. A given entry in the path table contains the following fields: outgoing interface, number of active flows, capacity, latency, and packet loss rate.

The flow table is indexed by a fixed length flow identifier, which is computed as a hash of five fields extracted from the packet header: source IP address, destination IP address, source port number, destination port number, and type. Each flow table entry contains two fields: a next hop pointer (denoted as nhp) and an activity counter. The nhp points to one of the path pointers stored in the associated IP routing table entry. In general, the IP routing table entry corresponds to the longest prefix match for the flow's destination IP address.

When a packet arrives to SAFIRE, its associated flow ID is computed as a hash of the five-tuple extracted from the packet header, as mentioned above. If the flow table entry corresponding to the flow ID is *active*, as indicated by the value of the activity counter, the packet is sent on the outgoing interface determined by the nhp. As discussed above, the nhp points to one of the path pointers associated with the flow's destination subnet in the IP routing table. This path pointer in turn points to the path table, which indicates the outgoing interface for the packet. Also, the activity counter associated with the flow is incremented by one.

If the flow table entry is *inactive*, the entry is rendered active by setting the activity counter to a predefined positive value. A full IP lookup is performed on the IP destination address contained in the packet header to find the appropriate routing table entry. The nhp is then set to point to one of the path pointers associated with this routing table entry. The associated path table entry is updated by incrementing the number of active flows by one. Finally, the packet is sent on the outgoing interface indicated by the path table entry.

B. Multipath Flow Distribution

The control plane of SAFIRE consists of a multipath flow distribution system and an interface to the CSI network sensing infrastructure. Flows are distributed over the paths in a multipath set based on an estimate of the per-flow TCP throughput that can be provided on each path. The approximate TCP throughput for a given path p, denoted as α_p, is estimated as a function of the path capacity, latency, and packet loss rate, which are derived from CSI updates and stored in the path table. When a new flow arrives, its nhp is set to point to the path pointer field of the associated IP routing table entry corresponding to the path p that offers the largest per-flow TCP throughput $\alpha_p/(N_p+1)$, where N_p represents the current number of active flows on path p.

TCP throughput along a given path depends on a number of parameters, including the path capacity, latency, and packet loss rate. In addition to these parameters, TCP throughput depends on the nature of cross-traffic that traverses part or all of the given path. In this paper, we shall assume that there is no cross-traffic. Cross-traffic can taken into account using a bandwidth probe mechanism, but this is beyond the scope of the present paper. For multipath flow distribution, we make use of the following simple approximation for TCP throughput: (cf. [6, 7]):

$$\alpha \approx \min\left\{C, k\frac{\mathrm{MSS}}{\mathrm{RTT}\sqrt{\varepsilon}}\right\}, \tag{1}$$

where C is the path capacity, MSS is the maximum segment size, RTT is the round-trip time, ε is the packet loss rate, and k is a constant of proportionality. Empirical measurements discussed in [7] suggest that setting $k = 1$ provides an approximate upper bound to the throughput of TCP. More accurate estimates of TCP throughput could be obtained using more sophisticated approximations or by tabulating empirical measurements.

C. Network Sensing Infrastructure

The CSI service [4] provides real-time link status information to SAFIRE nodes. CSI is derived from the Scalable Sensing Service (S^3) discussed in [8] and can provide estimates of capacity, bandwidth, latency, and packet loss rate between any two overlay nodes running the service. We assume that CSI is instantiated at each SAFIRE node in the multipath routing overlay. The SAFIRE control plane makes requests to CSI for the capacity, latency, and packet loss rate estimates associated with each of its attached overlay links. The SAFIRE path tables are updated using link status information from CSI.

A routing protocol integrated with CSI should distribute real-time CSI information to the SAFIRE nodes in the overlay. Such a routing protocol is beyond the scope of the present paper, but is necessary to scale the routing overlay to networks of practical size. In the current implementation of the routing overlay, SAFIRE nodes make explicit requests to CSI for link status information associated with paths represented in the routing table.

III. Multiflow File Splitting

In recent years, applications called *download managers* have become popular in the Internet. A typical download manager provides a number of features for accelerating and managing file downloads using various application-layer protocols, primarily *http* and *ftp*. Two popular and freely available download managers are *aria2* [9], primarily for Linux-based systems, and *FlashGet* [10], for Windows-based systems.

In this paper, the feature of most interest is that of downloading a file using multiple, simultaneous TCP flows. Most modern ftp and http servers provide the capability of initiating

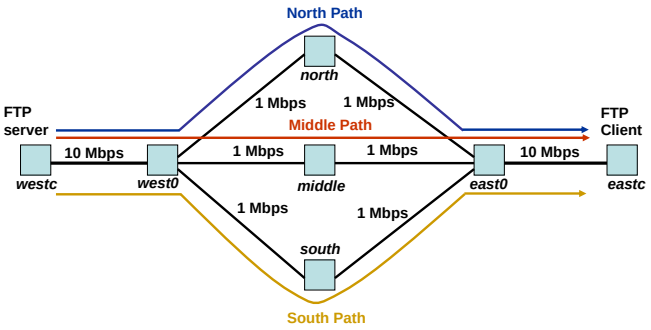

Fig. 1. Multiflow ftp download over multipath flow routing overlay.

a TCP flow to download a portion of a file between arbitrary start and end points. A download manager running on the client side uses multi-threading to initiate multiple, simultaneous TCP flows to download different parts of a file using different flows. However, the full benefit of using multiple flows cannot be exploited by conventional routers because all of the flows will be routed along the same path.

The main idea of this paper is to combine multiflow file splitting with multipath flow routing to maximize TCP throughput by exploiting path redundancy. Fig. 1 illustrates an example scenario of a multiflow/multipath file download from an ftp server *westc* to an ftp client *eastc* over a multipath routing overlay consisting of SAFIRE nodes labelled as *west0*, *north*, *middle*, *south*, and *east0*. The download manager initiates three flows, which are routed by the ingress node *west0* over the north, middle, and south paths, respectively, leading to the egress node *east0*. Assuming that the three paths between *west0* and *east0* are disjoint at the physical layer, the potential throughput gain from multipath flow routing is three times.

For comparison, we have also implemented packet striping on SAFIRE as an alternative multipath solution. Fig. 2 depicts a single-flow file transfer using packet striping over the multipath set between *west0* and *east0*. Packet striping of the TCP flow is performed by *west0*, which sends one packet on the north path, the next packet on the middle path, and the next on the south path, and then returns to the north path in round-robin fashion. When the path capacities are asymmetric, packet striping is performed using weighted round-robin, where the weights are chosen to be proportional to the path capacities.

IV. Testbed Environment

The multipath flow routing overlay was implemented on the Emulab network emulation platform [5] according to the topologies shown in Fig. 1 and 2. The server *westc* runs an ftp server over Linux, while the client *eastc* runs either FlashGet for multipath flow routing or the regular ftp command over Windows.

In the multipath flow routing scenario, FlashGet initiates a three-flow file transfer. The ingress SAFIRE, *west0*, distributes the three flows over the three paths leading to the egress, *east0*. In the packet striping scenario, *west0* transmits packets over the three paths using a weighted round-robin scheme, where the weights are proportional to the path capacities.

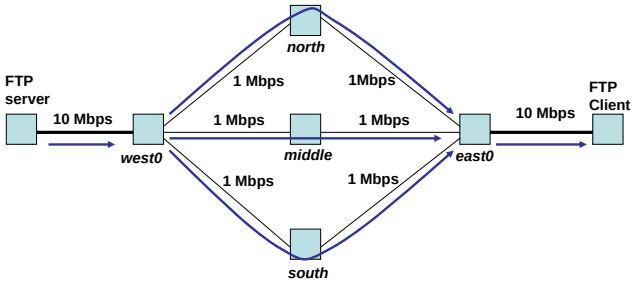

Fig. 2. Multipath ftp download using packet striping of a single flow.

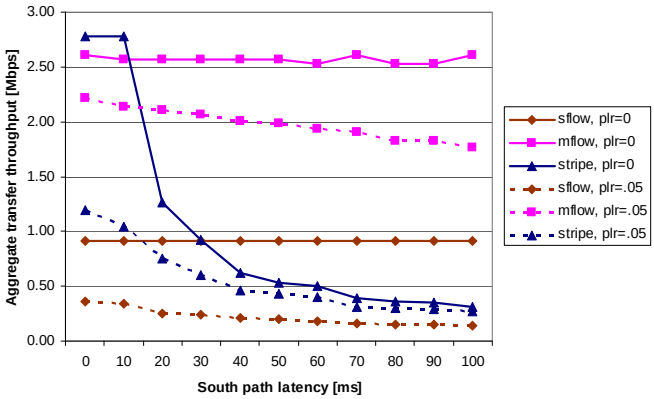

Fig. 3. TCP throughput vs. latency: south path capacity = 1 Mbps.

The SAFIRE data plane was implemented by extending the Click Modular Router platform [11] to perform multipath flow routing, as well as packet striping. Click provides an extensible user-level software routing platform for the data plane of an IP router. To perform adaptive multipath flow routing, the control plane interfaces with the CSI network sensing infrastructure to obtain link status information.

V. Performance Evaluation

Figs. 3-6 show the TCP throughput performance of single path flow routing on the south path, multipath flow routing, and packet striping, for various network scenarios run on Emulab.

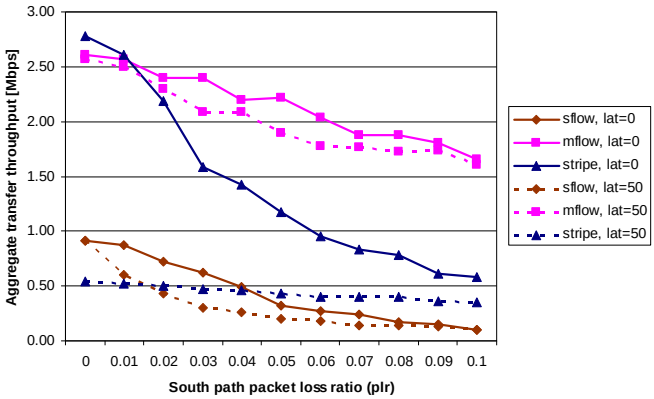

Fig. 4. TCP throughput vs. packet loss rate: south path capacity = 1 Mbps.

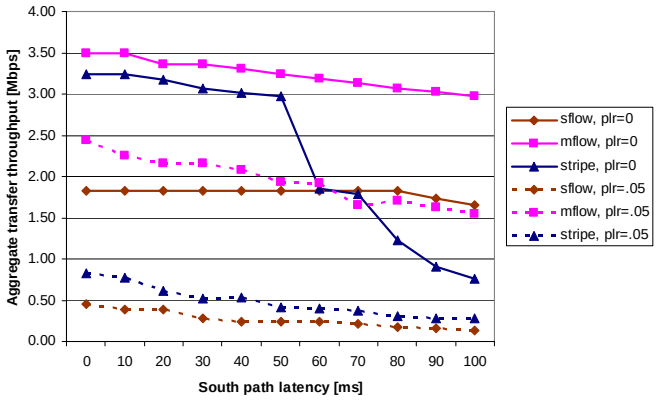

Fig. 5. TCP throughput vs. latency: south path capacity = 2 Mbps.

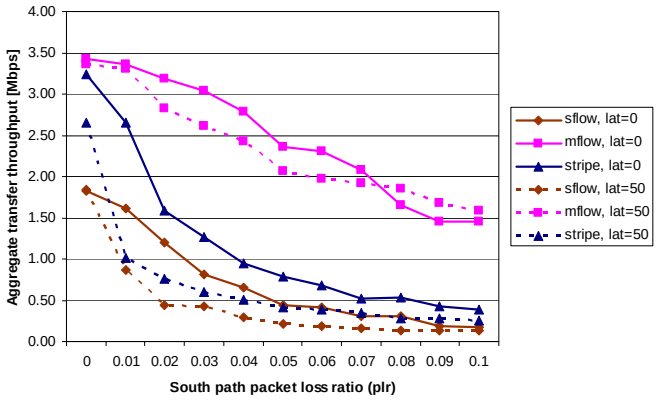

Fig. 6. TCP throughput vs. packet loss rate: south path capacity = 2 Mbps.

In all cases, the path metrics of the north and middle paths are set as follows: capacity = 1 Mbps, latency = 0, and packet loss rate (plr) = 0. In Figs. 3 and 4, the capacity of the south path is set to 1 Mbps, such that the aggregate capacity of the multipath set is 3 Mbps. In Figs. 5 and 6, the capacity of the south path is set to 2 Mbps, such that the multipath capacity is 4 Mbps.

Fig. 3 shows the aggregate throughput for the three transfer approaches vs. the south path latency when the packet loss rate is zero (solid curves) and 0.05 (dotted curves). Packet striping at zero south path plr nearly achieves the multipath capacity of 3 Mbps when the south path latency is less than 10 ms, but the throughput drops dramatically as the south path latency is increased from 10 to 20 ms, due to the increase in out-of-order packet deliveries to the receiver. Beyond a latency of 30 ms, packet striping performs worse than the single flow transfer (cf. [12]). By contrast, multipath flow routing (mflow) appears to be insensitive to the latency on the south path when the plr is zero. The throughput achieved by multipath flow routing is slightly less than that of packet striping when the south path latency is zero, due to the overhead involved in initiating multiple TCP flows. Note that when the south path plr is zero, single path flow routing (sflow) nearly achieves the south path capacity of 1 Mbps, irrespective of the latency.

Fig. 4 shows the aggregate throughput vs. the south path packet loss rate (plr) for the three approaches at zero latency (solid curves) and at a latency of 50 ms (dotted curves) on the south path. Observe that the two mflow curves are close to each other, which suggests that multipath flow routing is relatively insensitive to the south path latency. We also point out that when the south path plr increases beyond 0.1, the south path becomes practically unusable, yet multipath flow routing is still able to fully utilize the capacity of the north and middle paths. Packet striping at zero latency degrades relatively quickly as a function of the south path plr. It is interesting to note that when the south path latency is set to 50 ms, the throughput performance of packet striping is worse than that of single flow routing for small south path plr values and then becomes slightly better for larger values. This is because the overall plr for packet striping becomes less than that of single flow routing after a certain point.

Figs. 5 and 6 are analogous to Figs. 3 and 4, except that the capacity of the south path is doubled to 2 Mbps, so that the total multipath capacity becomes 4 Mbps. In Fig. 5, it is interesting to note that the performance of packet striping at zero south path plr becomes worse than that of single path routing as the south path latency is increased beyond 60 ms. When the south path plr is 0.05, packet striping performs slightly better than single path routing. Fig. 6 shows that the performance of striping degrades more quickly than that of multipath flow routing when the path capacities are asymmetric.

VI. Conclusion

Multipath flow routing, combined with multiflow file splitting can significantly increase TCP throughput by fully exploiting path redundancy in the network. Nodes in the routing overlay use real-time link status information from a network sensing infrastructure to control the distribution of flows onto a multipath set associated with a given destination subnet. Since multipath flow routing is implemented as an overlay, it can readily be deployed in the Internet.

References

[1] J. Moy, "OSPF version 2." IETF Internet RFC 2328, 1998.
[2] C. Villamizar, "OSPF Optimized MultiPath." IETF Internet draft, draft-ietf-ospf-omp-02, Feb. 1999.
[3] L. G. Roberts, "The Next Generation of IP - Flow Routing," in *Proc. SSGRR 2003S International Conference*, (L'Aquila Italy), July 2003.
[4] A. Bavier *et al.*, "Increasing TCP Throughput with an Enhanced Internet Control Plane," in *Proc. IEEE Military Comm. Conf. (Milcom '06)*, (Washington DC), Oct. 2006.
[5] B. White *et al.*, "An Integrated Experimental Environment for Distributed Systems and Networks," in *OSDI02*, (Boston, MA), pp. 255–270, USENIX Assoc., Dec. 2002.
[6] S. Floyd, "Connections with Multiple Congested Gateways in Packet-Switched Networks, Part 1: One-way Traffic," *Computer Communications Review*, vol. 21, pp. 263–297, Oct. 21991.
[7] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Computer Communication Review*, vol. 27, July 1997.
[8] P. Yalagandula, P. Sharma, S. Banerjee, S.-J. Lee, and S. Basu, "S^3: A Scalable Sensing Service for Monitoring Large Networked Systems," in *Proc. Workshop on Internet Network Measurement 2006*, (Pisa, Italy), Sept. 2006.
[9] http://aria2.sourceforge.net.
[10] http://www.flashget.com.
[11] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM Trans. on Computer Systems*, vol. 19, pp. 263–297, Aug. 2000.
[12] Y. He and J. Brassil, "NATALIE: An Adaptive, Network-Aware Traffic Equalizer," in *Proc. IEEE Int. Conf. on Comm. (ICC '07)*, June 2007.