Reducing Overhead in Flow-Switched Networks: An Empirical Study of Web Traffic

Anja Feldmann, Jennifer Rexford, and Ramón Cáceres
AT&T Labs – Research, Florham Park, NJ
{anja,jrex,ramon}@research.att.com

Abstract— **To efficiently transfer large amounts of diverse traffic over high-speed links, modern integrated networks require more efficient packet-switching techniques that can capitalize on recent advances in switch hardware. Several promising approaches attempt to improve performance by creating dedicated "shortcut" connections for long-lived traffic flows, at the expense of the network overhead for establishing and maintaining these shortcuts. The network can balance these cost-performance tradeoffs through three tunable parameters: the granularity of flow end-point addresses, the timeout for grouping related packets into flows, and the trigger for migrating a long-lived flow to a shortcut connection.**

Drawing on a continuous one-week trace of Internet traffic, we evaluate the processor and switch overheads for transferring HTTP server traffic through a flow-switched network. In contrast to previous work, we focus on the full probability distributions of flow sizes and cost-performance metrics to highlight the subtle influence of the HTTP protocol and user behavior on the performance of flow switching. We find that moderate levels of aggregation and triggering yield significant reductions in overhead with a negligible reduction in performance. The traffic characterization results further suggest schemes for limiting the shortcut setup rate and the number of simultaneous shortcuts by temporarily delaying the creation of shortcuts during peak load, and by aggregating related packets that share a portion of their routes through the network.

Keywords— **Traffic characterization, IP flows, HTTP protocol, switching, signaling, routing.**

I. Introduction

The explosive growth of Internet traffic adds urgency to the search for more efficient packet-switching techniques. To exploit recent advances in high-speed switch hardware, several proposals call for grouping sequences of related IP packets into *flows*, and sending them through fast switching paths, or *shortcuts*, through the underlying network fabric [1–6]. Shortcuts improve the performance experienced by network traffic but consume extra network resources to create and maintain. In this paper we explore the tradeoffs inherent in using different definitions for what constitutes a flow and different criteria for creating and maintaining a shortcut connection. We look at ways to reduce overhead in the network while still giving a majority of traffic the benefits of following a shortcut. The remaining packets are routed and switched along a default path.

We consider three parameters that determine flow and shortcut decisions: *aggregation*, *timeout*, and *trigger*. Aggregation refers to the addressing level at which traffic is combined to form a flow. For example, a flow may be defined to contain all traffic flowing between two IP hosts (i.e., host-to-host flows). Alternatively, a flow may contain only traffic flowing between two application processes running in two IP hosts (i.e., port-to-port flows). Timeout refers to how long a flow is idle before it is considered closed. For example, a flow may be defined as closed if no traffic at the chosen aggregation level has been detected in the previous 60 seconds. Trigger refers to how much traffic appears in a flow before a shortcut is established for that flow. For example, a shortcut may be established only after 10 packets that obey a flow definition have been detected. The selection of these three parameters directly affects the ability of the network to detect and shortcut long-lived traffic.

We explore the effects of varying these parameters on three metrics of interest: *the percentage of traffic that follows shortcuts*, *the shortcut setup rate*, and *the number of simultaneous shortcuts*. The percentage of total traffic that follows shortcuts is a measure of the performance gains achieved by using shortcuts. The higher this metric, the higher the performance. In contrast, the shortcut setup rate and the number of simultaneous shortcuts are measures of network overhead resulting from using shortcuts. The higher these two metrics, the higher the overhead. In contrast to previous studies, we characterize the *distribution* of these metrics to study the network load on a variety of time scales, in order to develop an understanding of how the traffic dynamics affect network overhead.

We base our study on temporal and spatial characteristics of Internet traffic. There is a growing body of work in measurement-based traffic characterization and its application to network design [6–16]. Our effort extends previous work in several ways. First, our traffic traces reflect the dominance of World Wide Web traffic in today's Internet. We focus on Web traffic and take a network-centric view, specifically not focusing on either client or server traffic. Second, we use long, continuous traffic traces. Working with a full week of data allows us to study the effects of looking at traffic on different time scales such as minutes, hours, and days. Finally, we consider the interaction of all three flow and shortcut parameters on a variety of time scales. The following summarizes our main observations and their implications:

- *Variability in traffic load changes with time scale.* For example, there are large variations in load in the 10- to 100-second time frame, but there are many periods of consistent load when viewed from a 1- to 2-hour time frame. This permits a network to make relatively long-term resource allocation decisions without tracking short-term variations.

- *Probability distributions of Web flow sizes have several modes.* For instance, there are many extremely short Web flows due to failed requests and cache validation messages. The network can set its triggers high enough to avoid the overhead of establishing separate shortcuts for these flows.
- *Aggregation and triggers both reduce overhead, but their effects are not additive.* Aggregation results in longer-lived flows, which negates some of the impact of triggers. On the other hand, aggregating traffic may allow a network to use larger triggers and still carry most traffic on shortcuts.
- *Aggregating consecutive and concurrent transfers from the same Web server yields substantial benefits.* In addition to lowering overhead, aggregation also reduces the unfairness that can result when a single client establishes multiple TCP sessions to the same server. These effects also preview some of the benefits of the persistent TCP connections called for by HTTP 1.1.
- *Aggregating traffic along portions of the route between the source and destination yields additional reductions in network overhead.* By combining traffic from server replicas and other nearby sites into a single flow, the network can reduce shortcut overheads without having to aggregate traffic from unrelated users.

The remainder of the paper is structured as follows. Section II describes our traffic measurement environment and an initial characterization of the packet-level trace data. In Section III, we focus on the probability distribution of flow sizes, in terms of the number of bytes and packets in a flow. These results highlight the unique flow dynamics in Web response traffic and the benefits of aggregating traffic at the host level. While the discussion in Section III is relevant to a variety of flow-switching schemes, Section IV focuses specifically on the network overhead for creating end-to-end shortcuts for long-lived flows. Finally, Section V looks at the potential benefits of integrating these shortcutting schemes with routing. We propose a flow definition that permits traffic aggregation along a subset of the route through the network, to reduce the setup rate and number of simultaneous shortcuts. Finally, Section VI concludes the paper with a discussion of future research directions.

II. Packet Trace Collection

The IP flow characterization draws on a continuous, one-week packet-level trace of Internet traffic. The trace contains diverse traffic to and from a mixture of end-point machines, including office personal computers, shared UNIX machines, proxy servers, and modem-connected hosts.

A. Trace Collection

For a realistic study of IP flow characteristics, we have collected extensive packet-level traces of the traffic on the T1 line that connects the AT&T Murray Hill location to the external Internet [17]. A single 10 megabit/second Ethernet segment carries all traffic to and from this T1 line, permitting efficient trace collection using `tcpdump` [18] on a personal computer operating in promiscuous mode. To collect long traces, each tcpdump output file contains header and timestamp information for 100,000 packets in a raw binary format. In the background, a Perl script compresses these output files and moves them to a multiprocessor compute server for more extensive post-processing. Using tcpdump, this machine reads each binary file to produce a condensed ASCII log that records the relevant information for each packet. Each entry includes a receive timestamp, along with the source and destination IP addresses and port numbers in the packet header.

Based on this information, an additional Perl script classifies packets into flows based on a timeout value, as well as a mask on the end-point IP addresses and port numbers. These masks can aggregate sessions with the same port, host, subnet, or net addresses. After applying the source and destination masks, the script can group packets with matching end-point addresses into a single flow, with the timeout value determining the maximum spacing between consecutive packets in the same flow. The script records starting time (the receive timestamp of the first packet), duration (based on the receive timestamps of the first and last packets), total number of packets, and total number of bytes (including the IP headers) for each flow. Then, various Splus functions process these log files to compute performance metrics at the flow level.

Most of the experiments in this paper evaluate a subset of the trace data to measure the flow statistics for specific types of traffic. For example, although the Ethernet segment mainly shuttles traffic to and from the external Internet, a small portion of the packets have both their source and destination IP addresses on the local Murray Hill subnet. This internal traffic accounts for approximately 7.5% of the bytes and 21% of the packets, mostly from short domain name service (DNS) messages. To focus our study on Internet flow characteristics, we have removed the internal traffic from the trace and have classified the remaining packets as incoming or outgoing packets, based on whether the source or the destination is an external IP address. The majority of the paper focuses on the incoming Internet traffic, which represents 76.1% of the bytes on the Ethernet segment, compared to 16.4% for outgoing traffic.

B. Traffic Characteristics

In this paper, we focus on a continuous one-week trace starting at 11am on Sunday February 16, 1997. The trace shows daily fluctuations, with the heaviest load during the work day, and smaller spikes in the evening hours; each weekday also shows a brief dip in network utilization during the lunch hour. Hourly averages of throughput remain below 0.6 megabits/second. However, much higher rates occur on a smaller time scale; for example, the Ethernet segment sustains throughputs of up to 3 megabits/second over one-minute intervals. Most of the incoming traffic stems from response messages from HTTP and FTP servers, as shown by the statistics in Table I. Web response messages on the well-known IP port 80 contribute over half of the incoming bytes. By applying the `tcpreduce` tool [19], we estimate that FTP-data transfers are responsible for over one-third of the transfers on “unknown” ports; the remaining traffic on unknown ports appear to stem mainly from

Protocol	Perc. bytes	Pkts/ flow	Bytes/ packet	Secs/ flow
http	52.33	16	612	5.3
smtp	3.28	22	251	6.0
dns	0.75	11	124	80.3
telnet	0.57	98	63	68.9
x11	0.26	118	66	249.3
ftp-cntrl	0.07	10	95	32.4
ntp	0.02	1	76	0.4
other	0.38	35	220	28.7
unknown	42.32	219	523	26.5

TABLE I

Incoming Internet traffic by protocol

real audio and other HTTP transfers.

Table I also includes the average flow sizes in packets and seconds, where the flow duration does not include the 60-second timeout value for these port-to-port flows. The table shows that short request-response protocols, such as the network time protocol, generate extremely short-lived flows. The relatively long DNS flows stem from communication between pairs of DNS servers, instead of isolated DNS responses to internal hosts. With the dramatic increase in the amount of Web traffic in the Internet, we focus the remainder of our study on HTTP flows. Despite the relatively low *average* size of Web flows, the heavy tail of the flow-size distributions and the ability to aggregate multiple transfers substantially reduce the overheads for shortcutting HTTP traffic, as shown in the next section.

III. Flow Characteristics for HTTP Responses

In this section, we investigate the dynamics of HTTP response traffic by characterizing the flow-size distributions as a function of traffic type, end-point aggregation, and timeout value. The results show that certain protocol features and user/application conventions have significant influence on the network's ability to establish efficient shortcut connections for Web response traffic.

A. *Unique Characteristics of Web Traffic*

To highlight the unique characteristics of HTTP traffic, Figure 1 graphs the distribution of flow sizes for incoming HTTP, Telnet, and SMTP packets. Each flow corresponds to packets with the same IP addresses and port numbers at both end-points and a 60-second timeout. The graph plots the density of the logarithm of the flow size; coupled with a logarithmic scale on the x-axis, plotting the density of the logarithm of the data facilitates direct comparisons between different parts of the graphs based on the area under the curve. By separating the flows by protocol, these plots highlight the unique application characteristics that generate the many peaks in Figure 1. Also, all three protocols have a fair number of short flows that typically correspond to failed service requests or occasional cases where consecutive packets arrive more than one minute apart.

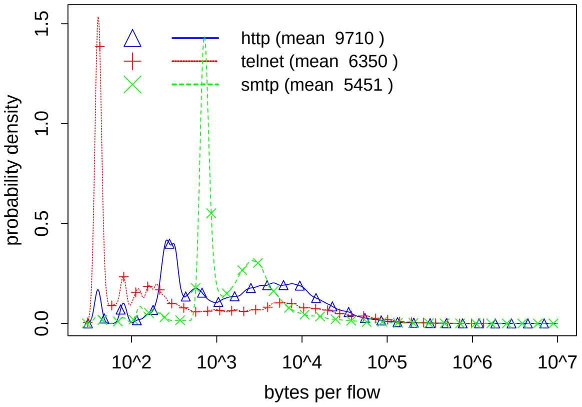


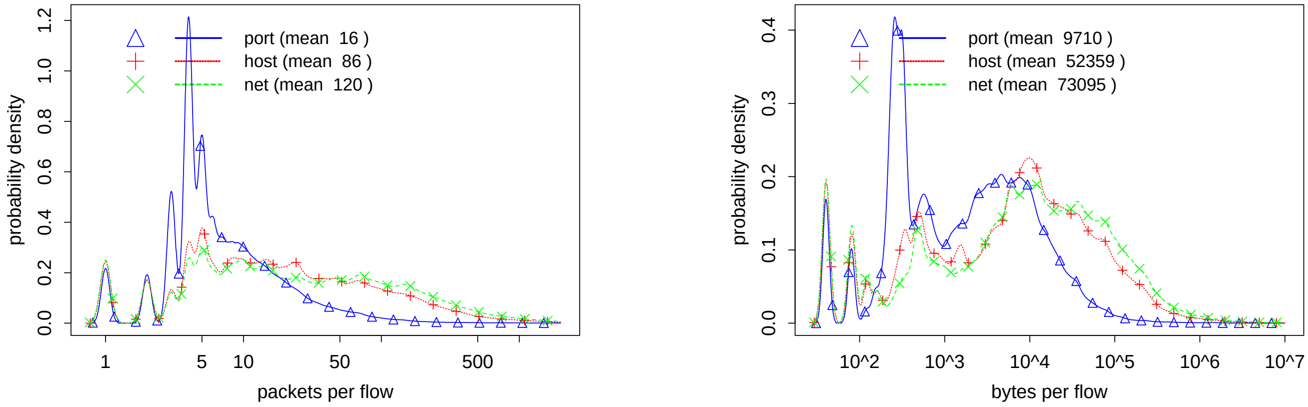
Fig. 1. Number of bytes in port-to-port flows

As shown in Table I, Telnet flows typically have the longest durations, even though 60-second periods of inactivity can split a single telnet session into multiple consecutive flows. Despite the relatively long duration of Telnet flows, these interactive flows do not generate as many bytes as the HTTP flows. SMTP traffic has a high concentration of flows with just under 1000 bytes. These flows stem from the minimum overhead for transmitting an e-mail message, while the remaining flows, centered at 3000 bytes, consist of longer messages. The HTTP responses include numerous flows with a large amount of data, as shown by the heavy tail in Figure 1; these results are consistent with recent characterizations of Web document sizes [20, 21].

With the relatively large timeout value, a port-to-port HTTP flow typically corresponds to a TCP session for a single transfer from a server to a client. The HTTP flow sizes in Figure 1 fall into three main categories. The first region consists of flows with less than 150 bytes, stemming from failed TCP sessions and HTTP error messages. In the second region, 24% of the flows have between 150 and 300 bytes. Many of these transfers involve small response messages that indicate that the client can use its cached copy of a Web page, although some small Web pages fall in the same size range. Finally, the remaining 69.5% of flows correspond to the actual transfer of Web data from the server to the client. Inspecting the HTTP response headers in the packet-level data verifies these trends: 72% of responses were "okay" retrievals, 18% were "not modified" messages, and 10% were various error messages (e.g., "no content," "moved temporarily," and "not found").

B. *Combining Multiple Web Responses*

To further characterize the dynamics of HTTP traffic, Figure 2 graphs the distribution of flow sizes for three different levels of end-point aggregation. At the lowest level, the port-to-port curves correspond to packets with the same IP addresses and port numbers at both end-points, effectively repeating the HTTP response results from Figure 1. By ignoring the port number at the two



(a) Length in packets

(b) Length in bytes

Fig. 2. Flow sizes for Web responses for different levels of aggregation

end-points, a single flow at the host-to-host level can include HTTP packets from different TCP sessions from the same Web server to the same Web client. Host-level aggregation flattens the peak in Figure 2(a) by combining consecutive transfers into a single flow. Since host-to-host aggregation decreases the total number of flows, the short single-packet flows represent a larger portion of the distribution, as seen by the slightly higher peaks at the far left in both Figure 2(a) and Figure 2(b).

For the larger flows, the host-to-host curve in Figure 2(b) closely resembles the port-to-port curve, except for a shift to the right by a factor of four. This can be partially attributed to the Web browser convention of opening four simultaneous TCP sessions for transmitting the inline contents within a Web page. Opening multiple TCP sessions allows the client to pipeline HTTP requests and increases throughput by circumventing the effects of TCP flow control. By aggregating traffic at the host-to-host level, the network can counteract the potential unfairness of this policy. In particular, the network could assign the same bandwidth to each HTTP shortcut connection. Enforcing this fair allocation, through traffic shaping or link scheduling in the network, can help ensure that aggressive clients cannot degrade the performance of other users by opening multiple TCP sessions to the server.

In addition to combining *concurrent* transfers at the Web server, host-to-host flows can capture *consecutive* accesses by the same client. With a 60-second timeout value, host-level aggregation can combine several transfers from the same Web server into a single flow. As more browsers and servers begin to use persistent TCP sessions that span multiple Web accesses [22], even port-to-port flows could include multiple Web transfers. In fact, the results in Figure 2(a) provide an initial indication of the potential performance benefits of this application-level aggregation. A larger flow timeout parameter increases the likelihood of capturing consecutive HTTP transfers between the same two endpoints. Experiments in Section IV with different timeout values can guide policies for when the network should close idle shortcut connection (or, similarly, when a server should close a persistent TCP session).

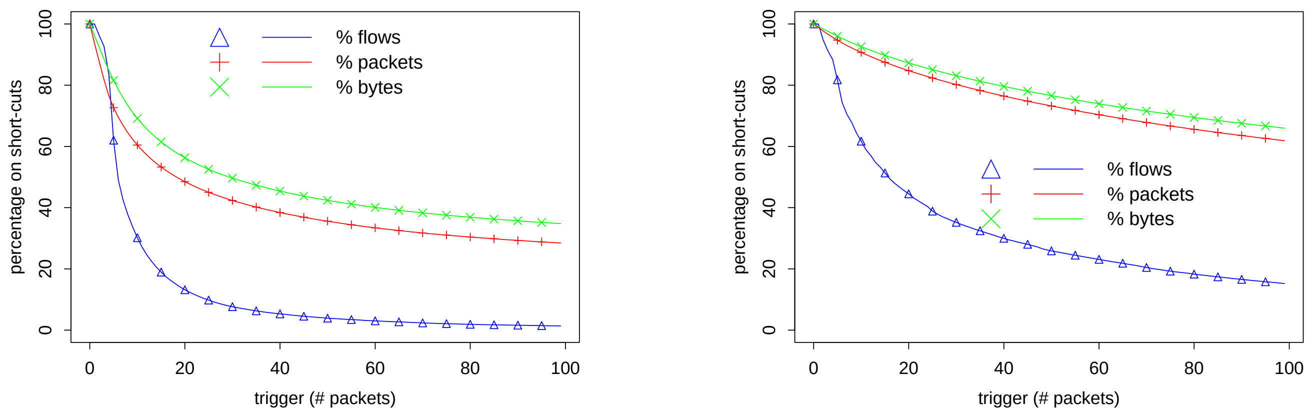
Aggregating traffic beyond the host level does not substantially change the flow length distributions, as shown in Figure 2. We define a *subnet* to include all hosts that share the first 24 bits of their IP addresses, and a *net* to include all hosts that share the first 16 bits of their IP addresses. Subnet aggregation combines multiple users that access the same Web server in a short span of time; similarly, subnet-to-subnet flows combine Web transfers from server replicas on a single local-area-network. However, our network traces do not exhibit this type of spatial locality on the time scale of the 60-second timeout. Likewise, even coarser aggregation at the net-to-net level does not have a significant impact on flow sizes, as shown in Figure 2. Instead, reducing the network overheads for transporting HTTP responses requires more aggressive techniques for detecting long-lived flows and for aggregating traffic over smaller portions of the route between the servers and the clients, as discussed in Section IV and Section V, respectively.

IV. Network Shortcut Overheads

The flow-size distributions give an initial indication of the network resources necessary to establish and maintain shortcut connections for long-lived flows. After a brief evaluation of how packet triggers affect the amount of shortcut traffic, this section studies how triggers, timeouts, and end-point aggregation affect the signaling and switching overheads in the network on a variety of timescales.

A. Proportion of Shortcut Traffic

After detecting a flow, based on the end-point addresses and a timeout value, the network must decide if and when to create a shortcut connection. Establishing a shortcut reduces the forwarding load on the default path, at the expense of signaling and maintaining a new connection.



(a) Port-to-port flows

(b) Host-to-host flows

Fig. 3. Percent shortcut traffic for different shortcut triggers

To quantify these tradeoffs, Figure 3 plots the amount of traffic that travels on a shortcut connection as a function of the triggering policy. The percentage of shortcut flows in Figure 3 corresponds to the cumulative distribution of flows that have at least x packets. The steep nature of the curve stems from the large number of short-lived flows in Figure 2(a). By waiting until a flow has transferred at least 10–20 packets, the network can avoid establishing shortcut connections for the large number of short-lived flows generated by control messages, cache hits, and small Web transfers. These basic results are consistent with the findings of previous Internet traffic studies [6–9].

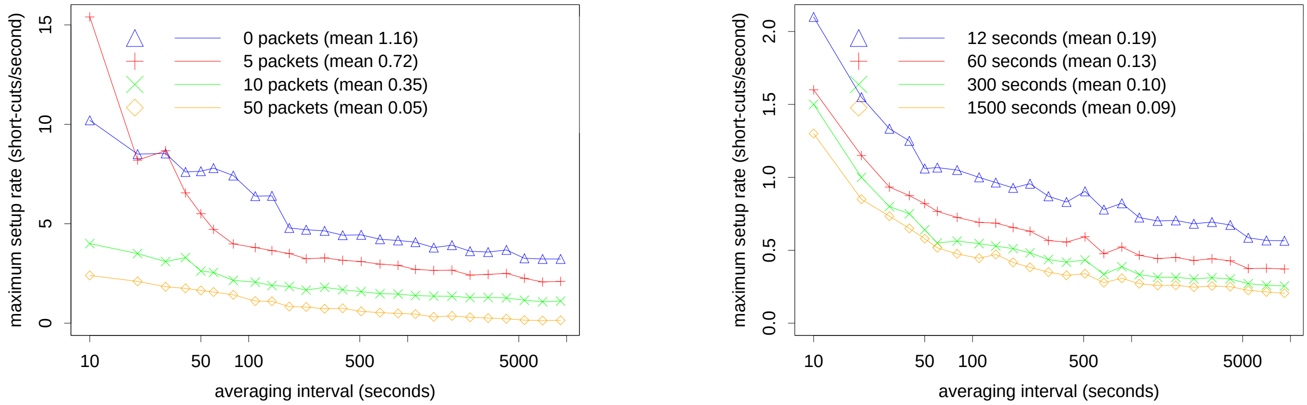
In addition to removing the large number of short-lived flows, the triggering policy also forces the first x packets of each long-lived flow to travel on the default path. Still, the heavy tail of the flow-size distributions in Figure 2 ensures that the network can forward a large proportion of the packets and bytes along shortcut connections. For example, in Figure 3(a), a 25-packet trigger filters over 90% of the port-to-port flows, while still permitting 45% of the packets and 53% of the bytes to travel along shortcuts. The slightly higher proportion of bytes stems from the large packets in long HTTP responses. Comparing Figure 3(a) and Figure 3(b), an x-packet trigger is somewhat less effective in reducing the proportion of shortcuts for host-to-host flows, since the coarser level of aggregation increases the number of packets in each flow. For example, applying a 25-packet trigger under host-level aggregation removes only 61% of the flows; as a result, shortcut connections carry 82% of the packets and 85% of the bytes.

B. Setup Rate for Shortcut Connections

The trends in Figure 3 translate directly into reductions in the setup rate for establishing shortcut connections. However, the setup rate can fluctuate dramatically across time, complicating the effort to provision signaling resources in the network. Experiments with the one-week packet trace show that the setup rate typically varies in proportion to the traffic load. To quantify these variations across time, Figure 4(a) plots the maximum shortcut setup rate over a variety of time scales, ranging from 10 seconds to nearly 3 hours. The graph plots the setup rate for four different packet triggers (0, 5, 10, and 50) for port-to-port flows with a 60-second timeout value. To generate the graph, we determine the number of shortcuts triggered in each interval, where the interval size is a multiple of 10 seconds. In general, the setup rate decreases as a function of the interval size, although small increases occasionally occur, due to the use of non-overlapping time intervals; a sliding-window computation could conceivably remove this minor effect, at the expense of a significant increase in computational complexity.

In Figure 4(a), the plot for a 0-packet trigger represents the overheads for establishing a shortcut for every flow; these baseline results also highlight the overheads for performing flow detection, even when the network employs a more conservative triggering policy. On a 10-second time scale, a 0-packet trigger introduces a maximum of 10.2 shortcuts/second to handle HTTP response traffic on a moderately-load T1 link. During peak day-time hours, the trace shows sustained *average* rates of 4–5 flows/second. Projecting to a higher-speed backbone network, the flow arrival rate could easily swamp the signaling resources in modern switches. Higher triggers and coarser flow aggregation can reduce this load. Compared to the port-level aggregation in Figure 4(a), host-to-host flows exhibit similar trends, with a factor of four smaller setup rates. To better characterize the burstiness of the flow-arrival process, we also measured the degree of self-similarity of one-hour segments of the trace using wavelet techniques [23]. Counting the number of flows that arrive within each 100-millisecond time period, we computed a Hurst parameter of 0.7 for port-to-port flows and 0.6 for host-to-host flows, suggesting that the flow-arrival process is indeed self-similar.

A combination of techniques may be necessary to reduce the signaling load to acceptable levels during times of peak



(a) Packet trigger (port-to-port flows)

(b) Timeout value (host-to-host flows)

Fig. 4. Shortcut set-up rate on different timescales with varying trigger and timeout values

network usage. To adjust to *short-term* load variations, the network can temporarily delay the establishment of shortcuts for flows, even if the packet trigger has been reached. For example, Figure 4(a) shows that a 10-packet trigger generates up to 4 shortcuts/second on a 10-second time scale, but this number drops to 2 shortcuts/second on a 100-second interval. Instead of provisioning for the higher load, the network could allocate processing resources for establishing 2 shortcuts per second. During brief periods of overload, the signaling processor can queue shortcut requests and continue to forward packets on the default path. The results in Figure 4(a) suggest that this would not substantially delay the creation of shortcut connections. Also, the processor can avoid establishing shortcuts for any flows that terminate in the meantime.

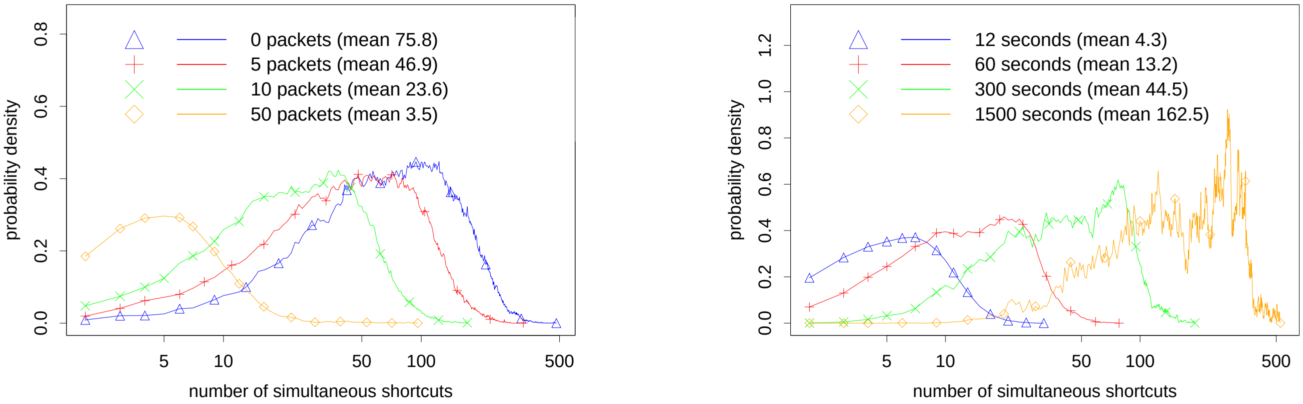
To adjust to *longer* periods of heavy load, the network may need to relax the flow definition to apply coarser endpoint aggregation and larger timeout values. Figure 4(b) experiments with different flow timeout values under host-level aggregation and a 10-packet trigger. Larger timeout values decrease the shortcut setup rate by capitalizing on the possibility of subsequent Web transfers between the same end-points. Compared to increasing the packet trigger, a larger timeout value can decrease the setup rate without forcing more packets to follow the default path. However, timeout values in excess of five minutes offer diminishing returns [9, 10], due to the decreasing likelihood of subsequent transfers from the same Web server.

C. Number of Simultaneous Shortcut Connections

In addition to affecting the signaling load, the timeout, trigger, and aggregation policies also influence the number of active shortcut connections across time. Depending on the underlying switching technology, the number of connections may impose a tighter restriction than the setup rate. Similar to the setup rate, the number of simultaneous shortcuts typically fluctuates in proportion to the link utilization. To focus directly on switch overhead, Figure 5(a) plots the probability density function of the number of shortcuts for four different trigger values, for port-to-port flows with a 60-second timeout. Moving to the right from a point on the x-axis, the area under the curve represents the proportion of time when the network must support at least x simultaneous connections. The legend indicates the average number of shortcut connections across the one-week trace, while the right side of each curve corresponds to the maximum value.

A 0-packet trigger introduces an average of 75.8 shortcuts, with sustained periods of more than 300 connections during peak day-time hours. In Figure 5(a), the curve for a 0-packet trigger serves as an upper bound for the other triggering policies, while also estimating the size of data structures necessary for the network to track active flows. A 10-packet trigger reduces the average number of shortcuts by a factor of 3.3 (from 75.8 to 23.6), while host-level aggregation reduces it by a factor of 3.8 (to 19.8); together, a 10-packet trigger and host-level aggregation reduce the average by a factor of 5.8 (to 13.2). Despite the large average and peak values for the 0-packet trigger, Figure 5(a) also shows a large portion of time with a fairly small number of connections. On the other hand, the absolute number of simultaneous shortcuts established by a particular policy will scale up with link speed and utilization, so that the number of shortcuts on a high-speed backbone link may exceed the capacity of current switches.

The above results illustrate that the ideal trigger, timeout, and aggregation options change with the time of day, suggesting the use of dynamic policies that vary across the day. For example, the one-week trace has three basic load periods corresponding to heavy load during the business day, moderate load on weekend days, and light load late at night, typical of a corporate networking environment; other parts of the Internet, such as residential access lines and backbone links, are likely to have different patterns. Although the network can support fine-grain flows with small trigger values during periods of lighter load, coarser



(a) Packet trigger (port-to-port flows)

(b) Timeout value (host-to-host flows)

Fig. 5. Distribution of the number of simultaneous shortcuts with varying trigger and timeout values

aggregation and larger triggers may be necessary for stable operation during heavy load. Alternatively, the network could reduce signaling load by employing a larger timeout value, at the expense of increasing the number of active shortcuts. Since the network does not close a shortcut connection until the timeout expires, a large timeout value can substantially increase the number of shortcut connections, as shown in Figure 5(b). The network can limit this effect by terminating inactive shortcuts as the available connection resources are depleted.

Depending on the network configuration, the establishment of shortcuts may be limited by either the number of connections or the signaling capacity of the switch, or both; in fact, the appropriate balance of timeout and triggering policies may vary with the time of day. Although timeouts, triggers, and end-point aggregation can substantially reduce network overhead, each technique offers diminishing returns beyond a certain point. In addition, overuse of these techniques can degrade application performance. For example, a large trigger forces more packets to travel on the default path, without enjoying the performance benefits of a shortcut connection. Also, aggregating traffic beyond the host level can introduce unfair interactions between users that must share a single shortcut. In the next section, we consider new techniques that reduce network overhead without requiring unrelated users to share a shortcut connection.

V. Traffic Aggregation Along Partial Routes

High-speed backbone networks may require more aggressive aggregation techniques to reduce signaling and switch overheads below the levels in Section IV. In this section, we propose a flow definition that generalizes the concept of end-point addresses to consider a *subset* of the path between the source and destination hosts. By drawing on this type of routing information, the shortcutting mechanism can group packets that have a common path through a portion of the network, such as the domain of a single service provider. A similar notion of partial routes has also been proposed in the Nimrod architecture to improve the scalability of routing computations [24].

A. Traffic Flows Along Partial Routes

The previous sections consider a flow model based on the IP address and port number at both the source and destination hosts. For the HTTP response traffic in our trace, the source addresses come from a diverse collection of Web servers that span across the Internet. As discussed in Section III-B, aggregating this HTTP response traffic beyond the host level does not substantially change the flow-size distributions, since the traces do not show significant subnet and net locality on the time scale of a 60-second timeout (see Figure 2). In a large network, a shortcut connection would typically carry traffic from just a single server to a single client (or small set of clients). Hence, whenever a client accesses a different Web server, the network would define a new flow and, ultimately, a new shortcut connection for the response traffic. Similarly, whenever a Web server initiates a response to a different client, the network would have to incur the overheads for establishing a new shortcut connection.

As an initial characterization of the diversity of routes, we performed traceroutes to each Web server in our one-week packet trace. To avoid mixing these UDP probes and ICMP replies with the traffic in the packet-level traces, the traceroutes were run just after the week of data collection completed. Although the traceroute results do not necessarily indicate the actual routes taken by the HTTP requests, or the reverse paths followed by the corresponding responses, they do give a basic indication of the fanout of traffic as it crosses the network. Also, recent studies indicate that a majority of routes are stable for periods of days or weeks [25]. The traceroutes from Murray Hill show 26 different "first hops" toward the 23, 060 different server IP addresses in the one-week trace. As hop-count increases, these numbers grow in an exponential sequence;

Address Aggregation	Flow Timeout	Set-Up Per Sec ∞	7	3	Num. Connections ∞	7	3
host-to-host	60 seconds	0.124	0.099	0.078	12.18	10.78	10.19
host-to-host	300 seconds	0.097	0.068	0.040	41.46	32.98	25.13
host-to-net	60 seconds	0.121	0.068	0.008	12.07	9.16	3.52
host-to-net	300 seconds	0.091	0.038	0.002	40.26	22.65	4.52

TABLE II

TRAFFIC AGGREGATION ALONG AN H-HOP PORTION OF THE ROUTE TO THE CLIENT

for example, the first seven hops have 26, 71, 137, 267, 406, 916, and 1508 different outcomes, respectively.

Based on these traceroute results, we can model the creation of shortcut connections across networks of different sizes. As an initial approach, we consider spheres of locality around the Murray Hill T1 link, based on the traceroutes to the server sites. To evaluate a three-hop region, for example, we assume that each of the 137 IP routers acts as an ingress switch for the HTTP response traffic en route to the T1 link. In this context, each flow originates at one of these routers (instead of using the actual source IP address of the Web server) and terminates at the client site. Although our traceroute experiments actually follow the forward path from the client to the server, the model still highlights the basic performance trends for the last h hops of the route from the server to the client. Similarly, the model can also project the overheads of establishing shortcut connections from a Web server (or set of servers) along the first h hops towards its client sites.

B. Partial-Route Aggregation

Drawing on this model of partial routes, we evaluated the overheads for 3-hop and 7-hop shortcut connections. The first row of Table II shows the results for host-to-host flows with a 60-second timeout and a 10-packet trigger, using the Perl scripts and Splus functions described in Section II (except that the h-hop router's IP address now acts as the "source" address of the flow). The mean set-up rate and number of shortcuts for *end-to-end* (∞) flows are slightly smaller than the average for a 60-second timeout in Figure 4(b) and Figure 5(b), since the experiments in Table II omit any flows that did not have a successful traceroute. Aggregating traffic along a portion of the route decreases both the setup rate and the number of simultaneous connections; 7-hop routes reduce the set-up rate and number of shortcuts 20% and 11% respectively, while 3-hop routes reduce them by 37% and 16%. In addition, route aggregation increases the proportion of traffic that travels on shortcuts, for the same trigger and timeout values. The 7-hop policy places 94.0% of the bytes on shortcut connections, compared to just 92.6% without route aggregation; 3-hop routes achieve an even higher proportion, with over 96.0% of the bytes traveling on shortcuts.

Partial-route aggregation reduces network overheads, while increasing the proportion of shortcut traffic, by combining concurrent and consecutive HTTP transfers when one or more Web servers communicate with the same client. To belong to the same flow, these transfers must travel through a common h-hop router on the timescale of the 60-second timeout. By focusing closely on specific flows in the trace, we find that partial-route aggregation combines transfers from replicas of the same Web site, as well as related servers at the same institution; quite often, these servers have the same net and subnet addresses. Aggregating across partial routes also combines transfers from different sites in the same web-hosting service, or other sites that happen to route through the same h-hop router. Compared to end-to-end flows based on the source and destination addresses, partial-route flows benefit more from larger timeout values, as shown by the second row in Table II. While a 60-second timeout may not be long enough to capture the user "think time" between consecutive Web accesses, a 300-second timeout with partial-route aggregation allows a single shortcut to aggregate transfers from different Web servers to the same client. For example, the larger timeout value reduces the average set-up rate by nearly a factor of 2 (from 0.078 to 0.040 per second) for the 3-hop aggregation policy.

Aggregating traffic along partial routes has an even larger benefit when a single shortcut connection can be shared by multiple Web clients. For example, a network may combine related users into a single flow end point, particularly when the Web clients belong to a single company or university. For the partial-route flow model and the Murray Hill trace, this effectively combines all of the users into one destination address, which receives traffic from a number of different h-hop ingress points. For the 7-hop configuration, with 1508 ingress routers, this reduces the average setup rate by 30% (from 0.099 to 0.068 shortcuts/second), beyond the results for host-to-host flows; the 3-hop setup rate experiences an even more dramatic reduction, falling by a factor of 9.8 (from 0.078 to 0.008). Similarly, the average number of simultaneous shortcuts falls by a factor of 15% and 65%, for the 7-hop and 3-hop routes respectively, compared to the results for host-to-host flows. A larger timeout value offers further reduction in the setup rate, at the expense of an increase in the number of simultaneous connections.

These initial results suggest that aggregating traffic along partial routes provides a desirable alternative to selecting coarse-grained end-to-end flows. Compared to flows between the source and destination end points, partial-route flows can reduce the shortcut setup rate without forcing unrelated clients to share the same shortcut. Since small network areas (lower h values) have lower overheads, large networks may benefit from dividing the network into multiple regions, with separate shortcut connections across each part of the route. Large networks are already likely to require at least a logical partitioning into different regions to support scalable addressing and routing for shortcut connections. Although partial-route flows may require packets from a single TCP session to traverse a sequence of shortcut connections that start and end inside the network, the reduction in the setup rate and number of connections in each region can play an important role in scaling to large network configurations.

VI. Conclusions and Future Work

Shortcutting of long-lived traffic flows offers an efficient way to capitalize on recent advances in high-speed switching hardware. Since the World Wide Web has a dominant influence on network dynamics in the modern Internet, we have performed a detailed characterization of HTTP response traffic to evaluate the basic cost-performance tradeoffs in flow switching. A comparison with other types of traffic highlights the unique influence of the HTTP protocol and user behavior on the flow-size distributions and the benefits of end-point aggregation. To characterize the network overhead for flow switching, we present the distribution of important metrics, including the percentage of traffic that follows shortcuts, the shortcut setup rate, and the number of simultaneous shortcuts, while varying the timeout, trigger, and aggregation policies. Finally, we evaluate new flow definitions that consider a subset of the path between the source and destination hosts.

Our results suggest several possible schemes for limiting the shortcut setup rate and number of simultaneous connections by temporarily delaying the creation of shortcuts during transient periods of heavy load. Since incoming packets can continue to follow the default path, the network has additional latitude to postpone establishment of shortcut connections. As part of future work, we plan to evaluate specific new policies that balance the short-term tradeoffs between processor and network load. These policies can extend existing schemes that delay connection setup requests in response to signaling failures [26,27]. Drawing on our experiments with partial-route aggregation, we are also investigating the policy and performance implications of combining traffic along a portion of the route, in lieu of employing end-to-end flows with coarser aggregation or larger triggers. Finally, to extend the traffic characterization work, we are studying a more detailed breakdown of Web traffic by content type, as well as the implications of push technology and the new features in the emerging HTTP 1.1 standards. These experiments should lend additional insight into the cost-performance tradeoffs of establishing shortcut connections for long-lived HTTP flows.

References

[1] B. Davie, J. Lawrence, K. McCloghrie, Y. Rekhter, E. Rosen, and G. Swallow, "Use of label switching with ATM." Internet Draft (draft-davie-mpls-atm-00.txt), November 1997.

[2] Y. Katsube, K. Nagami, and H. Esaki, "Toshiba's router architecture extensions for ATM: Overview." Internet Request for Comments (rfc2098.txt), February 1997.

[3] A. Acharya, R. Dighe, and F. Ansari, "IPSOFACTO: IP switching over fast ATM cell transport." Internet Draft (draft-acharya-ipsw-fast-cell-00.txt), February 1997.

[4] A. Viswanathan, N. Feldman, R. Boivie, and R. Woundy, "ARIS: Aggregate route-based IP switching." Internet Draft (draft-viswanathan-aris-overview-00.txt), March 1997.

[5] Multiprotocol Sub-Working Group, *MPOA Version 1.0 Straw Document*, February 1997.

[6] P. Newman, T. Lyon, and G. Minshall, "Flow labelled IP: A connectionless approach to ATM," in *Proc. IEEE INFOCOM*, pp. 1251–1260, March 1996.

[7] S. Lin and N. McKeown, "A simulation study of IP switching," in *Proc. ACM SIGCOMM*, pp. 15–24, September 1997.

[8] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network Magazine*, vol. 11, pp. 10–23, November/December 1997.

[9] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, "A parameterizable methodology for internet traffic flow profiling," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1481–1494, October 1995.

[10] R. Caceres, P. Danzig, S. Jamin, and D. Mitzel, "Characteristics of wide-area TCP/IP conversations," in *Proc. ACM SIGCOMM*, pp. 101–112, September 1991.

[11] S. Keshav, C. Lund, S. Phillips, N. Reingold, and H. Saran, "An empirical evaluation of virtual circuit holding time policies in IP-over-ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1371–1382, October 1995.

[12] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226–255, June 1995.

[13] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, February 1994.

[14] M. Acharya and B. Bhalla, "A flow model for computer network traffic using real-time measurements," in *Proc. Inter. Conference on Telecommunication Systems*, March 1994.

[15] H. J. Fowler and W. E. Leland, "Local area network traffic characteristics, with implications for broadband network congestion management," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1138–1149, September 1991.

[16] R. Jain, "Packet trains – measurements and a new model for computer network traffic," *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, pp. 986–995, September 1986.

[17] J. C. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy, "Potential benefits of delta encoding and data compression for HTTP," in *Proc. ACM SIGCOMM*, pp. 181–194, September 1997.

[18] V. Jacobson, C. Leres, and S. McCanne. `tcpdump`, available at ftp://ftp.ee.lbl.gov, June 1989.

[19] V. Paxson. `tcpreduce`, available at http://ita.ee.lbl.gov/html/contrib/tcp-reduce.html.

[20] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and causes," in *Proc. ACM SIGMETRICS*, pp. 160–169, May 1996.

[21] M. F. Arlitt and C. L. Williamson, "Internet web servers: Workload characterization and implications," *IEEE/ACM Trans. Networking*, vol. 5, pp. 631–644, October 1997.

[22] V. N. Padmanabhan and J. C. Mogul, "Improving HTTP latency," *Computer Networks and ISDN Systems*, vol. 28, pp. 25–35, December 1995.

[23] A. Feldmann, A. C. Gilbert, W. Willinger, and T. Kurtz, "Looking behind and beyond self-similarity: On scaling phenomena in measured WAN traffic," in *Proc. Allerton Conference on Communication, Control and Computing*, 1997.

[24] I. Castineyra, N. Chiappa, and M. Steenstrup, "The nimrod routing architecture." Internet Request for Comments (RFC 1992), August 1996.

[25] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. Networking*, vol. 5, no. 5, pp. 601–615, 1997.

[26] A. Feldmann, "Impact of non-poisson arrival sequences for call admision algorithms with and without delay," in *Proc. IEEE GLOBECOM*, pp. 617–622, November 1996.

[27] D. J. Mitzel, D. Estrin, S. Shenker, and L. Zhang, "A study of reservation dynamics in integrated services packet networks," in *Proc. IEEE INFOCOM*, pp. 871–879, April 1996.