A Simulation Study of IP Switching

Steven Lin

Nick McKeown

Technical Report CSL-TR-97-720

April, 1997

Steven Lin is supported by a National Science Foundation Graduate Fellowship. Nick McKeown is supported in this work by the Alfred P. Sloan Foundation, a Robert Noyce Faculty Fellowship, and Sumitomo Electronic Industries.

A Simulation Study of IP Switching

Steven Lin and Nick McKeown

Technical Report No.: CSL-TR-97-720
April 1997

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
William Gates Computer Science Building, A-408
Stanford, CA 94305-9040
<pubs@shasta.stanford.edu>

Abstract

Recently there has been much interest in combining the speed of layer-2 switching with the features of layer-3 routing. This has been prompted by numerous proposals, including: IP Switching, Tag Switching, ARIS, CSR, and IP over ATM. In this paper, we study IP Switching and evaluate the performance claims made by Newman et al. In particular, using nine network traces, we study how well IP Switching performs with traffic found in campus, corporate, and Internet Service Provider (ISP) environments. Our main finding is that IP Switching will lead to a high proportion of datagrams that are switched; over 75% in all of the environments we studied. We also investigate the effects that different flow classifiers and various timer values have on performance, and note that some choices can result in a large VC space requirement. Finally, we present recommendations for the flow classifier and timer values, as a function of the VC space of the switch and the network environment being served.

Key Words

IP Switching, simulation, label swapping

A SIMULATION STUDY OF IP SWITCHING

Steven Lin and Nick McKeown
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
sclin@leland.stanford.edu nickm@ee.stanford.edu

Abstract

Recently there has been much interest in combining the speed of layer-2 switching with the features of layer-3 routing. This has been prompted by numerous proposals, including: IP Switching [1], Tag Switching [2], ARIS [3], CSR [4], and IP over ATM [5]. In this paper, we study IP Switching and evaluate the performance claims made by Newman et al in [1] and [6]. In particular, using nine network traces, we study how well IP Switching performs with traffic found in campus, corporate, and Internet Service Provider (ISP) environments. Our main finding is that IP Switching will lead to a high proportion of datagrams that are switched; over 75% in all of the environments we studied. We also investigate the effects that different flow classifiers and various timer values have on performance, and note that some choices can result in a large VC space requirement. Finally, we present recommendations for the flow classifier and timer values, as a function of the VC space of the switch and the network environment being served.

What is IP Switching?

The Internet is overwhelmed with traffic, with user demand for bandwidth well in excess of supply. This should come as no great surprise -- the number of users of the Internet has grown exponentially for ten straight years, with each user adding more demand to the network. Over time, each user's traffic tends to increase, adding further to the demand. And at the same time, a larger proportion of traffic traverses multiple domains, swamping campus backbones, corporate intranets, and the core of the Internet. In response to the demand, network equipment vendors and service providers have developed and deployed faster and faster routers. Yet demand for bandwidth far outstrips the current link and switching capacity of the Internet.

The performance of network routers is usually limited by two main tasks: (1) Examining the destination address of incoming datagrams, indexing into a routing table to determine the next hop, and (2) copying incoming datagrams to their outgoing interface. The routing lookup is not straightforward; the router must find the longest prefix match over a space of approximately 2^{31} possible addresses. It is unusual for a router today to perform more than 100,000 address lookups per second. Perhaps surprisingly, the performance of most current routers is limited by their ability to copy data between interfaces. This is because datagrams frequently traverse shared busses multiple times before reaching their outgoing interface.

A variety of techniques have been proposed to accelerate routing lookups [7], and replace shared busses with switched backplanes [8], [9].

Alternative techniques have been proposed that replace longest prefix routing lookups with simple exact matching, and use layer-2 switches to perform fast data copying. We generically call these techniques "label swapping"; they include IP Switching [1], CSR [4], IP over ATM [5], ARIS [3], and Tag Switching [2].

IP Switching integrates IP routing with fast lookup and copying mechanisms of an ATM switch[1]. Based on IFMP (Ipsilon Flow Management Protocol) [10] [11], IP Switching has received a lot of attention.

Consider the IFMP-speaking IP Switch shown in Figure 1. We assume that the switch is surrounded by a number of identical neighbors. Our switch begins by establishing adjacency with its neighbors, identifying them as fellow IFMP switches. IP datagrams arrive at the switch encapsulated into AAL5 frames and segmented into ATM cells. These datagrams are part of a flow between two user processes, or between two end points in the network. Initially, all datagrams from all flows are received and transmitted on a single, default ATM virtual circuit (VC). Our switch attempts to identify flows, classifying them according to their expected

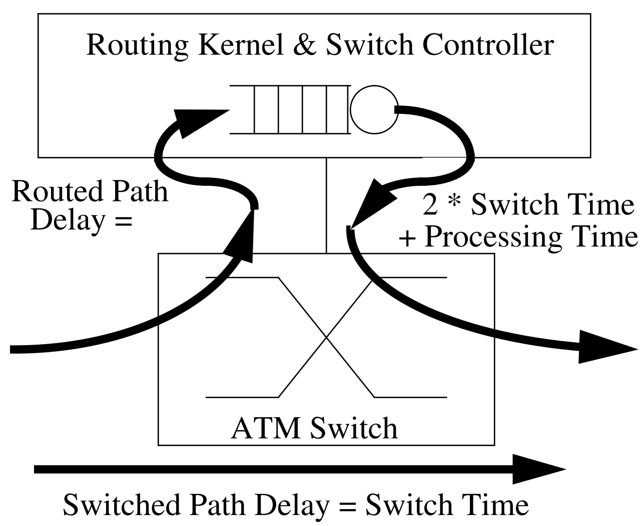

Figure 1: Simple Block Diagram of IP Switch

[1]Although IP Switching could conceivably be used on any layer-2 switch, for the remainder of this paper we only consider ATM, since this is what has been used in practice.

duration. If the switch decides that the flow will be long lasting, it will attempt to establish a unique VC for the flow, so that it may be switched quickly in hardware. The switch tells its upstream neighbor to begin sending all datagrams that are in that flow on a separate VC. This is achieved by sending an IFMP redirect message to the next switch that is upstream in the flow. Similarly, our switch will receive a redirect message from its downstream neighbor when the downstream neighbor detects a flow. We can think of the redirect messages as *binding* the flow to an ATM virtual circuit. When the flow has been successfully bound, our switch can perform hardware layer-2 switching from the incoming VC to the outgoing VC, without consulting any routing tables or performing any other processing.

This is shown in Figure 1, where the first datagrams in a flow must traverse the 'slow path' through a conventional IP router. The router reassembles cells into AAL5 frames, removes the IP datagram, and makes a routing decision. The datagram is encapsulated and segmented into cells before being sent on its way. In addition, a packet classifier attempts to recognize flows and estimate whether they are worth binding to an ATM VC. If so, a redirect message is sent to the upstream switch, and future datagrams will travel along the fast path. Higher performance is achieved if a large proportion of the datagrams are able to take the fast path. For this reason, we use the percentage of datagrams switched as the primary metric in our study.

Flow Classification

Each IP Switch contains a flow classifier: an entity that identifier flows and decides whether or not they should be bound to an ATM VC. IFMP defines two types of flow: (1) Flow Type 1: a flow of IP datagrams between two processes. In other words, datagrams that all contain the same values for the following IP and transport fields: IP version, source and destination addresses, header length, type-of-service, time-to-live (TTL), protocol, and transport port numbers. (2) Flow Type 2: a flow of datagrams between two IP addresses. All the datagrams in a type 2 flow contain the same IP version, source and destination addresses, header length, and TTL.

An IP Switch must have a policy to determine when a flow will be bound to an ATM VC. This policy is outside the scope of the IFMP protocol description, but must be present in any IFMP implementation. The IP Switch designer may choose from a variety of techniques to determine if a flow should be bound to a VC. On one hand, it is clearly desirable to bind as many flows as possible, because this will enable a large number of datagrams to be switched at layer-2, reducing the number of layer-3 routing decisions. On the other hand, if short-lived flows, or flows with a small number of datagrams are switched, this will waste the VC space of the switch. It will also lead to a large number of redirect messages.

Our simulation uses three types of flow classifier. Each classifier is designed to switch a large number of datagrams at layer-2, while wasting a small number of VCs, and sending as few redundant redirect messages as possible. The first classifier is the *X/Y Classifier*. This watches until it sees X datagrams within a Y second window that all match a specific flow. The switch then binds the flow to an ATM VC, switching any future datagrams that meet that flow specification. The motivation behind the X/Y Classifier is that if there have already been X datagrams from a flow in Y seconds then it is reasonable to expect that there will be more datagrams from this flow in a short amount of time. The only remaining problem is to determine values for X and Y.

The second classifier is the *Protocol Classifier*, which simply assigns all TCP packets to flows. The rationale for this classifier is that connection-based communications generally last longer and have more datagrams sent over a short time than connectionless-based communications.

Finally, the *Port Classifier* uses the transport-layer port numbers to decide which flows to bind. In our study, we decided to bind all TCP flows going to or from the following ports: ftp-data (20), telnet (23), smtp (25), http (80), and nntp (119). The goal is to select those applications which tend to generate long-lasting flows, and/or flows which contain a large number of datagrams. It could be argued that additional or alternative services/ports be used instead of the ones above; these were picked because they made up the largest percentage of TCP packets seen in the traces we analyzed.

Delays in Flow Establishment

There are also several delays associated with the creation and deletion of flows, with values unspecified in the IFMP description. These delays are either chosen or will occur naturally in any implementation. The following delays were modeled by our simulation:

Flow Creation Delay: In practice, there is a delay between the time a flow is detected, and the time at which the upstream node begins sending datagrams on the new VC. This delay is due to the time required to form the redirect message, its propagation delay upstream, and the processing time of the message at the upstream node. We model this delay as a constant.

Flow Deletion Delay: There must be some decision made as to when to delete a flow. According to the IFMP specification, the upstream neighbor will time-out a flow if it does not receive additional redirect messages. To prevent flows from timing out prematurely, a switch sends redirect messages upstream at regular intervals. In our simulation, we approximate the time-out by deleting a flow after some constant number of seconds of inactivity.

Questions about IP Switching Performance

In our study, we set out to answer six main questions about the performance of IP Switching:

- Are the claims of switching performance presented by the inventors of IP Switching valid?

Trace Name	Description	Date / Time	Duration	# Packets	Avg. Packets/ Sec	% TCP	Top 4 Ports	# of Flows FT1	FT2
DEC-PKT-1	Corporate Gateway	3/8/95 (Wed) 22:00	1 hour	2,983,221†	829	72.2	dns nntp ftp-data smtp	75,438	45,591
DEC-PKT-2	Corporate Gateway	3/9/95 (Thu) 02:00	1 hour	3,467,733†	963	76.8	nntp dns ftp-data rlogin	75,739	37,356
DEC-PKT-3	Corporate Gateway	3/9/95 (Thu) 10:00	1 hour	3,909,046†	1086	73.5	dns nntp ftp-data http	142,193	75,768
DEC-PKT-4	Corporate Gateway	3/9/95 (Thu) 14:00	1 hour	5,049,790†	1403	76.5	ftp-data dns nntp http	153,527	70,745
LBL-PKT-4	Corporate Gateway	1/21/94 (Fri) 14:00	1 hour	896,690†	249	96.2	telnet nntp ftp-data rlogin	13,184	5,680
LBL-PKT-5	Corporate Gateway	1/28/94 (Fri) 14:00	1 hour	747,204†	208	90.7	telnet ftp-data nntp rlogin	10,708	5,757
FIXWEST1	Backbone Router	2/28/96 (Wed) 13:45	12.866 minutes	11,550,347	14,962	78.7	http dns ftp-data smtp	752,439	303,661
FIXWEST2	Backbone Router	9/26/96 (Thu) 12:17	20.566 minutes	15,315,131	12,411	61.9	http dns ftp-data smtp	777,389	365,467
UCB	Campus Backbone	1/16/95 (Mon) 19:02	6 minutes	864,805	2402	82.0	telnet nntp rlogin dns	32,883	14,153

Table 1: Trace Statistics

†These numbers reflect those packets analyzed in the simulation and include all TCP and UDP packets. There were other packets in the raw traces, but not enough information associated with these other packets to analyze them.

- To work effectively, does IP Switching require an exceptionally large VC space?
- How do the flow creation delay and flow deletion delay affect performance?
- Which flow classifiers work best in different environments?
- Does IP Switching induce large amounts of datagram misordering?
- Given a fixed VC Space and knowledge about the environment in which the IP Switch will be used, how should parameters be chosen to provide the best performance?

Experimental Setup / Procedure

Nine network traces were analyzed by a high-level simulation of an IP Switch. Six of the traces (DEC-PKT-1 through DEC-PKT-4, LBL-PKT-4 and -5) are each one hour in length and were taken on corporate gateways to the Internet. These traces are freely available from the Internet Traffic Archive[2]. Two more, shorter traces were taken at the FIX-WEST west coast federal interexchange node, and were obtained from the National Laboratory for Applied Network Research (NLANR)[3]. Finally, the last trace was a four-day

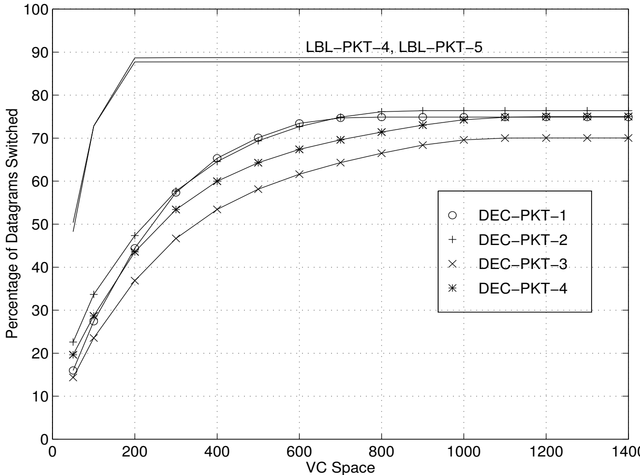

Figure 2: % Datagrams Switched vs. VC Space; Corporate Traces, Flow Type 1

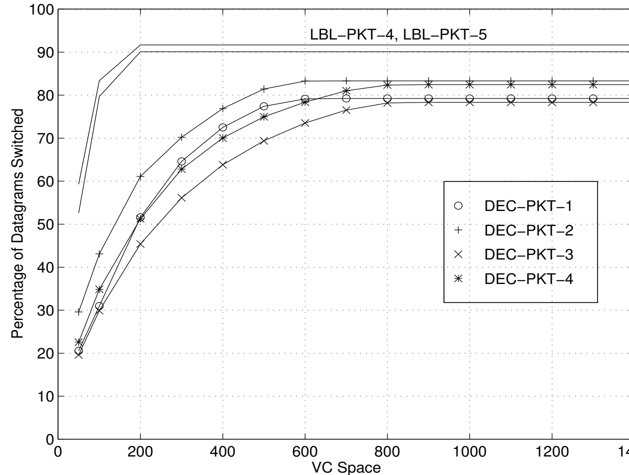

Figure 3: % Datagrams Switched vs. VC Space; Corporate Traces, Flow Type 2

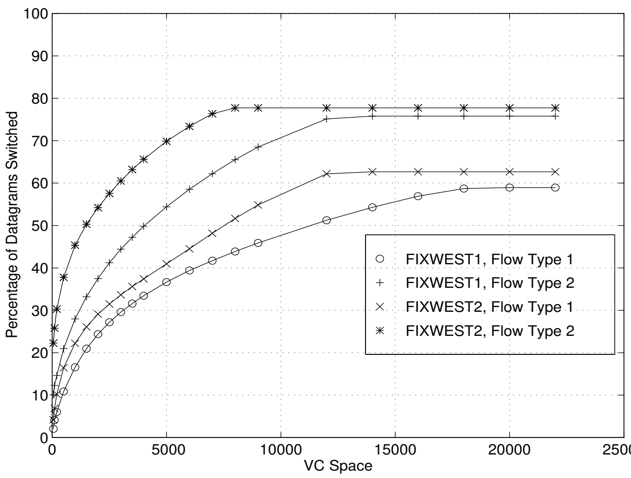

Figure 4: % Datagrams Switched vs. VC Space; ISP Traces

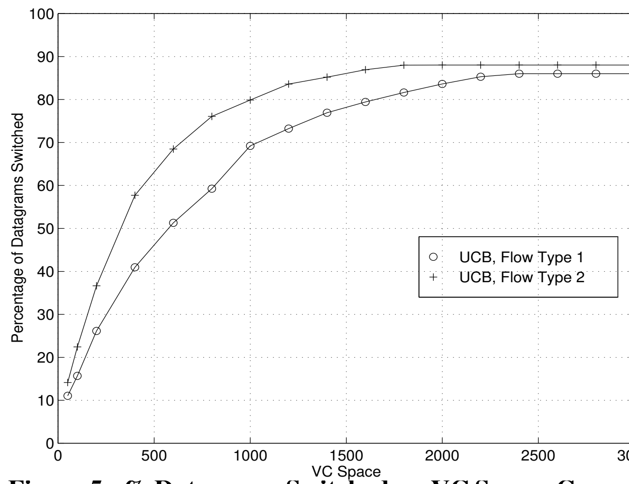

Figure 5: % Datagrams Switched vs. VC Space; Campus Trace

long trace of two campus backbone rings at the University of California, Berkeley[4] (at time of this writing, only six minutes of the trace has been analyzed; we expect similar behavior throughout the entire trace).

Some statistics from the traces are shown in Table 1. The "number of packets" column represents the number of datagrams actually processed by the simulator and includes all TCP and UDP datagrams. The "% TCP" column shown the percentage of IP datagrams with the protocol field set to TCP. The "top 4 ports" column shows, in order, the four most popular source and destination ports for TCP and UDP. Finally, the "# of flows" column shows the number of flows of type 1 and flows of type 2.

[2]Internet Traffic Archive traces are available at http://town.hall.org/Archives/pub/ITA/html/traces.html

[3]FIXWEST/NLANR traces are available at ftp://oceana.nlanr.net/Traces/FR+/. Also see the NLANR home page at http://www.nlanr.net for more information.

[4]The UC Berkeley trace is currently unavailable.

Our simulation modeled a single IP switch. We were able to vary the VC space of the switch, the flow creation and deletion delays, the flow type to use, and the flow classifier. Results generated by the simulation include percentage of datagrams switched (vs. those forwarded by the routing kernel), and maximum number of VCs used during the simulation. The source code for the simulation is freely available via anonymous ftp at: ftp://klamath.stanford.edu/pub/outgoing/issim.tar.gz.

Results

Percentage of Datagrams Switched and VC Space Requirements:

We begin by addressing our first two questions: Are the claims of switching performance presented by the inventors of IP Switching valid? And to work effectively, does IP Switching require an exceptionally large VC space?

Figures 2 through 5 show the results of a set of simulations in which the VC space of the switch is varied. A new

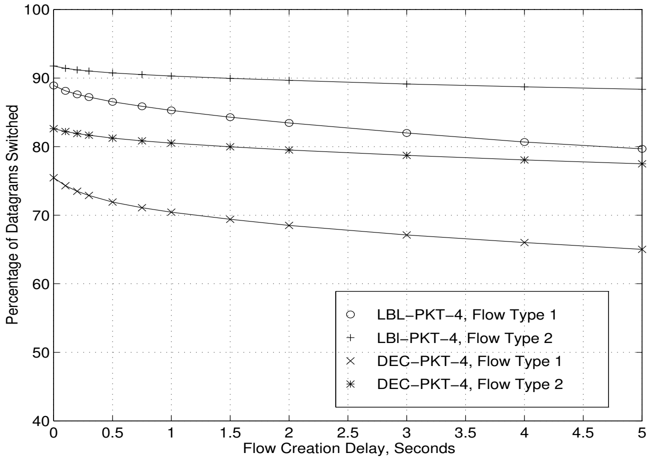

Figure 6: % Datagrams Switched vs. Flow Creation Delay; Corporate Traces

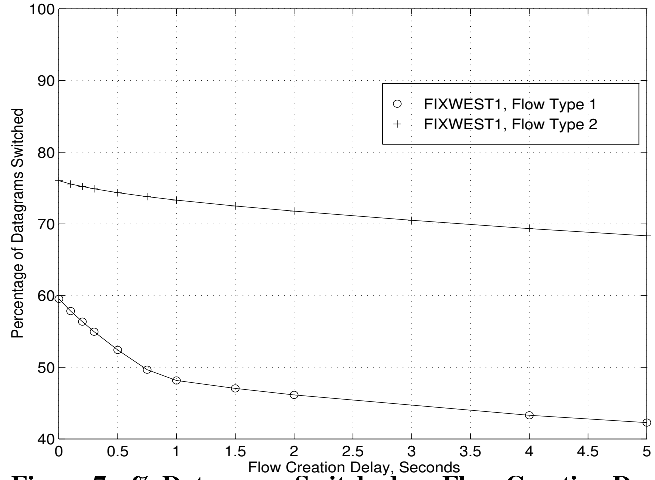

Figure 7: % Datagrams Switched vs. Flow Creation Delay; ISP Traces

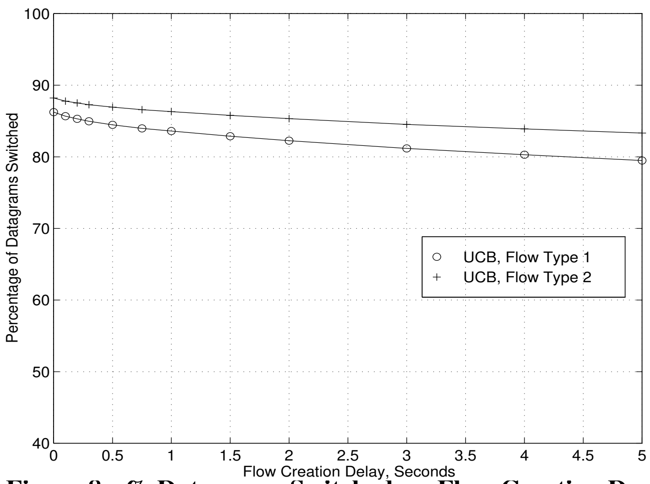

Figure 8: % Datagrams Switched vs. Flow Creation Delay; Campus Trace

flow can only be bound to an ATM VC if there is at least one VC available. In these experiments, an X/Y classifier was used with X=10 packets and Y = 20 seconds. From these plots, we can begin to answer our questions. First notice that in every trace, at least 75% of the datagrams are switched when flow type 2 is used (we will see later that this percentage can be slightly increased with a different choice of simulation parameters). This seems to corroborate the inventors' claim that about 80% of datagrams will be switched [1] [6]. The percentage is lower when flow type 1 is used; this is because a flow of type 2 contains multiple flows of type 1. To be bound, each flow of type 1 must fulfill the X/Y requirements of the classifier, whereas a flow of type 2 must fulfill these requirements only once.

Notice that for certain environments, namely at the corporate gateway and on the campus backbone, maximum performance can be reached with as few as one or two thousand VCs. However, then the Internet backbone traces are examined, it is apparent that about 10,000 VCs are required to reach maximum performance. Although a high percentage of packets can still be switched, this VC usage is likely to increase with time and may become excessive for some hardware.

Finally, we notice that the behavior of similar traces is similar. For example, all four DEC traces exhibit similar behavior, as do both LBL traces. In order to clarify our results, the remainder of the plots feature only one representative plot from each data set, even though all nine traces were processed by the simulator.

Flow Creation Delay:

Next we try to answer our questions on delay: How do flow creation and deletion delays affect performance? We start by examining the sensitivity of performance to the flow creation delay.

As mentioned earlier, the flow creation delay is the time from when a redirect message is sent to the time at which datagrams begin to appear on the new VC. We model this delay as a constant, and see how performance is sensitive to its value.

As seen in Figures 6 through 8, the fraction of datagrams switched drops as the flow creation delay increases. This is expected; increasing the time to redirect a flow is equivalent to increasing the time to classify the flow. Both allow more datagrams to be forwarded through the routing path before hardware switching takes over. Because the backbone traces have a higher datagram arrival rate, they are affected more by this delay than the gateway and campus traces. Even at the maximum tested value of 5 seconds, however, the percentage of packets switched drops by less than 18% in all cases. In practice, the flow creation delay for an IP switch is on the order of 10 ms. For delays in this neighborhood, there is onlya negligible effect on performance.

Flow Deletion Delay:

We now turn our attention to the flow deletion delay. This parameter is unspecified in the IFMP specification, but must be chosen by the implementor. There are conflicting

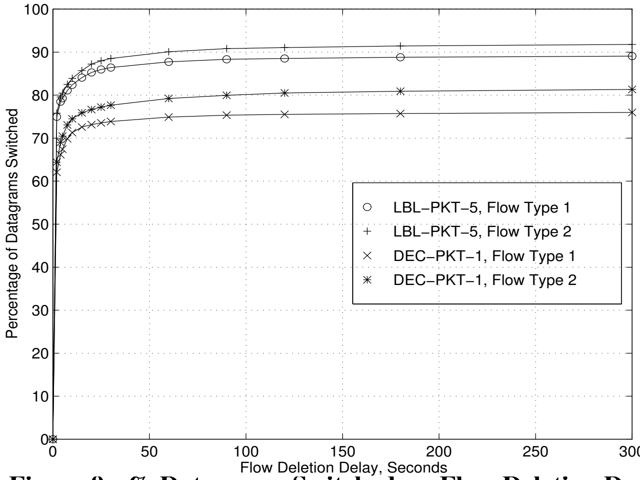

Figure 9: % Datagrams Switched vs. Flow Deletion Delay; Corporate Traces

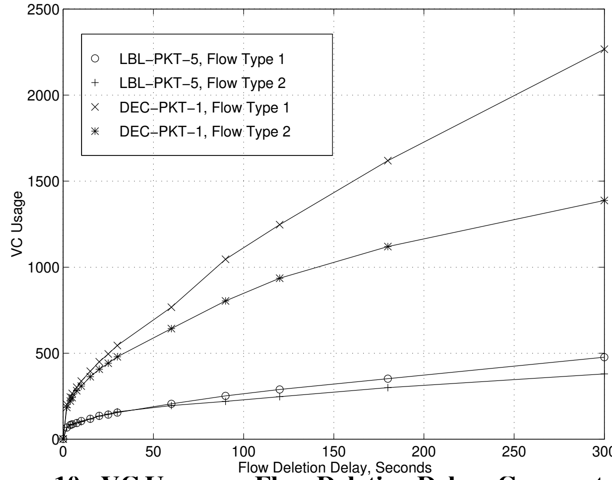

Figure 10: VC Usage vs. Flow Deletion Delay; Corporate Traces

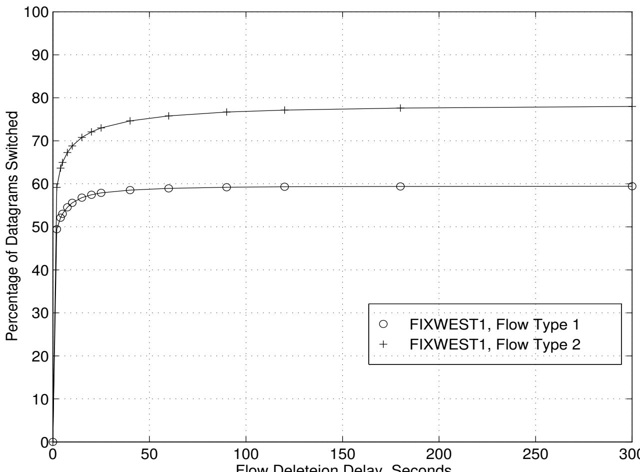

Figure 11: % Datagrams Switched vs. Flow Deletion Delay; ISP Traces

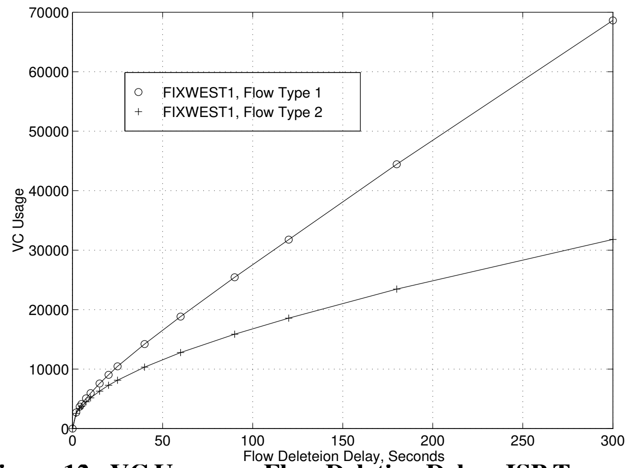

Figure 12: VC Usage vs. Flow Deletion Delay; ISP Traces

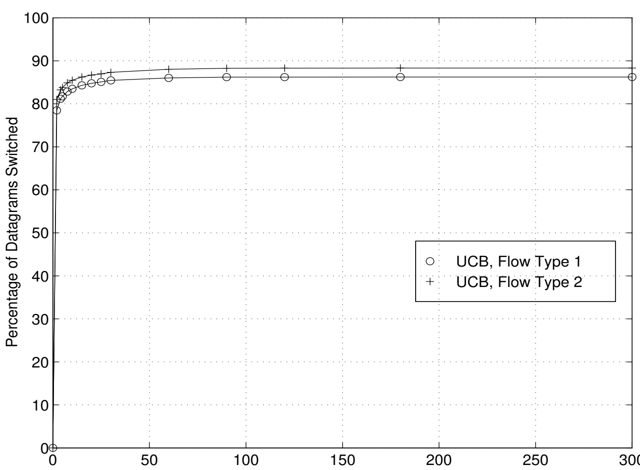

Figure 13: % Datagrams Switched vs. Flow Deletion Delay; Campus Trace

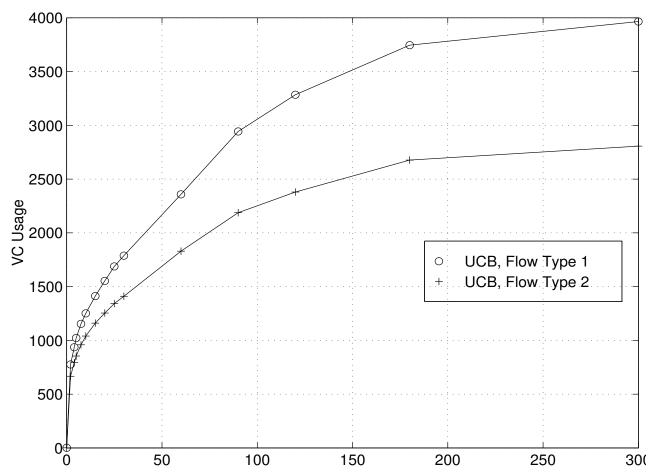

Figure 14: VC Usage vs. Flow Deletion Delay; Campus Trace

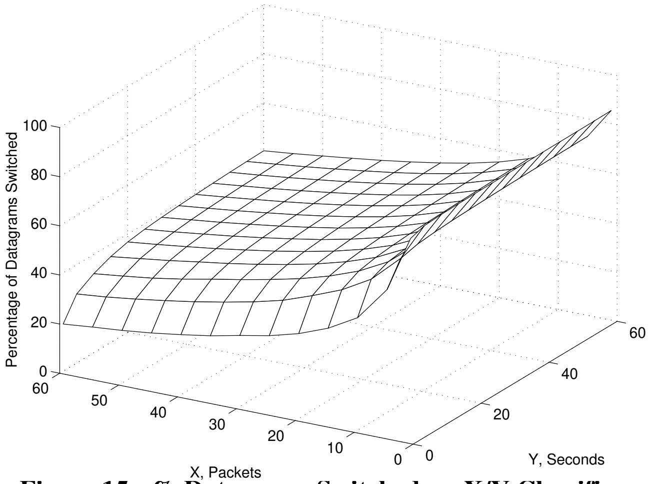

Figure 15: % Datagrams Switched vs. X/Y Classifier; FIXWEST2, Flow Type 1

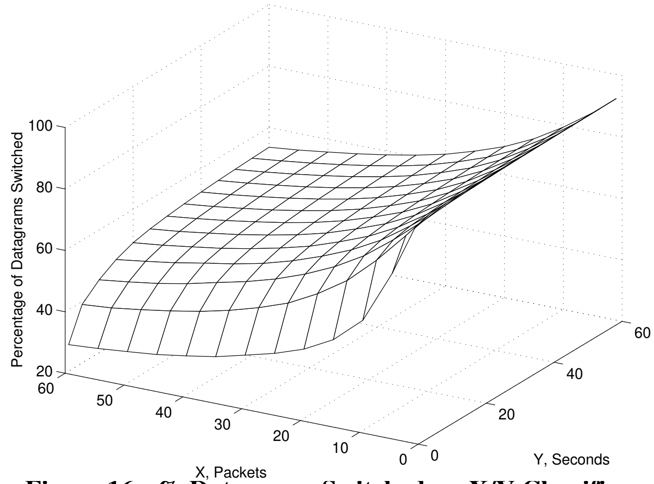

Figure 16: % Datagrams Switched vs. X/Y Classifier; FIXWEST2, Flow Type 2

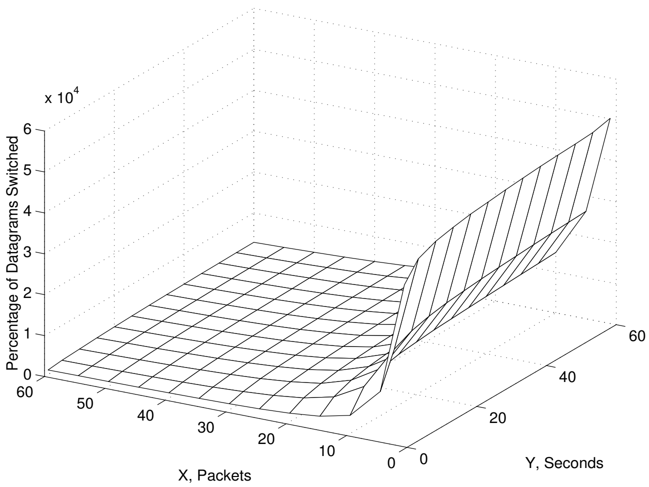

Figure 17: VC Usage vs. X/Y Classifier, FIXWEST2, Flow Type 1

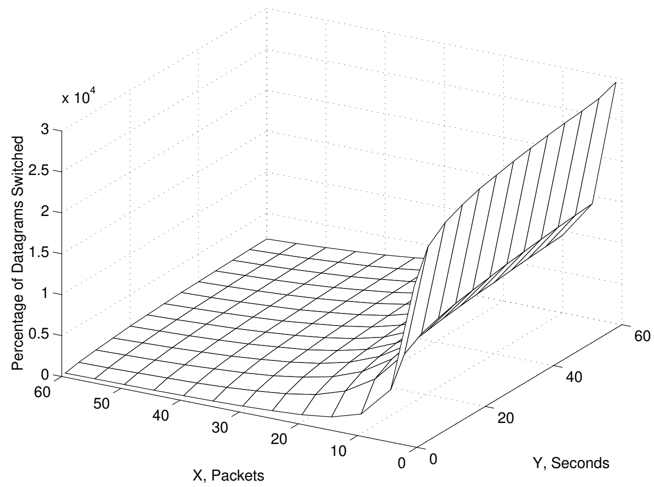

Figure 18: VC Usage vs. X/Y Classifier, FIXWEST2, Flow Type 2

goals in the selection of this value. The dilemma here is that a higher percentage of datagrams can be switched by being reluctant to delete flow bindings too early. However, VC space can be conserved by attempting to delete the flow binding (thus freeing a VC) as soon as possible.

From the simulation results in Figures 9 through 14, it seems that a fairly small flow timeout value (30 to 60 seconds) is sufficient to achieve close to maximum performance, while preventing an excessive waste of VC space. If, however, the selected deletion delay is too large, a switch can easily require over 32,000 VCs. In the Internet backbone where the traffic arrival rate is high, this can occur even at seemingly reasonable choices, such as 2 or $2^1/_2$ minutes.

Flow Classifier:

Our next question is: Which flow classifier works best in different environments? All of the previous plots were made with an X/Y classifier, with X=10 and Y=20. We start by seeing how changing X and Y affects the performance of the system. Figures 15 through 18 show results from one of the backbone traces as X is varied from 2 to 60 datagrams and Y is varied from 1 to 60 seconds. The other types of traces show similar trends, although with lower VC usage.

Using the extremely permissive classifier of “2 packets in 1 minute”, Figure 16 shows that as many as 90% of the datagrams qualify for switching. However, as shown in Figure 18, this performance is tempered by the extremely high VC usage. On the other hand, a strict classifier of “60 packets in 1 second” conserves VC space but makes it unlikely that a flow will be classified. Obviously, the best solution is somewhere between the two. It seems that the best choice depends on the VC space of the switch. If a large VC space is available, a more permissive classifier will produce better performance. However, if the VC space is limited, a permissive classifier will waste some VCs on less productive flows, and so a stricter classifier is a better choice. For example, if only 5000 VCs are available using flow type 2 in a FIXWEST-like environment, the “15 packets in 10 seconds” classifier, resulting in 69% packets switched, is a good compromise.

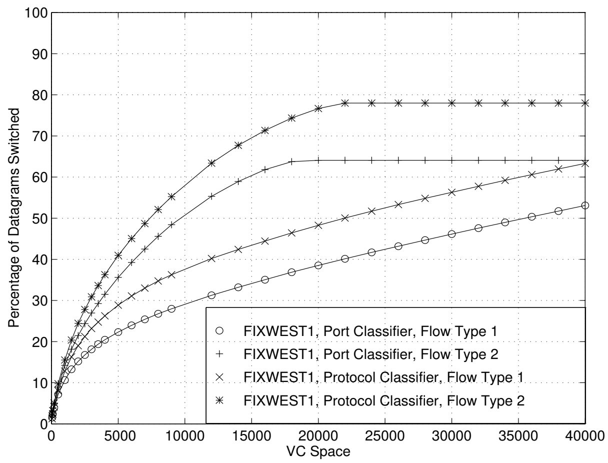

Figure 19: % Datagrams Switched vs. VC Space

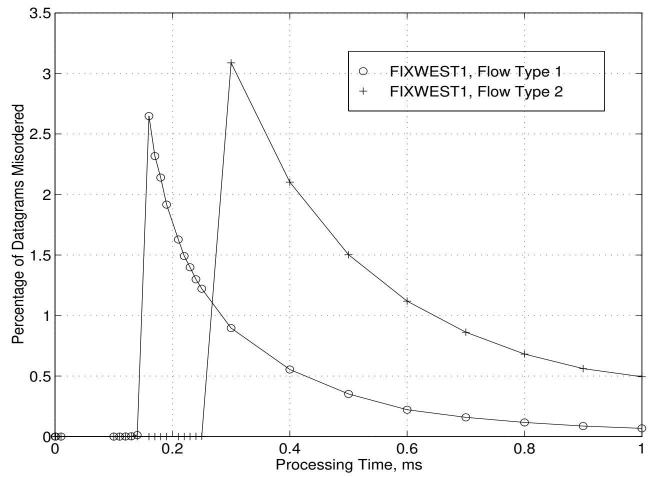

Figure 20: % Datagrams Misordered vs. Processing Time

It should be pointed out that in the gateway and campus traces, even using the "2 packets in 1 minute" classifier, no more than 6000 VCs are ever used (and all classifiers with X > 2 packets require fewer than 2500 VCs). In other words, as before, for this type of environment, as long as the VC space is reasonably large, a permissive classifier may be used.

We also considered two other types of flow classifier: a Protocol Classifier and a Port Classifier. Although neither the protocol nor port fields appear in the flow type 2 descriptor, it is still possible to use these fields for deciding when to establish a flow binding, but disregard the fields when deciding which flow a given datagram is part of.

Figure 19 shows the performance of these two classifiers on one of the corporate gateway traces, as a function of the switch VC space. Although the Protocol classifier comes very close to the performance of the X/Y classifier in terms of number of datagrams switched, it requires approximately twice as many VCs as the X/Y classifier to achieve this performance. The Port classifier is not able to achieve performance comparable to the X/Y classifier, even with a VC space of 40,000 VCs.

Datagram Misordering:

Finally, we examine the question: Does IP Switching induce large amounts of datagram misordering? Recall that IP is a non-reliable network layer [12]. In other words, if packets are lost, duplicated, or misordered, there is no violation of the letter or the spirit of the specification. At the same time, however, these anomalities may reduce performance of the transport layer.

IP Switching can potentially misorder IP datagrams. Looking back at Figure 1, observe that before a flow is switched at layer-2, all packets travelling via the default VC must be switched once to enter the routing hardware, then processed, and switched again before leaving the IP Switch. It is possible that while some datagrams associated with the flow are being processed by the routing kernel, arriving datagrams on a new VC are switched and depart before the earlier datagrams.

To examine this, we set the switch latency to be 10 microseconds (approximately four ATM cell times at 155 Mbps). Incoming datagrams that are not part of a detected flow are placed onto a FIFO queue, where they wait to be routed. The router works at a constant processing time per datagram. Note that this processing time determines the router's "packet per second" capacity. Figure 20 shows the percentage of datagrams misordered as a function of router processing time. For this figure, we say that a datagram is misordered if it leaves the IP switch before a previous datagram from the same flow. As the processing time increases from 0 seconds/datagram, the percentage of datagrams misordered increases, up to a maximum of about 3.5% for flow type 2. At these peak points, the routing kernel is just barely able to keep up with the influx of datagrams needing to be routed. The routing queue is long, leading to the high percentage of misordered datagrams. As we further increase the processing time, the simulation begins to break down as the queue becomes unstable. As the queue grows without bound, fewer flows are classified and so fewer datagrams are misordered; however, performance is extremely poor in this region of the plot. With flow type 1, instability occurs for routers that route less than approximately 6,250 pps; whereas for flow type 2, instability is encountered for routers with less than approximately 3,300 pps routing capacity. Assuming that the IP Switch is not operated near the unstable region, fewer than 0.1% of datagrams are misordered. An informal survey of 105 workstations in Stanford University's (non IP Switched) public computing labs revealed that an average of 0.11% of TCP datagrams arrive misordered. This would seem to indicate that a single IP Switch causes about the same proportion of datagram misordering as seen in conventional networks.

Conclusions

There are a number of parameters that determine the performance of an IP switch. Some are determined by the switch

VC Space	Gateway	Campus/Enterprise Backbone	Internet Backbone
1K VCs	Clasifier: X = 5 / Y = 15 sec. Flow Deletion Delay: 30-120 sec. Expected Performance: 85%	Clasifier: X = 40 / Y = 40 sec. Flow Deletion Delay: 30-60 sec. Expected Performance: 79%	Clasifier: X = 50 / Y = 10 sec. Flow Deletion Delay: 30 sec. Expected Performance: 38%
2K VCs	Clasifier: X = 5 / Y = 60 sec. Flow Deletion Delay: 30-120 sec. Expected Performance: 90%	Clasifier: X = 10 / Y = 45 sec. Flow Deletion Delay: 30-60 sec. Expected Performance: 89%	Clasifier: X = 30 / Y = 10 sec. Flow Deletion Delay: 30 sec. Expected Performance: 47%
8K VCs	Clasifier: X = 2 / Y = 60 sec. Flow Deletion Delay: 30-120 sec. Expected Performance: 93%	Clasifier: X = 5 / Y = 60 sec. Flow Deletion Delay: 30-60 sec. Expected Performance: 92%	Clasifier: X = 10 / Y = 10 sec. Flow Deletion Delay: 30 sec. Expected Performance: 71%
16K VCs	Clasifier: X = 2 / Y = 60 sec. Flow Deletion Delay: 30-120 sec. Expected Performance: 93%	Clasifier: X = 2 / Y = 60 sec. Flow Deletion Delay: 30-60 sec. Expected Performance: 95%	Clasifier: X = 5 / Y = 40 sec. Flow Deletion Delay: 30 sec. Expected Performance: 84%
32K VCs	Clasifier: X = 2 / Y = 60 sec. Flow Deletion Delay: 30-120 sec. Expected Performance: 93%	Clasifier: X = 2 / Y = 60 sec. Flow Deletion Delay: 30-60 sec. Expected Performance: 95%	Clasifier: X = 2 / Y = 60 sec. Flow Deletion Delay: 30 sec. Expected Performance: 92%
∞ VCs	Clasifier: all packets Flow Deletion Delay: ∞ Expected Performance: 99%	Clasifier: all packets Flow Deletion Delay: ∞ Expected Performance: 98%	Clasifier: all packets Flow Deletion Delay: ∞ Expected Performance: 97%

Table 2: Summary of Recommendataions

designer; for example, the number of VCs supported by the switch and the per-packet processing time in the router. Others may be determined by the manager of the switch: the flow classifier and the flow deletion delay. Yet others are determined largely by the environment in which the switch operates; for example, the delay from the time a redirect message is sent until datagrams begin to arrive on the new VC.

We summarize all of our findings by making some specific recommendations for different environments. These are shown in table 2, and assume that flow type 2 is used (see below)[5].

As expected, our study shows that the switch designer should aim to support as many VCs as economically viable. Most ATM switches today support between 4096 and 65,536 VCs. Our results indicate that switches supporting at least 10,000 VCs will see acceptable performance with today's traffic. However, the volume and diversity of Internet traffic is growing rapidly. We predict that before long, more than 65,536 VCs will be required to provide adequate performance in the core of the Internet. Note that this is the case for flows of type 1 and flows of type 2.

Our study also shows that a slower router will lead to more misordered datagrams and eventually instability of the IP Switch. However, in all the environments we studied, we found that the router need only forward as little as 6,250 packets per second. This is readily achievable today with low-cost general purpose computers.

We found that the X/Y classifier works better than either the port or protocol classifiers. It is simple to select values for X and Y that lead to over 75% switched datagrams. When enough VCs are available, our recommended values are: X = 2 packets, and Y = 60 seconds. Note also that the X/Y classifier does not depend on well-known port or protocol numbers; it requires no modifications as different applications come into favor.

Our findings lead us to caution against using large flow deletion delays. These lead to unnecessary exhausting of VCs, and therefore reduce switching performance. We found that a value of 30 seconds suits all of the environments tested.

Finally, our results indicate that the flow creation delay should be kept well below one second. In practice, this should be readily obtained — we expect the delay to be no larger than 20 ms.

We now turn to another decision: should we use flow type 1 or flow type 2? Because the flow type 1 description is a superset of the flow type 2 description, using flow type 2 results in a lower VC usage and a higher switching percentage. This is reflected in our results. In all cases, flow type 2 results in better performance with lower VC usage. Why, then, would anyone use flow type 1? When using flow type 2, all packets from the same source/destination pair are lumped together in the same flow. This is certainly more efficient. However, it does not allow the switch to distinguish different application flows, making it impossible to provide differing classes of service. In reality, we expect that most flows will

[5] These recommendations are based on the set of networks studied. Results will vary from network to network, and these values should be used only as a guide.

be classified as type 2, with the network administrator selecting certain specific protocol or port values (i.e. specific high-priority applications) for which to use flow type 1.

It is clear that the volume of Internet traffic is growing exponentially. We predict that IP switches will soon need in excess of 64K VCs to perform well. As the traffic intensity and number of flows increase, it appears that the number of VCs required will also increase; perhaps to the point of overwhelming ATM switches. We can alleviate this problem by using less specific flow descriptions. Using a flow specifier based on IP address prefixes (i.e. aggregated netids) rather than the full 32-bit address should allow multiple type-2 flows to map into one prefix-based flow. This would reduce the number of VCs required, but also require a modification to the IFMP specification.

Acknowledgments

Steven Lin was supported by a National Science Foundation Graduate Fellowship. We would also like to acknowledge Jeff Mogul of Digital Equipment Corporation for the DEC-PKT-* traces, Vern Paxson of the Lawrence Berkeley Laboratory for the LBL-PKT-* traces (see [13]), Richard Edell of the University of California, Berkeley for the UCB trace, NLANR for the Fixwest traces, and the members of Ipsilon Networks' technical staff who assisted us with questions about the IFMP protocol.

References

[1] Peter Newman, Tom Lyon, and Greg Minshall. "Flow Labelled IP: A Connectionless Approach to ATM." Proc. IEEE Infocom '96, vol. 3, pp. 1251-60.

[2] Yakov Rekhter, Bruce Davie, Dave Katz, Eric Rosen, and George Swallow. "Cisco Systems' Tag Switching Architecture Overview." IETF RFC 2105, Feb. 1997.

[3] Arun Viswanathan, Nancy Feldman, Rick Boivie, and Richard Woundy. "ARIS: Aggregate Route-Based IP Switching." IETF Internet Draft, March 1997.

[4] H. Esaki, K.-I. Nagami, and M. Ohta. "High Speed Datagram Delivery over Internet using ATM Technology." Proc. Networld+Interop, Paper E12-1, Las Vegas, NV, Mar. 1995.

[5] G. Parulkar, D. Schmidt, and J. Turner. "IP/ATM: a Strategy for Integrating IP with ATM." Proc. ACM SIGCOMM, Cambridge, MA. Sept. 1995.

[6] Ipsilon Networks. "IP Switching: The Intelligence of Routing, the Performance of Switching." White Paper. Available at http://www.ipsilon.com/productinfo/techwp1.html.

[7] W. Doeringer, G. Karjoth, and M. Nassehi. "Routing on Longest-Matching Prefixes." IEEE/ACM Trans. Networking, Feb. 1996, pp. 86-97.

[8] C. Partridge. "A Fifty Gigabit per Second IP Router." Submitted to SIGCOMM 96.

[9] Nick McKeown, Martin Izzard, Adisak Mekkittikul, William Ellersick, and Mark Horowitz. "The Tiny Tera: A Packet Switch Core." IEEE Micro Mag., Jan-Feb 1997.

[10] P. Newman, W. L. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. "Ipsilon Flow Management Protocol Specification for IPv4." IETF RFC 1953, May 1996.

[11] P. Newman, W. L. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. "Transmission of Flow Labeled IPv4 on ATM Data Links." IETF RFC 1954, May 1996.

[12] Jon Postel. "Internet Protocol." IETF RFC 791, Sept. 1981.

[13] V. Paxson and S. Floyd. "The Failure of Poisson Modeling." IEEE/ACM Trans. Networking. 3 (3), June 1995, pp. 226-244.