Nondeterministic classifier performance evaluation for flow based IP switching

Jouni Karvo, Mika Ilvesmäki
Helsinki University of Technology
P.O.Box 1100, FIN-02015 HUT, Finland
Email: `{jouni.karvo,mika.ilvesmaki}@hut.fi`

Abstract

In modern IP networks, processing cost in network nodes is considered as a bottleneck. This problem is tackled with traffic based IP switching. The performance of traffic based IP switching depends heavily on flow classification. We demonstrate a method to evaluate the performance gains available with this technique with an optimal nondeterministic classifier giving a practical lower bound for processing cost and compare it with two real life classifiers using recorded traces.

Keywords

IP switching, flow classification, cost optimization, Internet, traffic measurements

1 INTRODUCTION

In modern IP networks, processing resources needed for packet routing are considered as a bottleneck. Several proposals have been made to reduce the load caused by routing for Internet flows in broadband networks. The general method is to label flows and to apply link layer switching functions instead of network layer routing functions. Two main approaches to IP switching exist; these are the flow based IP switching and the topology based IP switching solutions.

1.1 IP flow

IP switching, regardless of the practical realizations, is based on detecting or predicting IP flows. An IP flow is a series of IP packets that share some common properties, such as the IP address prefix, or IP address pair and perhaps also the TCP/UDP port number pair.

Various flow metrics were widely deployed in (Claffy *et al.* 1995, Claffy 1994),

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35388-3_42

H. R. van As (ed.), *High Performance Networking*

where a flow was defined to be *a series of packets travelling from a constant source to a constant destination.* The definitions of source and destination, and therefore the definition of flow, may be freely chosen to include anything from IP address prefixes defining parts of networks to IP address and TCP port quadruples defining applications. The level of flow definitions is referred to as flow granularity. The higher or finer the granularity the more flows are destined to be created.

1.2 IP switching solutions

Traffic based flow switching may assign several flows to one connection. This aims for a more efficient and accurate use of resources. Connections are torn down when a flow timeout occurs because no more packets are being sent on a connection. This timeout value is usually in the order of 30–120 seconds as suggested by Claffy *et al.* (1995). Opinions on whether the traffic based approach scales well when the amount of traffic increases differ (Claffy 1994, Lin and McKeown 1997), but since this issue depends on the connection space available (VC space, in the case of ATM) and on the definition of the flow itself (granularity and timeout values), conclusive statements can not yet be made. Traffic based IP switching is an approach dealing with the problems of short vs. long duration Internet traffic on top of ATM.

The traffic based IP switch is an ordinary routing processor augmented with a flow classifier. The flow classifier detects packet flows and assigns them to their own connections thus reducing the amount of packets that are forwarded through the routing processor. The current available technological solutions for traffic based IP switching use local decision making when determining the flows to be switched. If global decision making is to be used it means using either ATM signalling or RSVP.

Topology switching relies on predefined connections based on routing information and traffic monitoring. A 'topology connection' might include several multiplexed connections to the same part of the network. On the other hand topology based switching might be keeping connections up even if no data is being transmitted. Also the different QoS levels needed require setting up additional paths to a destination with varying QoS characteristics, thus possibly wasting connection space and resources. The issue of aggregated QoS demands needs also further investigation. In a network where switching is based on topology the edge routers recognize aggregated traffic flows which are then switched, rather than routed, through the core network.

The processing capacity available in an IP switch is divided between switched and nonswitched traffic. This division essentially constitutes the process of flow classification. The switched flows can then further be divided to different categories (services) thus providing the possibility for prioritisation and Quality of Service (QoS) for individual flows. The issue of assigning different levels of prioritization or QoS to particular services is not dealt with in this work, but remains as a separate item of research.

1.3 Technological issues

A number of different technological solutions have recently emerged. These solutions include Ipsilon's flow based IP Switching (Newman *et al.* 1996b, Newman *et al.* 1997), Cisco's topology based Tag Switching (Rekhter 1997), Toshiba's Cell Switch Router (CSR) (Katsube *et al.* 1996, Katsube *et al.* 1997, Esaki *et al.* 1997) including both the Flow based and the Topology based approach. Other suggestions include Telecom Finland's Switching IP through ATM (SITA) and IBM's Aggregate Route Based IP switch (ARIS). Also the Internet Engineering Task Force's (IETF) Multiprotocol Label Switching (MPLS) workgroup is actively pursuing the subject, with several internet drafts available. In addition, the ATM Forum's MPOA (MPOA 1997) can be viewed as a kind of IP switching solution, although MPOA is argued to be burdened by the heavy load induced by the UNI signalling. The emerged technological solutions, although slightly different from each other, all aim to offer the flexibility of routing combined with the speed, and possibility for QoS, of asynchronous transfer mode (ATM) switching.

The main difference between these solutions, in addition to technical issues and actual implementation, is the way these approaches deal with different levels of traffic aggregation. For instance, Ipsilon's IP switching uses a very fine level traffic granularity defining traffic flows at the finest level of practical use: IP address and TCP port quadruples. On the other hand, for instance Cisco's Tag switching concentrates more on defining and setting up connections based on routing information and aggregated traffic flows from different parts of the network. The MPLS workgroup is currently including both previous views in its plans to introduce IP switching to the Internet. In this article, we concentrate on traffic based flow switching at IP address pair and IP+TCP address/port pair level.

1.4 Processing cost of traffic based IP switching

In IP switching, a connection is established for a selected number of flows, that are expected to have enough packets to be carried, so that the cost of the connection is lower than the cost of the routing decisions of individual packets.

Newman *et al.* (1996a) propose an approach where connections are established for certain types of flows, or protocols. We call this kind of criteria *static*. The *dynamic* criteria are based on measuring the packet flows against some criteria to decide when to establish a connection. The decision criteria are called *classifiers*. Lin and McKeown (1997) compare several classifiers, namely *X/Y Classifier*, *Protocol Classifier* and *Port Classifier*.

The *Port Classifier* is a static classifier based on the assumption that certain types of flows, such as telnet (rlogin) connections are probably longlasting, so a connection is established when the first packet of that kind of connection arrives at the IP switch. The *Protocol Classifier* is also a static classifier that establishes connections on a coarser granularity level (IP address pair). This approach results in lower number of established connections since several fine granularity level flows are multiplexed to a single connection. The *X/Y Classifier*, presented by Lin and McKeown (1997), is a dynamic classifier that examines the flow and makes a prediction on the future behaviour of the flow according to the past. It requires that X packets arrive in Y seconds. An X/Y classifier with $Y \to \infty$ is called a *Packet Count Classifier*, which has also been studied in (Newman *et al.* 1996a, Ilvesmäki *et al.* 1997).

In (Che *et al.* 1997) a dynamic classifier in which the classification criteria are updated based on the resource usage of the IP switch is presented, and in (Ilvesmäki *et al.* 1998) the classification criteria are taught and updated to the system by using a neural network classifier.

Some work has been made in simulating IP switching, see (Lin and McKeown 1997, Ilvesmäki *et al.* 1997, Ilvesmäki and Luoma 1997). The work shows that significant performance gains can be realized using IP switching techniques. In this paper, we propose a *Nondeterministic Classifier* to derive a theoretical value characterizing the maximum attainable gain in processing cost available from IP switching. The *Nondeterministic Classifier* is a classifier that always "guesses" the right answer to the question "To switch or to route" before packets of a flow start to arrive to a node. This way, the classifier is optimal: real life classifiers always lose in processing cost since at least the first packet needs to be inspected and routed regardless of the later decision of switching.

Since our purpose is to illustrate the behaviour of the classifiers and to present a classifier comparison scheme rather than to model traffic, we do not use analytic expressions for the probability distributions associated to the IP flows, but we have used real traces. The traces are drawn from three different types of networks.

In section 2 we develop a model for the nondeterministic classifier. In section 3 we refer shortly to real life classifiers and we use the developed model and real life traces to show the maximal attainable performance gains of IP switching compared to the optimal conditions for port and packet count classifiers. Section 4 gives us a brief summary of the results.

2 PROCESSING COST MODEL

In this section, we develop the model for our nondeterministic classifier. A nondeterministic classifier is a theoretical construct that always makes optimal classification decisions, even before seeing the first packet of the flow. We first list our assumptions, then describe the processing cost calculation and finally present the nondeterministic classifier.

2.1 Assumptions

In this paper, we limit ourselves to the case where the decision on connection establishment does not affect the traffic demand, for the simplicity of the model. If TCP timeouts would be tight, this might not be the case, since routing — taking more time than switching — might introduce longer queueing delays that result in misordered or resent packets.

Another assumption taken is that packet labelling, i.e. assigning incoming packets in the already established connections, on the edge of the IP switching network takes approximately the same processing cost as packet routing. We assume that the connection establishment decision is made according to the same criterion on the route of the flow, and that connection establishment takes approximately the same processing cost independently of the initiating node of the establishment in the network. With these assumptions, we may conclude, that by optimizing the processing cost of a single node inside the network, we optimize the whole network behaviour.

The classification of the packets requires processing cost that can be associated in the routing cost of the packets. Another possibility is to assume that flow classification neither with the nondeterministic classifier nor with any real life classifier does require processing capacity. Since our goal is to derive a practical lower bound for processing cost, this assumption is safe.

The background processing load of an IP switch, such as keeping up the links and logging events, is assumed not to depend on the routing or switching decisions. We also assume that flows through our IP switch are independent, which leads us to the conclusion that by minimizing the processing cost of all separate flows results in minimisation of the processing cost of the whole switch.

The number of connections that the network node is able to support is limited. The number of simultaneous connections is in some networks, such as an office's LAN, fairly small, and in some other networks, such as backbone networks of large systems rather high (Lin and McKeown 1997). We assume that the connection table in our systems is sufficiently large. Again, this assumption is safe: if a connection cannot be established when needed due to insufficient resources, total processing cost increases.

We assume that the processing cost of a connection is independent of the state of the system and the length of the flow. I.e. we do not explicitly consider the cost of maintaining the connection, and assume that the results are not affected significantly by this. Note further that omitting the connection maintenance cost lowers the calculated cost which further stresses the performance bound nature of our calculations. Thus our calculations represent the minimum cost attainable for the system in that sense.

2.2 Model

Let $\overline{c}_p$ denote the expected cost of routing one packet, and $\overline{n}_p$ the expected number of packets in each flow. Let $\overline{c}_c$ denote the expected cost for connection establishment. Note that packets on an already established connection do not incur processing cost.

Let

$$c = \frac{\overline{c}_c}{\overline{c}_p} \tag{1}$$

denote the relative cost of establishing a connection for a flow to the cost of routing each packet. For example, if we would say "Establishing a connection is 15 times as expensive for the processor as routing a packet", we would choose $c = 15$.

The processing cost is evaluated using traces. Let n_r denote the number of packets in the trace that are routed, and n_c the number of connections established. The values of n_r and n_c depend on the decisions made by the classifier used. The normalized processing cost is then calculated as follows:

$$C = \frac{1}{n_P}\left(n_r + c \cdot n_c\right), \tag{2}$$

where n_P is the total number of packets in the trace. The processing cost is normalized with n_P to give more easily understandable results.

2.3 Nondeterministic classifier

A nondeterministic classifier is a classifier that knows in advance how many packets would be carried on a flow. I.e., it guesses always correctly (minimizing the processing cost) whether a flow should be switched or routed. We are not able to build such classifiers. All classifiers behave worse than this classifier, thus our nondeterministic classifier represents a goal of performance gain towards which we may compare our classifiers. We also may check whether for our traffic the performance gain attainable by IP switching would be feasible.

When we want to minimize the processing cost of an IP switch, we would establish a connection for all flows, for which $\overline{c}_c < c_r$, where c_r is the processing cost for routing all the packets of the flow, i.e. $c_r = n_p \cdot \overline{c}_p$, where n_p denotes the number of packets in the flow. Thus, the lowest processing cost is achieved when $c < n_p$, or the flows that have more packets than $\lfloor c \rfloor$ are switched and other flows are routed. Here, $\lfloor c \rfloor$ denotes the greatest integer smaller than or equal to c. Finally, let $f(n)$ be the discrete probability distribution function for the flow length: $f(n) = P$["the flow is n packets long"]. The normalized processing cost C from equation (2) when nondeterministic classifier is used is then

$$C = \frac{1}{n_P}\left(\sum_{n=0}^{\lfloor c \rfloor} n f(n) + c \sum_{n=\lfloor c \rfloor + 1}^{\infty} f(n)\right), \tag{3}$$

where n_P is the total number of packets in the trace.

3 REAL LIFE CLASSIFIERS

The static classifiers are based on a predefined set of attributes, such as IP address and port pair. When a packet with matching attributes arrives, the connection is es-

Table 1 Traces used in this work

Name	*Location*	*Length*	*Nr of packets*	*Media*	*Other information*
tct	HUT/LAN 17.6.1997 9.50am	1 hr	142968	Ethernet	-
ebb	HUT/Campus Area Network 29.5.1997 9.00am	1 hr	1107188	10 Ethernet	-
dec	Digital's primary Internet access point 10:00, Thu March 9th, 1995	-	4086848	-	WWW-archive*

*http://ita.ee.lbl.gov/html/contrib/DEC-PKT.html

tablished. This means that we shift from the flow per flow classification done by the nondeterministic classifier to flowtype per flowtype classification, which introduces performance loss. Also, the first packet needs processing effort which further downgrades performance.

The dynamic classifiers are based on measuring IP flows and making decisions on connection establishment based on measurement information. Since measurement and routing compete for the same processing capacity, we need to have simple measurement data. This kind of data is e.g. the number of packets sent on a flow, or the number of packets sent in Y last seconds.

To establish a dynamic classifier, we need to find a condition that describes the state of the traffic process, and is easily measured in a real life IP switch. When the classifier notes that the condition is true, it establishes a connection. For packet count classifier, the condition is defined as $n_a > n_x$ where n_a denotes the number of packets received on the flow and n_x a threshold value. The packet count classifier is based on the assumption that the probability distribution of packet number in the flows is heavy tailed.

We investigated the behaviour of traffic based IP switching using our nondeterministic classifier and two real life classifiers: the port classifier and the packet count classifier. While earlier studies, see e.g. Lin and McKeown (1997), have used predetermined values for the classifiers, we estimated the optimal values for the classifier parameters from traces. Three traces were used (table 1). The traces represent a small IP Local Area Network (tct), a department wide Campus Area Network (ebb) and a major Internet access point (dec). Teardown time of 60s was used in all calculations.

The performance of the classifiers was analyzed using values of cost ratio c from 1 to 100. The performance of the nondeterministic classifier was analysed adding the

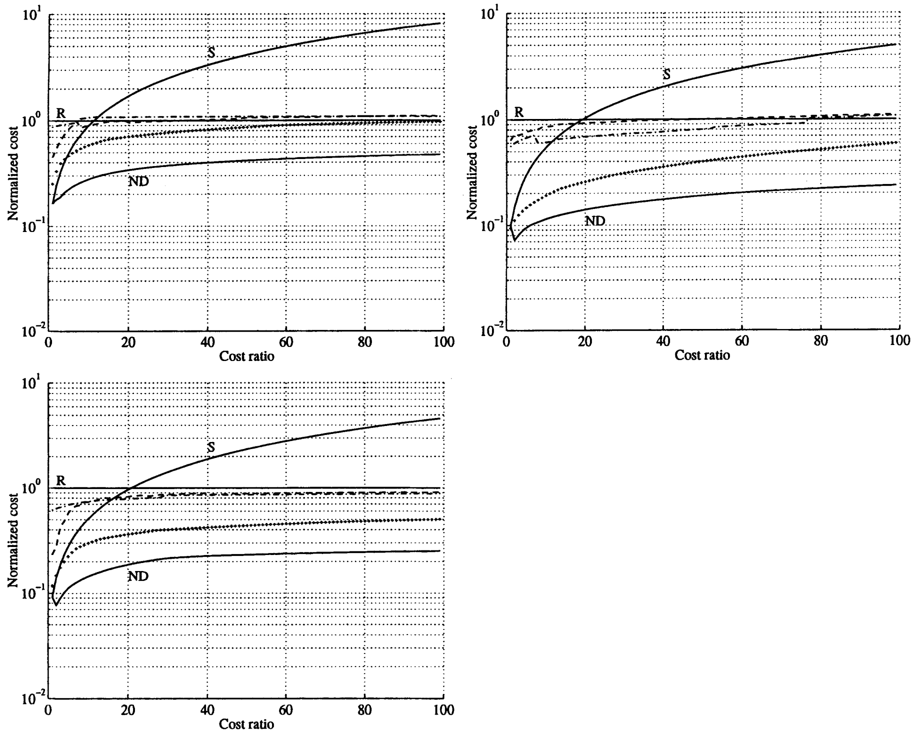

Figure 1 Normalized processor cost C with port classifier. Top left: dec trace, top right: tct trace, bottom: ebb trace. Y axes: the total normalized processing cost C, X axes: the cost of connection establishment vs. cost of packet forwarding, c. The line R represents the cost of routing all packets, line S the cost of establishing a connection for each flow, and the line ND the cost for the nondeterministic classifier. ··· the cost for the port classifier with port list created by the corresponding trace, and — · —, — — the costs for the port classifiers with port list created with other traces.

routing cost for all the packets for flows that have less packets than the value of c and connection establishment cost for all other flows.

Three real life classifiers of each type were created off line using one trace as input for classifier parameter estimation. The resulting classifiers were evaluated using all three traces in turn as input, and the results are presented in figures 1 and 2.

Figures 1 and 2 show the lowest achievable normalized processing cost as the cost for the nondeterministic classifier. IP switching seems to offer even 90 % reduction to the routing cost as an upper limit. Even if the cost ratio c grows to high, maybe even slightly unrealistic, values a cost reduction of 50 to 60 % is possible in theory.

The port classifier parameters are estimated by grouping together all flows with the same IP address pairs and source port numbers. The mean remaining flow length after receiving the first packet is calculated for each group. For the flow groups that

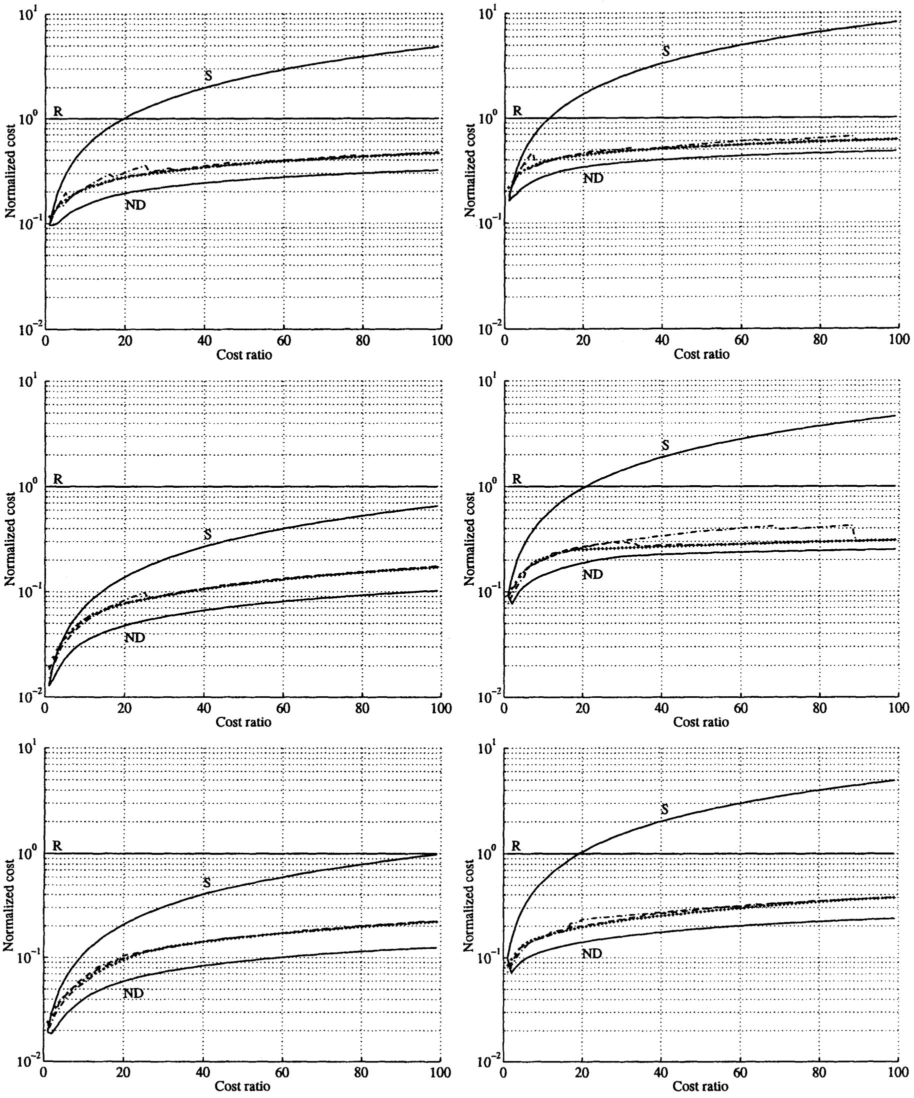

Figure 2 Normalized processor cost C with packet count classifier. Top: dec trace, middle: tct trace, bottom: ebb trace. Left: flow granularity at IP address level, right: flow granularity at IP address and port level. Y axes: the total normalized processing cost C, X axes: the cost of connection establishment vs. cost of packet routing, c. The line R represents the cost of routing all packets, line S the cost of establishing a connection for each flow, and the line ND the cost for the nondeterministic classifier. $\cdots$ cost for the packet count classifier with parameter n_a estimated from the corresponding trace, and — · —, — — the costs for the packet count classifiers with n_a estimated from other traces.

have the mean flow length greater than the value of the cost ratio c, the classifier should establish a connection for all flows belonging to the group. For the flow groups that have the mean flow length smaller than the value of the cost ratio c, all packets of the flows in the group should be routed. Thus the required parameters for the port classifier is the list of the port pairs port pairs and source and destination address pairs for which connections are established. After it has been found the classifier is ready for classification. When distinguishing flows, we used IP address pairs, and source and destination port pairs. Figure 1 shows the results for port classifier.

The real life port classifier seems to be capable at the best to only 50 to 60 % improvements. Figure 1 shows that the port classifier performs adequately only in the same or very similar network than with which it has been created. When the classifier is used in another network its performance degrades significantly (for our traces, to only 10 to 30 % improvements). The processing cost may in some cases even be higher than for pure routing. In different parts of the network there exist different kinds of traffic profiles. Consequently, when using a port classifier one needs either to determine the switched ports separately for each network or to use a system that can adapt to the changes in the traffic profile. More generally, it can be said that implementing any kind of consistent Quality of Service on a connection to the Internet is difficult, since different services require different handling on different parts of the network.

The parameter for the packet count classifier is estimated by calculating the processing cost if n_a first packets of the flows would be received and after that the connections would be established. Then the value of n_a for which the cost attains its minimum is chosen for the threshold value of the classifier, i.e. minimize

$$C = \frac{1}{n_P}\left(\sum_{n=0}^{n_a} nf(n) + \sum_{n=n_a+1}^{\infty} (n_a+c)f(n)\right), \tag{4}$$

subject to $n_a \geq 1$, where $f(n)$ is the flow length distribution. The parameter attained is the threshold value n_a. When working, the classifier decides to establish a flow when the number of received packets exceeds the threshold value. We calculated the processing cost with two levels of granularity: first with IP address pairs, then with IP address pairs and source and destination port pairs. The optimal thresholds found are shown in figure 3 and the costs in figure 2

Figure 3 shows that as the cost ratio grows the packet count threshold value also grows. This is expected, since the higher the packet count threshold value is the less connections are set up. This behavior minimizes quite effectively the cost of packet count classifier as seen in figure 2. The cost of packet count classifier follows the nondeterministic classifier cost with a slightly worse performance. The maximum cost reductions with the packet count classifier are in the 90 % level, and even for high practical cost ratios the classifier has relatively reasonable levels of processing cost reduction (about 40 to 70 %). The results also show that the cost reduction is rather insensitive to the selection of the packet count threshold value, which suggests that the same packet count classifier can be used in a variety of networks.

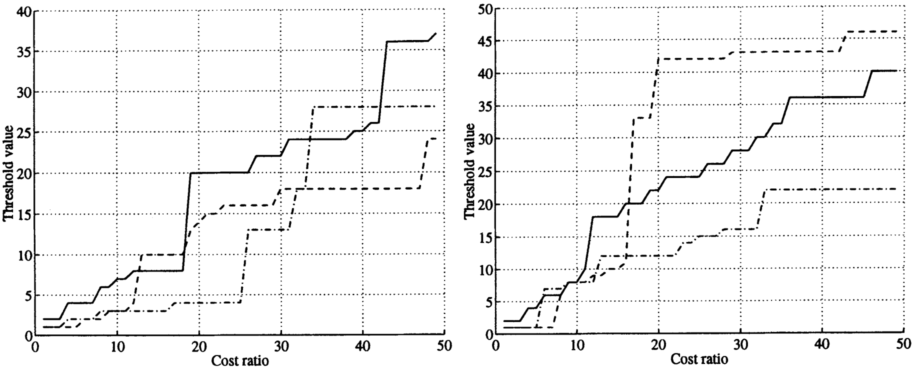

Figure 3 Optimal threshold values used for packet count classifier. Left: IP address level granularity, right: IP address and port level granularity. – dec trace, – – ebb trace, – · – tct trace.

4 CONCLUSIONS

The flow classifier has a significant impact on IP switching system performance. In earlier works different classifiers were compared but no bounds were present for the maximal performance gains available. In this paper, we proposed a nondeterministic flow classifier for performance evaluation of flow based IP switching solutions. The nondeterministic classifier is an optimal classifier with which other classifiers can be compared.

As an example, we evaluated the performance of two real life classifiers by a comparison to the nondeterministic classifier. The results show that even 90 % of processing cost reductions may be achieved by flow based IP switching in theory. The practical classifiers perform worse.

The port classifier offers moderate cost reductions but is heavily dependent of the underlying traffic. The packet count classifier offers a good performance and is less sensitive to the underlying traffic. Neither of these classifiers supports implementing QoS classes for flows well.

We claim that the nondeterministic classifier based performance comparison is widely applicable and relatively easy to implement. Furthermore, it gives comprehensible results.

ACKNOWLEDGEMENTS

The authors wish to thank Prof. Jorma Virtamo and Prof. Gunnar Karlsson for their comments on the early stage of the work. The research was funded by TOVE and IPANA projects at the Helsinki University of Technology.

MPOA (1997) *Multiprotocol over ATM Version 1.0.* ATM Forum, 07.04.1997.

Che, H., Li, S.-Q., and Lin, A. (1997) Adaptive resource management for IP/ATM hybrid switching systems. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 328–339.

Claffy, K.C. (1994) *Internet Traffic Characterisation.* Ph.D. thesis, University of California, San Diego.

Claffy, K.C., Braun, H.-W., and Polyzos, G.C. (1995) A parametrizable method for internet traffic flow profiling. *IEEE Journal on Selected Areas in Communications* **13**(8) 1481–1494.

Esaki, H., Matsuzawa, S., Mogi, A., Ichi Ngami, K., Jinmei, T., Kon'no, T., Katsube, Y. (1997) Cell switch router (csr) – label switching router supporting standard atm interfaces. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 2–10.

Ilvesmäki, M., Kilkki, K., and Luoma, M. (1997) Packets or ports – the decisions of IP switching. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 53–64.

Ilvesmäki, M., and Luoma, M., (1997) IP switching in a simplified ATM environment. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 65–76.

Ilvesmäki, M., Luoma, M., and Kantola, R. (1998) Flow classification schemes in traffic based multi-layer IP switching – comparison between conventional and neural approach. Accepted to *Computer Communications* **22**.

Katsube, Y., Nagami, K.-I., and Esaki, H. (1996) *Cell switch router – Basic concept and migration scenario.* Toshiba R&D Center.

Katsube, Y., Nagami, K., and Esaki, H. (1997) *Toshiba's router architecture extensions for ATM: Overview.* Toshiba R&D Center.

Lin, S., and McKeown, N. (1997) A simulation study of IP switching. In *Proceedings of ACM SIGCOMM* September 1997.

Newman, P., Lyon, T., and Minshall, G. (1996a) Flow labelled IP: A connectionless approach to ATM. In *IEEE INFOCOM Joint Conference on Computer Communications* San Francisco, California, March 1996, **3**, 1251–1260.

Newman, P., Lyon, T., and Minshall, G. (1996b) Flow labelled IP: Connectionless ATM under IP. In *Networld & Interop,* Las Vegas, April 1996.

Newman, P., Minshall, G., and Lyon, T. (1997) IP switching: ATM under IP. *www.ipsilon.com,* January 1997.

Rekhter, Y. (1997) Tag switching architecture – overview. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 11–19.