# Advanced IP classification techniques for a hybrid routing node in a DWDM metropolitan network

Carlos Macián[1], Peter Domschitz[2]
macian@ind.uni-stuttgart.de, pdomsch@rcs.sel.de

## Abstract

Although the progress in the optical technologies will put an unheard-of transmission capacity at the user´s disposal in the near future, to fully take advantage of that bandwidth the nodes in the network have to be able to provide QoS and Value Adding Services (VAS) to those same users at wire-speed. One of the most critical aspects, which is being researched inside the high-speed routing activities within the GIGA-RING project (itself part of the german initiative KomNet), is packet classification in a hybrid routing node. In this paper, we present a novel approach that overcomes the scalability issues of this question by means of a hierarchy of filter databases based on the semantics of the rules. We also demonstrate how our model helps the nodes support QoS and VAS with an example of outsourcing Virtual Private Network (VPN) services.

## 1. Introduction

Within the German initiative KomNet (Innovative Communication Networks for the Future Communication Society, see [1]), a research project called GIGA-RING has been established to develop and demonstrate the benefits of future broadband access networks. Metropolitan area networks (MANs) are designed to close the gap between an optical core network and a flexible scenery of last-mile networks. Wavelength division multiplex (WDM) technologies are used for transmission as well as for routing among the access systems attached to the MAN via optical access nodes. A ring-formed optical transport network with hybrid routing nodes for the metropolitan area has been designed and set up, and will be exhaustively tested within a field trial. The GIGA-RING project consists of two activities: One part is optic-oriented, focusing on the research and demonstration of the potential of new architectural and management aspects for DWDM (Dense Wavelength Division Multiplex) optical access networks. Subject of the other project activity is an innovative, universal but cost efficient electronic routing node, which is able to process the bandwidth offered by the DWDM-based metropolitan network. The electronic switching and routing functions are projected for data rates up to some Gbps, covering all network protocol layers up to the ATM and IP protocol level. Therefore, this hybrid routing node is designed for a flexible interconnection of existing and future broadband last-mile networks using various technologies. This paper is focusing on the advanced, QoS-oriented IP functions of the node, including support for virtual private networks (VPNs).

The immense bandwidth that the DWDM technology promises to deliver cannot be fully taken advantage of without network nodes that are able to process packets at that same speed. Ideally, those nodes should perform their task without abandoning the optical domain, but although there have been constant improvements aiming thereto, it seems that hybrid routing nodes, performing part of the packet processing optically and part electronically are still the only viable alternative in the immediate future. In recent years the design of routing nodes suffered a strong revolution, with researchers as well as manufacturers swifting towards more distributed approaches which take the processing burden out of a central unit and distribute it among the network cards [1], [3], [4]. That, together with the substitution

1.Corresponding author. Postal address: University of Stuttgart, Institute of Communication Networks and Computer Engineering (IND), Pfaffenwaldring 47, D-70569 Stuttgart, Germany.
2.Alcatel Corporate Research Center, Stuttgart, Germany.

of the bus as switching plane against solutions more amendable to parallel transmission of packets or cells, like shared memory or crossbar switches, have taken the routers to the point of delivering throughputs of hundreds of Gbps and beyond (see [5] and [6] for just two examples).

Existing implementations typically achieve high speeds at the cost of flexibility, which makes them optimal for the routing of best-effort traffic, but stretches them to their limits as soon as QoS or Value Adding Services (VAS), like Virtual Private Networks (VPNs), come into play. One of the tasks which more strongly unveil this fact is the classification of packets. Since the first step towards being able to deliver QoS or VAS is the ability to discriminate the packets to be able to serve them differently, it is crucial that routers can perform classification at wire-speed. In this paper we will describe a novel approach to this problem, which shall enable routers to classify packets based on any number of header fields at high speeds. We will furthermore show how such a design can improve the flexibility of the nodes and help them to provide VAS. For the sake of brevity, we will restrict ourselves to the description of the main properties of our model, deferring most details and numerical results for a future paper.

The remainder of the paper is structured as follows: Section II presents our hybrid routing node; section III describes the packet classification problem and the previous work done to solve it; section IV presents our model and analyses its properties, and section V concludes the paper by providing a summary and looking into the future of our research.

## 2. The GIGA-RING hybrid routing node

A global trend is the increasing demand for fast IP networks in all environments, but specially in the access area, where a strong backlog exists. New technologies like WDM have had little impact in this segment so far, but that is set to change: The aggregate traffic from emerging high-speed last-mile infrastructures (xDSL, HFC, WLL, ...) calls for powerful optical technologies.

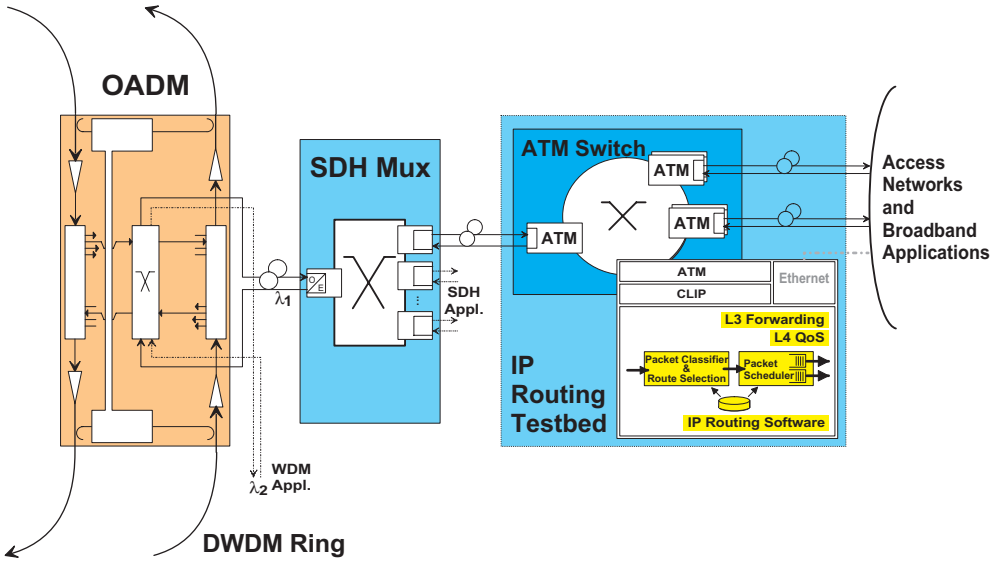



**Figure 1:** The GIGA-RING hybrid routing node

Hybrid opto-electronic routing nodes are expected to be key elements of future MANs. The GIGA-RING research activity addresses this challenge by analysing the overall system requirements, structures and concepts without focusing on a specific implementation of functional models. Rather, it is intended to point the way for future system developments.

Until mid 2000 a first demonstrator for the hybrid routing node will be supplied, combining an optical add/drop multiplexer (OADM, in quasi-fixed configuration with protection) and an electronic gigabit

switching node, whose fundamental structure is depicted in Fig. 1.

The electronic part is characterized by the function blocks of the ATM and particularly the IP layer. The SDH multiplexer works as a connecting link, offering synchronous data streams extracted from a wavelength channel selected by the OADM to the higher layers. The architecture of the switch/router is designed for high performance and optimized for the interoperation of ATM and IP, following the idea of establishing a flexible platform for the prototypical implementation and examination of advanced concepts and protocols in the metropolitan area. These new concepts will be evaluated in a field trial involving fast IP applications running over an ATM core delivered to real users.

## 3. The packet classification problem

The packet classification problem, aiming at deciding what kind of treatment a packet deserves based on a corpus of filters or rules, is a well-known multidimensional matching problem. In a way, it can be seen as an extension of the IP Routing Lookup problem, in which the treatment of the packet (simply, to which output port to send it) is decided by looking at just one header field (the destination address) and looking for the matching rule (the longest matching prefix) in a destination prefix database. That much more simple operation was supposed to be one of the main bottlenecks in the new router architectures, since it is a very time-consuming task. Recently, new matching algorithms have been proposed which solve or at least leverage that problem (see for example [13] for a good survey on the subject).

In today´s routers, the broader question of packet classification was seen as a minor problem, because the filtering was performed usually only for security purposes (firewalling), which contain only a small number of rules to permit communication with known partners, and blocking all the rest. Classical search algorithms are run over that filter database every time a packet arrives, to see if it is conform with the security policy. The problem is, those algorithms do not scale well with the number of rules and/or with the length of the header segment to compare, and both variables can be expected to increase strongly in the near future.

The delivery of QoS or VAS in the IP world is centred around the concept of *flow*. If we assume that the proportion of traffic requiring QoS is going to grow strongly in the future, mainly due to the broad dissemination of real-time multimedia applications and the willingness of the people to pay a little bit more for a better service, the importance and the difficulty of distinguishing and separating flows to deliver individual QoS guarantees will greatly increase. This is the approach taken by the Integrated Services Architecture [7].

Another proposal, Differentiates Services, tries to overcome the scalability question that arises in Integrated Services by aggregating traffic in a reduced set of classes [8]. But even in that case, there are points in the network (the edges) where the (de-) aggregation of the traffic has to take place, involving once again a potentially huge number of individual flows which have to be individually classified.

Furthermore, the inclusion of VAS like VPNs (and, in the future, QoS-enabled VPNs) makes it even more difficult: Think of a scenario in which a customer outsources its VPN to its Internet Service Provider (ISP). The ISP has to guarantee that the traffic of different VPNs will not interfere, and possibly deliver different degrees of QoS to the different traffic types (real-time, non-real-time, ...) that every client might define. In such a case it might be necessary to very accurately analyse the content of the packet header to decide first to which VPN it belongs, and then to which traffic class and priority level.

The number of rules or filters needed in the routers for classification is directly proportional to the number of flows to be considered. Add to the increasing number of rules that the identification of a typical flow could require the analysis of its destination address, source address, source port, destination port and Type Of Service (TOS) Byte, making a total of some 102 bits spanning five header fields. Not all the algorithms can cope with such a broad key to match. Clearly classification algorithms have to scale well both with the number of rules and the length of the key (i.e., the header chunk to be matched).

We do not claim the size of the filter database to be going to grow unboundedly in the near future, but we do claim that current solutions can not cope with realistic scenarios, which are not that far away. Recall the previous example: Let´s say that the ISP has 20 VPNs to support. Every VPN has defined 200 addresses or prefixes spread around the globe with which a communication shall be allowed (other delegations, partners, providers, etc.). Every client wants at least three different levels of QoS (say, for real-time, non-real-time and best-effort traffic). For security reasons, they set up different rules for different services (HTTP-Access, FTP and the like). Say, 10 different services. That makes 20*200*3*10 = 120,000 rules, which even for this quite restricted example is already well two orders of magnitude over actual databases [9]. Notice that this problem is independent of the way the (possibly aggregated) traffic is transported over the network, since it still has to be classified and separated at the edges.

Since the classification problem is the multidimensional case of the well-studied IP Address Lookup problem, the most common approach in the literature has been to try to extend the solutions for the former to the later. Most proposals use evolutions of the binary tree search algorithm over the whole length of the key, like the grids-of-tries [11]. Others have try to apply recursivity to the problem, like in [9] or [10]. Others try to decompose the multidimensional problem in *K* unidimensional ones, where *K* is the number of header fields to be matched, and then try to find a filter that matches the concatenation of the results of the individual classifications [11]. A geometric approach has also been taken in [12].

In all the cases there are three main variables to consider to judge the quality of a classification algorithm: The speed at which a matching filter is found (in the average as well as in the worst case), the memory requirements of the chosen solution and the time that it takes to make an update in that structure (i.e., to add or remove a filter or rule).

To the best of our knowledge, none of the afore mentioned solutions can minimize all three variables. Most of them try to find a good compromise of the first two at the cost of very expensive updates, with some rare exceptions (see [13]). In our opinion, that is a dangerous approach, because, as we asserted already, the basic unit to be classified is the flow, and a flow is intrinsically dynamic. Recent measurements show that it is not unusual for any core router today to support 100K flows simultaneously at any one moment, with update rates of several hundred flows per second. If we envision such orders of magnitude as the possible evolution of edge routers in the future, we clearly see that to support fast update rates is mandatory. In our proposal, we strongly address this issue by trying to concentrate related rules in contiguous positions in our data structure, a feature that we name the ***locality*** of the rules.

Most of the proposals neglect that locality and disperse related rules over the whole data structure, or even replicate them in various positions. We believe that the cause is that up to now no attention was paid to the ***semantics*** of the rules. The diverse rules are not independent from each other, but grouped in blocks with „similar meanings“, i.e. which address a same function: Enforcing a security policy, discriminating a certain service (HTTP-Access, for example), differentiating among VPNs, etc. That is to say, we could take advantage of that semantic relationship among rules -which is purely logical- and transform it in a physical relationship by locating them together in our classifier. Furthermore, this semantic grouping is recursive: There is a subgroup of rules which address a certain service for traffic corresponding to a certain VPN, etc. (see Fig. 2). Said another way: There is a natural ***hierarchy*** concerning the rules in a packet classifier, which has never been taken into account before.

Another important point related to this is that not all packet headers give the same amount of information concerning the group the arriving packet belongs to, and the cost to retrieve that information is also different. We call this property the ***discrimination capability*** of every header field, for it tells us how useful this header is to distinguish or discriminate among the various groups the packet could belong to and how high is the cost of doing so. Obviously, the discrimination capability varies depending on the scenario. For example, recall the VPN outsourcing service provided by a certain ISP that we mentioned before. In that environment, the Source Address of a packet delivers an almost unambiguous information about its VPN, which the Destination Address might not so easily do. Furthermore, the Source Address provides that information with a much simpler operation, since the number of possible source addresses (20 in our example, if every VPN has only an address prefix) is naturally in an edge node much smaller than the number of possible destination addresses (200 * 20 = 4,000 in the example).

By paying attention to the semantics of the rules we can target a hierarchy of groups of rules, which has the field with the highest discrimination capability at the root, the second highest in a further step, and so on. Although, as we said, this hierarchy and the fields which will be considered at every step are application-dependent, we believe that a near-optimal combination can be pre-computed at boot time by analysing the structure of the initial filter set. Its exact content will change due to the updates, but it is reasonable to assume that its main characteristics (like the number of VPNs, or the number of permitted destinations) will not be severely affected, assuring the resiliency of the chosen structure with time. In this paper we will nevertheless centre only on VPN-support, which we think is a very important scenario nowadays, bringing together the two principal novelties of future networks: QoS and VAS.

## 4. The hierarchical model

A question which always arises when designing a packet classifier is the choice between a hardware or a software implementation. Although software implementations have classical advantages, like the flexibility or the re-programmability, to achieve the throughputs that are to be expected, it is our opinion that only hardware solutions are viable. Moreover, the idea of hierarchy introduced on the previous section is highly amendable to pipelining, as we will see. An adequate design also permits to take advantage of another classical feature of hardware implementations: the use of parallelism. Hence, we decided to optimize the design for a hardware implementation from an early stage.

A rough representation of our model can be seen in Fig. 2. In the presented example, a hierarchy of

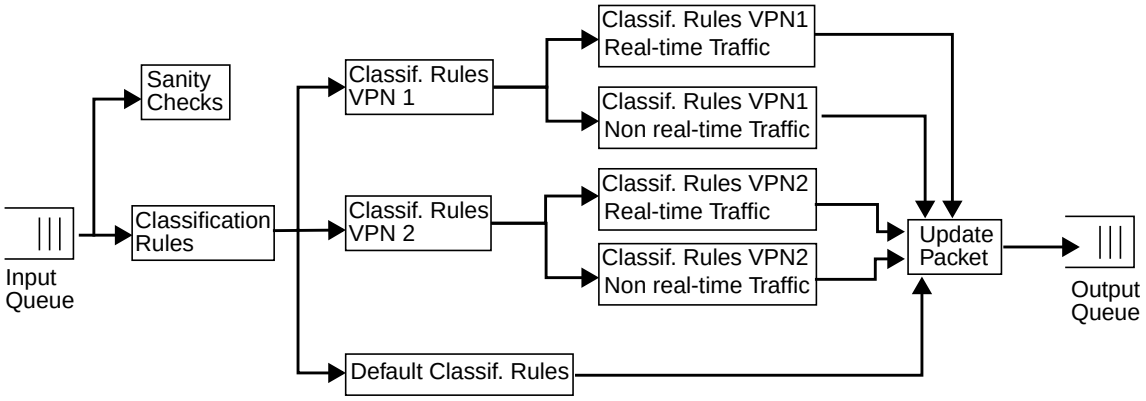



**Figure 2:** Example of a 3-step hierarchical classifier supporting 2 VPNs plus default traffic

three steps has been depicted for a simple scenario with just two VPNs and two types of traffic: real-time and non-real-time.

The fact that the classification shall be performed at the input of a router to avoid wasting valuable resources (like bandwidth in the switching backplane or space in the output queues), permits the coupling of two independent functions inside the packet classifier: The verification of the correctness of the packet (the so-called „sanity check“[1]) and the classification itself. These two functions can easily be performed in parallel, serving the former as a validation for the later. Shall the packet not be considered “sane”, then independently of the classification result it shall be discarded.

Our approach consists in starting with a rough discrimination of the filter group whom the packet might match against, and progressively bound the subgroup with every further step. In other words: We establish our hierarchy along the line of maximum negative slope of discrimination capability. Ideally only

1.In fact, the sanity check performs various tests over the packet: That the (source, destination) address exists, that the time to live has not reached zero, that the length of the packet makes sense, etc.

one header field is checked every time, although the model itself does not preclude the use of two or more fields in any one step, if it is found to be more suitable. As an example, a combination of destination port and value of the TOS-Byte might serve to define the priority and service of the packet (Service: Real-time, Priority: Highest), and since the number of possibilities is tightly bounded (in our example, we supposed only 3 priorities and 10 services), it might be more efficient to couple them.

In every step only one field is examined, allowing the use of the new algorithms developed for the unidimensional case (IP Address Lookup[1]). The fields considered important for the classification are the five mentioned before: Source and destination addresses and ports, plus the TOS-Byte (or the DS-Field for Differentiated Services). The point in the hierarchy at which they are examined could depend on the concrete scenario. For our example, it is clear that the source address can almost unambiguously define the VPN and should therefore be used on the first step.

After a number of steps, depending on the characteristics of the concrete filter database, it will not be possible to efficiently further refine the classification. At this point, a list containing all the filters which correspond to this subgroup will be used to perform the last step of the classification. The packet is then confronted with the original multidimensional matching problem, albeit in a much more restricted context, i.e., the number of rules to compare against has been greatly reduced. In this way we overcome the scalability problems that most of the existing algorithms have when the size of the rules databases grows beyond some hundreds or thousands of elements.

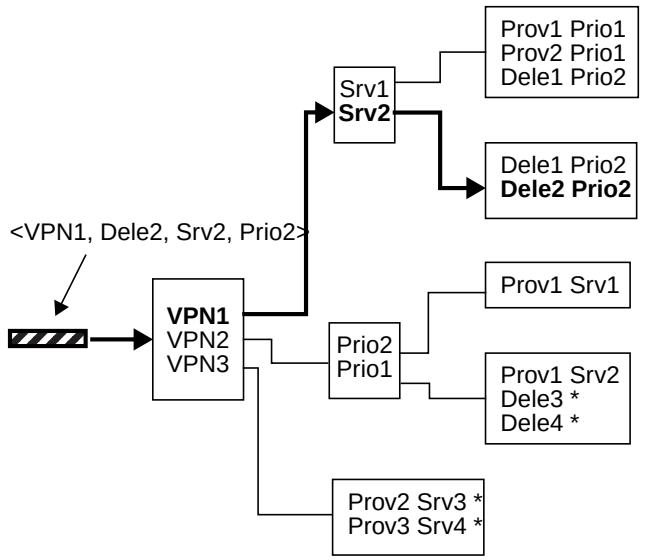
At every step, depending on the characteristics of the original database, a different matching algorithm can be used, to take advantage of their different properties. Some algorithms are better suited for longer keys, for example (like a combination of source address and source port: 32 + 16 = 48 bits), while others scale better with the number of rules, albeit only when the length of the key is kept small. At the same time, to achieve the high classification rates we aim at, we treat every step in our hierarchy as a logical pipeline. In this way we can perform a classification every clock cycle, independently of the algorithms used.

It should be noted that our structure situates the rules in its complete form (the combination of values for the five header fields) only at the end of the hierarchy. In the previous steps only a list containing the values of one of the fields is stored. Moreover, in that list only the values which actually are found in existing rules will be stored: If we define just three different priorities in the TOS-Byte, like in our example, it makes no sense to have a list with all the 255 possible values. By locating related rules together we make it simpler to trace where a new rule shall go (or an old one shall be removed from). This increase in the locality of the filters thus highly simplifies the update of the database: To add or remove a rule, only the corresponding small table at the end of one of the branches of the hierarchy must be updated, plus its ancestors, if the adding (or removing) of that rule introduces (or deletes) a value not yet present on them (a value that no remaining rule possesses). In this way, the update rate that can be achieved is highly improved.

Consider the example depicted in Fig. 3, which represents the classifier resulting from the rules in the table next to it. For clarity, we have used names instead of numbers for the different fields. We consider an scenario with three VPNs, seven possible destinations (distributed between providers -Prov- and delegations -Dele-), four different services and just two priorities. The chosen hierarchy derives from the analysis of the ruleset. As stated, the header field with the highest discrimination capability is the source address, being therefore placed at the root. The next steps are different for every VPN. If a packet defined by <VPN1, Dele2, Srv2, Prio2> arrives at the classifier, it will follow the highlighted path prior to achieve classification. At every point, the way to take will be selected by examining the chosen header field, except in the last step, where two fields have to be checked. The packet has been compared against much smaller tables, and in all cases but one only one dimension had to be checked.

Let’s consider the addition of the rule <VPN1, Dele5, Srv2, Prio1>. Since all the prefixes were already

1.Although a rule can be specified as: a range of values, a prefix, a suffix and an exact value, all of them can be transformed into prefixes, as explained for example in [11].




| Src Addr | Dst Addr | Dst Port | Priority |
|---|---|---|---|
| VPN1 | Prov1 | Srv1 | Prio1 |
| VPN1 | Prov2 | Srv1 | Prio1 |
| VPN1 | Dele1 | Srv1 | Prio2 |
| VPN1 | Dele1 | Srv2 | Prio1 |
| VPN1 | Dele2 | Srv2 | Prio2 |
| VPN1 | Dele1 | Srv1 | Prio2 |
| VPN2 | Prov1 | Srv1 | Prio2 |
| VPN2 | Prov1 | Srv2 | Prio1 |
| VPN2 | Dele3 | * | Prio1 |
| VPN2 | Dele4 | * | Prio1 |
| VPN3 | Prov2 | Srv3 | * |
| VPN3 | Prov3 | Srv4 | * |

**Figure 3:** Classification example

present in the corresponding tables but Dele5, only the last table has to be updated as stated in Fig. 4. If, on the other hand, the filter <VPN2, Prov1, Srv1, Prio2> had to be removed, since there are no more rules corresponding to the value Prio2 for VPN2, the second and third tables in that branch have to be updated as expressed in Fig. 4. In fact, being the last table now empty, it is simply removed.

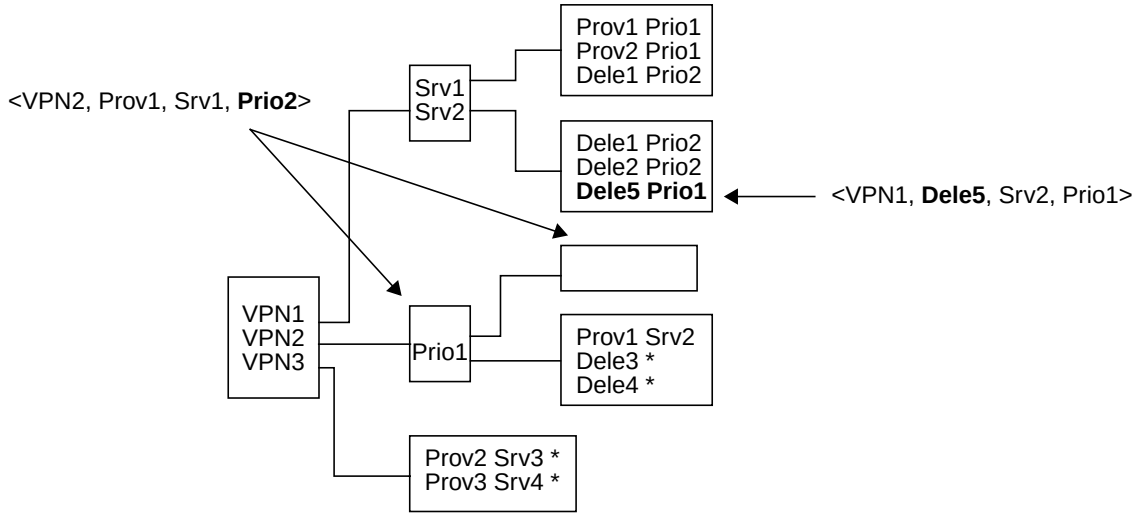



**Figure 4:** Addition and removal of filters

It is difficult to have access to real filter databases, and the ones which are available are far too small to serve as a testbed for our experiments. That is why for the validation of our model we had to create our own databases by analysing the contents of some smaller databases from the routers at the Institute of Communication Networks and Computer Engineering and extrapolating their contents to the sizes we expect to find in the near future. At the same time, we extended the functions of the filters to account for the introduction of emerging QoS-aware and value adding services in the networks.

At the moment of writing, we have started the measurements of our model by means of emulation in a software platform based on Linux as well as by simulation. First steps towards implementing a prototype have also been taken. As a most interesting scenario, we are considering the case depicted in this paper, where an ISP is offering VPN services to many companies at the edge of a core network. Several

configurations (depth and structure of the hierarchy, matching algorithms employed along it, etc.) are being tested. The results of those measurements, as well as a detailed description of our prototype and an analysis of the effects of the security issues typically present in VPNs (like encryption, authentication and tunnelling), which have also been taken account of in our model, are deferred for a future paper.

## 5. Conclusion and further work

The change of paradigm in the IP world towards the inclusion of new QoS-aware and value adding services presents severe challenges to the network nodes. One of the most critical examples is the classification of packets at wire-speed. The algorithms developed to solve this multidimensional matching problem have generally tried to achieve high-speed and an acceptable storage requirement at the cost of very low update ratios. In this paper we have presented a classifier which scales well thanks to its hierarchical approach based on the semantics of the rules; performs classification at high-speed through the use of a hardware implementation of the hierarchy, the assumption that fields with a better discrimination capability should be checked first, and by keeping tables small and generally unidimensional; and presents a favourable structure to achieve high update ratios by maximizing the locality of the rules in its tables. We have furthermore demonstrated its virtues in a VPN-scenario, for it brings together the two principal novelties of future IP networks: QoS and VAS.

The exhaustive testing of our model by means of simulation and emulation in software will give path to its implementation in hardware. Meanwhile we are working on new classification algorithms specially designed to take advantage of the characteristics of our model.

The described work has partially been funded by the German Federal Ministry for Education and Research (BMBF) through the project GIGA-RING (01 BP 815/3) within the KomNet program [1]. The authors alone are responsible for the content of the paper.

## References

[1] http://www.hhi.de/komnet

[2] Partdrige, C. et al.: A 50 Gb/s IP Router, IEEE/ACM Transactions on Networking, vol. 6, n . ß, June 1998

[3] Keshav, S. and Sharma, R.: Issues and trends in Router design, IEEE Communications Magazine, May 1998

[4] Kumar, V.P. et al.: Beyond best-effort: Router architectures for the differentiated services of tomorrow's Internet, IEEE Communications Magazine, May 1998

[5] NX64000, Nexabit Networks, http://www.nexabit.com

[6] SP2400, NEO Networks, http://www.neonetworks.com

[7] Herzog, S.: RSVP extensions for policy control, IETF RFC 2750, proposed standard, January 2000

[8] Blake, S. et al.: An architecture for differentiated services, IETF RFC 2475, informational, October 1998

[9] Gupta, P. and McKeown, N.: Packet Classification on Multiple Fields; SIGCOMM'99, Harvard University

[10] Gupta, P. and McKeown, N.: Packet Classification using Hierarchical Intelligent Cuttings; Hot Interconnects VII, August 1999, Stanford University

[11] Srinivasan, V. et al.: Fast and Scalable Layer Four Switching; SIGCOMM'98, Vancouver

[12] Lakshman, T. and Stiliadis, D.: High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching; SIGCOMM'98, Vancouver

[13] Tzeng, H. and Przygienda, T.: On fast address-lookup algorithms, JSAC, vol. 17, nr. 6, June 1999