# Virtual Routers Supporting Active Networking

Florian Baumgartner and Torsten Braun
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10, 3012 Berne, Switzerland
baumgart|braun@iam.unibe.ch

## Introduction

One of the central problems dealing with Active Networks is to prove the benefits of this new technology. This is in particular challenging when Active Networking is mixed up with other technologies as for example Differentiated Services. On one hand it is necessary to provide a sufficient number of nodes, being capable to support both Active Networking and Differentiated Services, on the other hand the number of available network equipment is limited and simulators do not provide the desired flexibility.

## Virtual Active Routers

To solve these problems and to support the easy evaluation and demonstration of new concepts, the idea of Virtual Active Routers (VAR) has been developed. These user space programs allow the emulation of a standard IP router with it's usual functionalities as traffic conditioning, Differentiated Services, advanced queueing systems, routing and Unix-like front end [BB00b]. In addition, these VARs offer an interface for an Active Networking platform.
Each VAR emulates an IP router and may be connected to other VARs on the same or on another host or to the real IP world. A large network topology may be emulated on a couple of hosts. This emulated network can forward real IP traffic and can be connected to a physically existing IP network (Figure 1). It makes no difference to applications whether a part of a network is emulated or real.

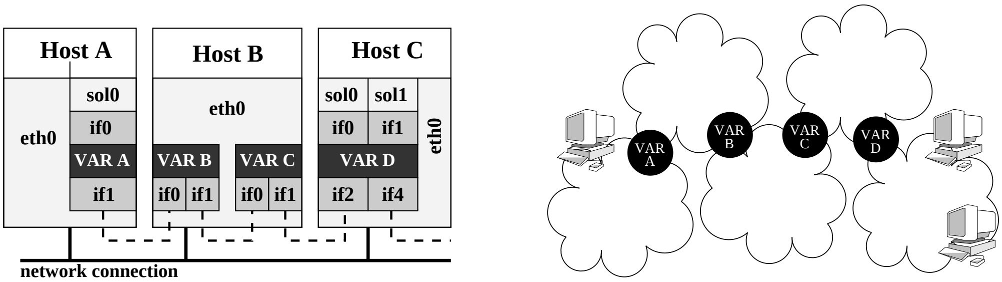


Figure 1: *Four VARs are set up on three hosts (left) and results in the emulated network shown on the right.*

An additional benefit is the possibility to set up several topologies in parallel. These topologies will not interfere as long they are not connected. In the combination with Active Networking the Virtual Router platform allows also to establish overlays on a real topologies, establishing some kind of abstract network as proposed in [BS99].

## Capsule Interpreter and Programmable Filters

For the development of QoS supporting mechanisms by active networking technology a prototypical Capsule Interpreter (CIP) was implemented in Tcl. Tcl was chosen because of its simplicity and flexibility. As a result of the currently ongoing work, this CIP will also be available for Linux. In that case, mixed topologies consisting of real Linux routers and VARs may be established.
The VAR provides as a central instance the programmable filter (PF) [BB00a]. A new module (e.g. the CIP) can notify this PF to forward specific packets to the module. For example, the CIP listens after startup to IP packets with the Router Alert option set and the received code may establish new filters to process certain packet types or flows. So, complete queueing systems can be applied, overriding the VAR's built-in packet processing. To decrease processing load the PF also provides some high level functionality. So, the bandwidth of a flow may be measured or an action performed, when a certain event occurs. For communication purposes between capsules and to provide a general mechanism to store information, the CIP provides a database allowing to specify which capsule gains which type of access to the information. This database is also used as an interface to query system states, information about the router etc.

## Mobile Agents, Reservation Domains and QoS

We are currently using the VAR technology to evaluate concepts for the management and provisioning of Quality of Service. One of the major problems providing QoS in the Internet are the different technologies and standards being used. In addition to conflicting principles (end to end vs. traffic aggregation) there may also be internal methods or special versions used by an ISP to reserve certain resources. Especially Differentiated Services allows the unrestricted use of DiffServ code points (DSCPs) within an ISP's network. Active Networking technology can be used to map reservation between different reservation technologies.
The network can be considered as being divided into several Reservation Domains (RD), supporting specific reservation types as shown on figure 2. As capsules have to be forwarded and executed withing a RD, the size of a RD may also be limited by access rights or the provided AN platform. In any case, the border of the RD is the point, where action has to take place. After injection of the code, the capsules are distributed by a kind of flooding algorithm and search for a compatible transition point by scanning the border routers. There they can supervise incoming reservations. As a typical case the mapping of an incoming RSVP reservation to a Differentiated Service traffic aggregate, as described in [BBBG00] shall be presented here. Another problem might be the mapping between different DSCPs.

RSVP flows shall be aggregated to a DiffServ tunnel. This requires the encapsulation and aggregation of the data at the ingress router, the configuration of intermediate routers, and the decapsulation at the egress point.
In the first step, a RSVP *path*-message is sent from the sender to the receiver in order to determine the data path. We use this *path*-message to trigger the initial tunnel setup. The *path*-message from host F (see figure 2) to host A is forwarded by E. The filter at E receives the message, sets up a tunnel start point, encapsulates the message in an AN capsule, and forwards it through the RD, configuring intermediate routers and setting up the tunnel endpoint in the egress outer B. Here, *path*-message is decapsulated and forwarded in it's original format.

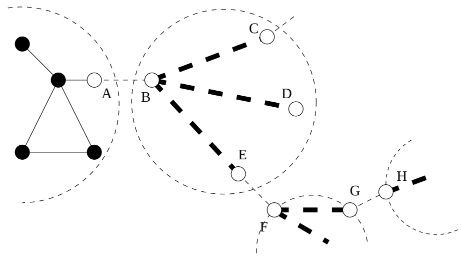

Figure 2: *Tunnel setup through Reservation Domains*

As shown in Figure 2 the router B might be the end- or start point for a lot of tunnels each connecting a RD ingress with a RD egress router and transporting the traffic aggregate between these routers.
The final mapping of a RSVP flow to the tunnel is triggered by the RSVP *RESV*-message. When E receives the message it will check the available tunnel resources, launching another capsule to adjust the tunnel settings if necessary and will configure the router E to map the packets of this flow to the tunnel. Dependent on a smart network provisioning a reconfiguration of the tunnel might occur rather infrequently [DGB00].
The drawback of such an approach is the required de- and encapsulation at the borders of RDs. Because of administrative and security reasons it is not desirable to allow foreign mobile code to be executed in an ISP's network. To avoid that problem, a *path*-capsule received by B queries whether a decapsulation is necessary or whether the next RD is able to handle the encapsulated packets. Even if this requires the design of a communication interface between these agents, this will be easier to achieve than to allow and secure the execution of foreign code.

## References

[BB00a] Florian Baumgartner and Torsten Braun. Quality of service and active networking on virtual router topologies. *The Second International Working Conference on Active Networks, IWAN*, October 2000.

[BB00b] Florian Baumgartner and Torsten Braun. Virtual routers: A novel approach for qos performance evaluation. *Workshop on Quality of future Internet services, QofIS'2000*, September 2000.

[BBBG00] R. Balmer, F. Baumgartner, T. Braun, and M. Günter. A concept for rsvp over diffserv. *IEEE, ICCCN'2000*, October 2000.

[BS99] Marcus Brunner and Rolf Stadler. Virtual active networks - safe and flexible environments for customer-managed services. *Proceedings of the Tenth International Workshop on Distributed Systems: Operations and Management (DSOM'99)*, 1999.

[DGB00] Gabriel Dermler, Manuel Günter, and T. Braun. Towards a scalable system for per-flow charging in the internet. ATS, 2000.