Implementation of Differentiated Services over ATM

Torsten Braun, Arik Dasen, and Matthias Scheidegger; Institute of Computer Science and Applied Mathematics, University of Berne, Switzerland
Karl Jonas and Heinrich Stüttgen; C&C Research Laboratories Heidelberg, NEC Europe Ltd., Germany

Abstract

The paper describes the concept and the implementation of Differentiated Services over ATM. ATM components are being used in order to implement DiffServ traffic conditioning components such as shaping and policing. The implementation architecture based on a Linux router platform and some initial performance measurement results are presented.

1 Introduction

The implementation of Differentiated Services (DiffServ) [1] such as Premium and Assured Services requires to implement additional traffic conditioning components within IP routers, in particular within the boundary routers of a DiffServ domain. Such traffic conditioning functions are among others classification, marking, metering, shaping, and policing. Traffic conditioning puts additional burden to the routers and may limit their performance. In particular, shaping is one of the most expensive operations if implemented in software. Instead of implementing these traffic conditioning functions in software, hardware implementations can be used in order to overcome potential performance bottlenecks. The approach described within this paper uses available ATM hardware for these purposes. In particular, shaping and policing functions are already implemented in ATM switches and ATM network interface cards (NICs). We discuss whether ATM can be used to implement the most popular DiffServ services Premium Service (Expedited Forwarding, EF) and Assured Services (Assured Forwarding, AF). We describe the implementation options that have been realized on a Linux based router platform. The developed solution is similar for both EF and AF.

2 Differentiated Services Operation

In the case of EF service, a so-called first-hop (FH) router in the customer network has to classify EF flows, mark them with an appropriate Differentiated Services Codepoint (DSCP) and shape a flow[1] to the EF peak rate pre-negotiated among the customer and its ISP (Figure 1). The egress router (ER1) of the customer has to shape the whole EF traffic to the pre-negotiated rate. The ingress router (IR1) at ISP1 then performs policing based on the profile pre-negotiated with customer 1, while ER2 again performs shaping based on the profile negotiated among ISP1 and ISP2. IR2 again performs policing functions.

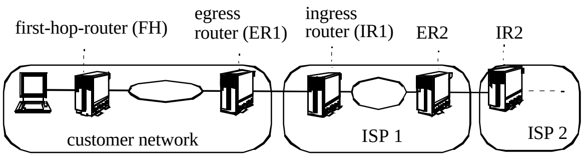

Fig. 1 Normal Differentiated Services

AF is based on the same architecture with the difference that shaping and policing is replaced by remarking packets to higher drop precedence values. In addition, dropping mechanisms considering the drop precedence values are implemented in egress and ingress routers.

3 Expedited Forwarding over ATM

This section describes how available shaping and policing functions in ATM switches and ATM NICs can be used to set up a DiffServ implementation which avoids the deployment of software functions for shaping and policing within IP routers. Figure 2 depicts a scenario with hosts and Diffserv routers interconnected via ATM switches. The sending host is connected via any link technology such as Ethernet to

[1] Micro-Flows (MFs) are single application level flows, while flows consist of several aggregated MFs.

the first-hop router. Boundary routers (egress and ingress routers) of different domains are interconnected via ATM switches in order to move policing functionality to the ATM switch and shaping functionality to the ATM NIC of the egress router. The same applies for the first-hop and egress router of the customer. In the following subsections, we describe two alternative implementation approaches.

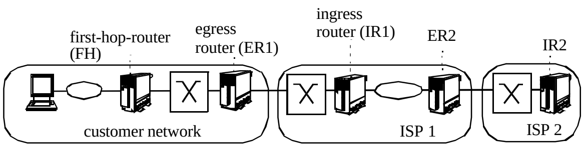

Fig. 2 DiffServ over ATM Scenario

3.1 Implementation Approach 1

With the first approach, the FH classifies all EF packets according to the defined profile information configured within FH. In the next step, the various flows must be shaped in order to achieve conformance. For shaping at FH, we set up a number of PVCs between FH and ER1. For each PVC we create a Classical IP over ATM interface at both of the routers. Each of these logical ATM interface address pairs share a common IP sub-network. The IP over ATM interfaces can be configured with a shaping rate. For each flow to be shaped, we use one CBR PVC and configure FH and the ATM switch between FH and ER1 so that the ATM NIC at FH performs CBR shaping and the ATM switch performs CBR policing on this PVC. Of course, this approach can lead to over-reservation if the peak CBR bandwidth can exclusively be used by the corresponding VC only. However, some ATM switches include priority mechanisms allowing unused CBR bandwidth to be used by lower priority connections.

Within FH, the forwarding behavior must be changed so that FH forwards all packets of a certain EF flow over the corresponding CBR PVC. This requires enhanced forwarding that considers flow information such as IP source addresses, DSCPs, port numbers or even higher protocol information in addition to destination addresses.

Another shaping step must be performed at the edge between the customer network and ISP1. For that purpose, ER1 takes all packets from the EF PVCs and forwards them over a single PVC towards ISP1. The ATM NIC at ER1 at the border of the customer network performs shaping according to the rate negotiated between the customer and ISP1, while the ATM switch at the border of ISP1 performs policing functions. The same behavior applies at the boundary between ISP1 and ISP2 in routers ER2 and IR2. In all of these cases, the egress router has to classify the EF service packets and to forward them over the PVC that has been established for EF traffic to the next ingress router. This means that the egress router must consider the DSCP for forwarding.

Egress routers could be replaced by an ATM switch if each flow is shaped by a first-hop router. Figure 3 shows the corresponding scenario. Here, the FH performs VC shaping for each EF flow. The aggregate shaping at the DiffServ domain border is then achieved by an ATM switch bundling several EF flows into a single permanent virtual path (PVP). On the other side, VP policing should be configured at ISP1.

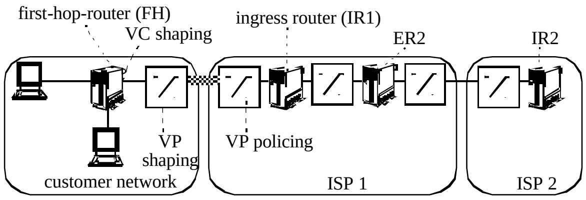

Fig. 3 DiffServ over ATM with VP aggregation

Figure 4 depicts our DiffServ implementation architecture without ATM support [5]. Packets are forwarded to a device, classified, processed by a service handler, which may include marking, shaping, or policing, and queued for transmission. Finally, a queuing mechanism such as weighted fair queuing (WFQ) or priority scheduling must collect the packets from the different queues and schedule them for transmission over the output interface.

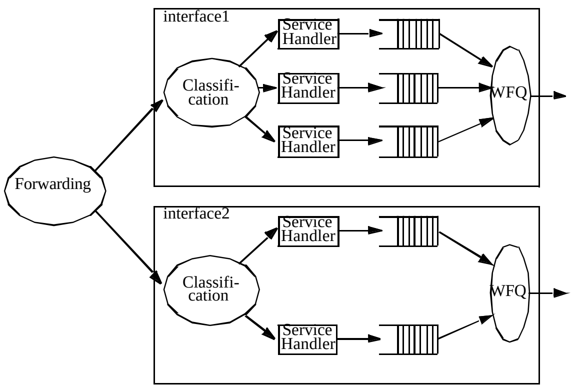

Fig. 4 DiffServ Implementation

Figure 5 shows the same architecture if the second interface is replaced by an ATM interface. In this case, the traffic is classified earlier and packets of classified EF flows are directly forwarded to particular logical ATM interfaces using the enhanced forwarding functions. Since only one flow is processed by a logical ATM interface, no further classification is required. In addition, no output queuing such as WFQ is required since the packets to be transmitted over this logical interface come from a single queue. The service handlers used with this interface can be used to mark the packets to be transmitted accordingly.

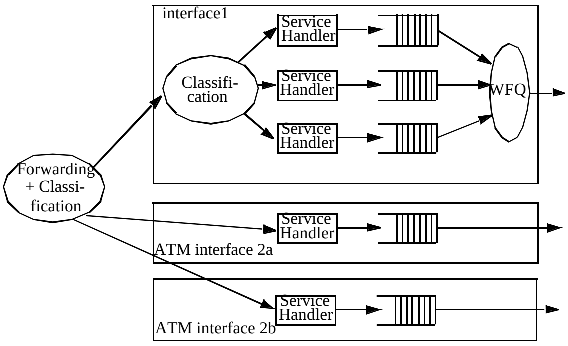

Fig. 5 DiffServ Implementation over ATM I

In the following we assume, that packet classification and marking is performed in a first-hop router rather than within an end-system. Two options exist for the implementation of approach 1. Both are based on the *iproute2* [4] package, which allows to select the outgoing router interface dependent on other parameters than destination addresses such as DSCPs, port numbers, protocol identifiers and IP source addresses. For example, *iproute2* can be used for advanced routing by the following commands:

```
$ ip rule add tos 0xB8 table 1
$ ip route add 10.1.2.0/24 via
10.2.92.1 table 1
```

In the example above, packets to network 10.1.2 with the DSCP set to 0xB8 (EF) are forwarded via interface 10.2.92.1.

Another useful package is *ipchains*, which supports classifying/marking with the following command:

```
$ ipchains -A input -s daffy -d weasel
-i eth0 -t 0x01 0xB8
```

Here, all packets from daffy to weasel are marked with the EF DSCP (0xB8) in the input chain before forwarding/routing is done by the Linux kernel.

Iproute2 can now be combined with either *ipchains* or with our DiffServ implementation [5] in the following ways:

1. For the first option, i*pchains* [6][7] can be used for flow classification and changing the Type-of-Service (ToS) field, i.e. the DSCP. Then, the packets are forwarded by *iproute2* using ToS-based routing.
2. The second option is to classify/forward packets by *iproute2* and mark them by the DiffServ implementation. *Iproute2* selects the ATM interface associated with the special EF ATM PVC dependent on classification parameters such as flow information. The service handler queuing discipline of the DiffServ implementation is then responsible for marking the packets with the EF DSCP.

3.2 Implementation Approach 2

There are two main reasons why the approach 1 is not satisfactory. First, the implementation architecture explained in section 3.1. does not fit well in our existing DiffServ implementation since some additional classification has to be done within the routing decision. Second, there is a tremendous waste of (private) IP addresses and logical interfaces. For each PVC one pair of logical ATM interfaces and IP addresses must be used. To avoid these two problems we developed and implemented a second approach that does not use *ipchains* or *iproute2*, but the complete classification is now performed by the DiffServ implementation.

This second implementation approach is based on sophisticated ATM queuing disciplines which fit perfectly in our existing DiffServ implementation. It can send traffic directly over a particular PVC without using any private IP addresses. Also, no special routing decision has to be made since this queuing discipline uses the standard outgoing interface. The queuing discipline has been realized as a kernel module in the same way as all the other DiffServ components.

The new modules were designed to be similar to the QoS modules coming with the standard kernel distribution which ensures interoperability between them. In fact, new and standard modules can arbitrarily linked together. Note, that this concept does not cause additional packet copies since only control data structures are exchanged among the different modules.

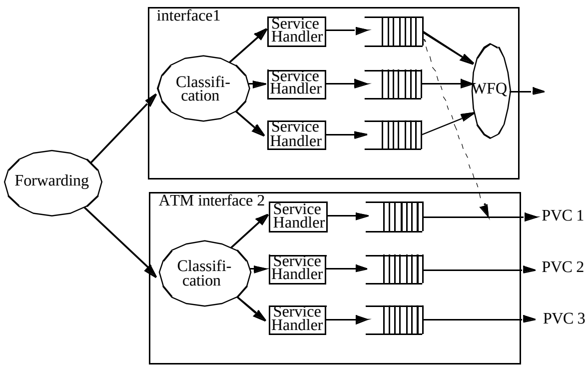

Fig. 6 Diffserv Implementation over ATM II

Figure 6 shows the structure of implementation approach 2. Packets are first forwarded and classified before being processed by a service handler (e.g. for AF, EF, or best-effort packets). Some ATM related information is attached and the packet is sent to a

FIFO or any other attached queuing discipline. The dequeue function of that queuing discipline encapsulates and sends the packets over the specified ATM PVC. Note that we also can filter packets on a non-ATM interface and send them over an ATM PVC (see dashed arrow in Figure 6). This would allow to interconnect two routers via Ethernet for best-effort traffic and ATM for EF (and AF) traffic.

Figure 7 shows the structure of our DiffServ implementation using the ATM queuing discipline for EF in a first-hop router in more detail. After marking by the service handler, the classifier classifies them and sends EF packets to the according ATM queuing discipline or to another EF queuing discipline, e.g. based on a token bucket filter implemented in software.

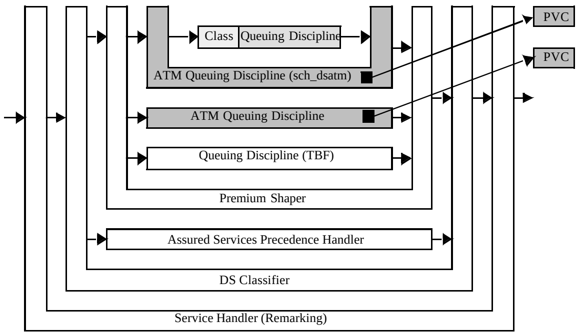

Fig. 7 DiffServ Implementation with ATM Queuing Discipline for EF

3.3 Performance Measurements

The implementation architecture described above has been implemented on a Linux based router platform. As ATM NICs we used the Efficient Networks ENI 155P card with the Efficient 155 Mbps PCI Linux driver provided in the ATM on Linux package version 0.59 [2].

The test scenario used for performance measurements is depicted in Figure 8. Two hosts (daffy and elmer) are connected to a first-hop router which is interconnected via a default ATM PVC (2 Mbps) and an ATM PVC (5 Mbps) for EF traffic to another router which is directly connected to the receiver host (weasel). For the performance measurements we used ttcp and ftp on the three end systems.

For the evaluation of approach 1, two TCP flows have been generated: flow 1 from daffy to weasel and another flow 2 from elmer to weasel. The two routers in between have been setup in order to separate the two flows in both directions over the two PVCs for EF and best-effort traffic respectively. The EF flow got 4.03 Mbps while the BE flow got 1.65 Mbps only. This means that in both cases approximately 80% of the bandwidth configured at the ATM switches have been achieved. The difference is mainly due to ATM cell (10%) and AAL/IP header overhead. Therefore, the results show clearly that classification of EF flows and its shaping over ATM works well.

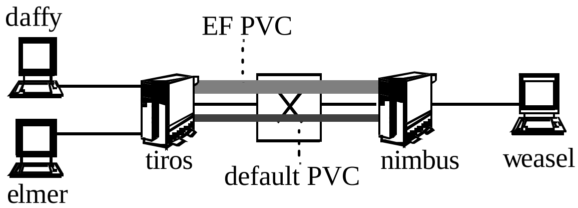

Fig. 8 DiffServ over ATM Implementation Scenario

For the evaluation of approach 2, several different tests have been performed. For the first test a single EF flow has been generated in order to prove the shaping functionality at the router´s ATM NIC. 4.21 Mbps have been achieved for that flow. In a second test, additional aggressive background traffic has been generated. However, the EF flow still got exactly the same throughput which proves that the EF traffic has been protected by the router and the ATM VC and no single byte got lost due to the background traffic. Shaping to 10 Mbps yielded an end-to-end throughput of 8.84 Mbps. Policing at the ATM switch reduced the bandwidth in the 5 Mbps case from 4.21 to 4.07 Mbps. The results of approach 2 are slightly better than those with the first approach. We get about 85-90% of the configured bandwidth and again, which is close to the upper theoretical limit due to overhead by ATM cell, AAL, IP, and TCP headers.

4. AF Service over ATM

For normal AF Service, a first-hop router classifies the packets according to some profile with the corresponding AF class. In addition, the drop precedence is selected dependent on whether the flow exceeds certain bit rate limits to low, medium and high drop precedences. The same behavior is applied in the other routers with the difference that the packet classification is more coarse grained in the backbone. Two options exist for mapping AF over ATM:

1. The first option keeps the AF router implementation with all its marking functions as it is. The AF traffic is forwarded over a certain AF PVC to the next hop. Again, the forwarding behavior must consider information such as IP source address, DSCP etc. in addition to the IP destination address. Low

drop precedence packets should be mapped to CLP=0 cells, while medium and high drop precedence packets should be mapped to CLP=1 cells of this PVC. Of course, this mapping requires appropriate support via an API. The mapping ensures that ATM cells with CLP=0 are forwarded with higher probability than CLP=1 cells. The ATM PVC for AF packets can be setup with a peak rate equal to the negotiated rate. This option leaves the task of packet marking (which requires token buckets in the routers) within the routers. The implementation of this option can use the enhancements described for EF above. The configuration of the AF packet marking and dropping components (in particular the parameters for the Random Early Discard (RED) mechanism) must match the ATM QoS parameters.

2. The second option tries to use ATM leaky bucket mechanisms for correct marking of AF packets. Two approaches are possible for the second option:
   - The first one is a rather simple approach. Between an egress and an ingress router shaping and policing functions in ATM NICs and switches could be used, i.e. marking non-conforming cells as CLP=1. At the ingress router, packets consisting of cells with CLP=0 should get low drop precedence while the other ones should get high drop precedence values. This has the disadvantage that only low and high drop precedence values are used, but not medium drop precedence. In addition, in the case that the peak bit rate is exceeded slightly, each packet might always have a single cell that is marked with CLP=1. This can then lead to marking more, or even all, packets with high drop precedence than in the normal DiffServ case.
   - Another approach is to mark packets in the ingress router with low drop precedence if the number of CLP=1 cells are below a certain threshold, while packets are marked with high drop precedence when the number of CLP=1 cells exceeds another threshold. All other packets are marked with medium drop precedence. Such a scheme must be made consistent to the marking schemes performed on a packet basis in IP routers. It also requires that a policing function based on CLP marking is available within the ATM switch.

   The implementation of both of these approaches for the second option requires that DiffServ packet processing has access to the CLP values of the ATM cells. If cell reassembly is performed by an ATM NIC in hardware, the CLP bits of the ATM cells are not be visible to the software driver on top of it. This means that this approach cannot be implemented over ATM NICs with ATM SAR functions implemented in hardware.

Therefore, it is more reasonable to use efficient ATM cards with SAR hardware implementations and keep the original IP enhancements for AF service. Otherwise, the introduced overhead on AAL level might increase to a similar level as for AF enhancements of the IP layer. So, it might be the best if the AF implementation on IP level remains unchanged but sophisticated PVCs are provided for AF flows or aggregated flows in order to protect them from aggressive best-effort flows.

For AF service over ATM implementation we, therefore, propose the first option, which is identical with the EF service implementation over ATM with the difference that CLP marking of ATM cells should be done dependent on the DSCPs of the IP packets. Figure 9 shows the detailed implementation structure of the first option. AF packets are handled by the appropriate service handler and queued by the corresponding queuing system (e.g., a RED queue). Then, the packet may be sent via a dedicated PVC corresponding to the AF flow.

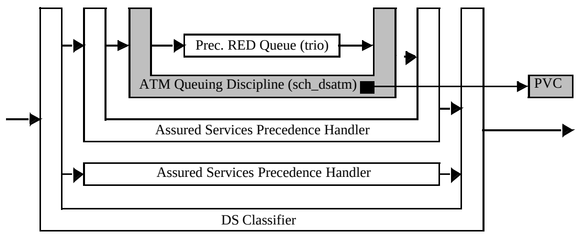

Fig. 9: DiffServ Implementation with ATM Queuing Discipline for AF

The queuing discipline does not differ much from the queuing discipline for the EF described above. We actually use the same code for both services but for AF we have enhanced the implementation by mapping the AF dropping precedences to the CLP bit in the ATM cell header. As proposed in [8] nrt-VBR PVC should be used for AF services. Since the ATM NICs being used in our system lacks any nrt-VBR support, CBR has been used for AF, too.

Conclusions

This paper discussed several options for implementing Differentiated Services over ATM including performance measurements. In particular, the presented implementation approaches allow to avoid software implementations of shaping and policing in DiffServ routers. Moreover, the implementation architecture can easily be integrated with Multi-Protocol Label Switching (MPLS).

References

[1] IETF Differentiated Services Working Group, http://www.ietf.org/html.charters/diffserv-charter.html

[2] W. Almesberger: Linux-ATM, http://lrcwww.epfl.ch/linux-atm

[3] W. Almesberger: atm/doc/usage.txt in atm-0.59.tar.gz in [2]

[4] A. Kuznetsov: iproute2, ftp://lrcftp.epfl.ch/pub/people/almesber/misc/iproute2-2.1.99-now-ss990203.tar.gz or ftp://ftp.sunet.se/pub/Linux/ip-routing/iproute2-2.2.4-now-ss990824.tar.gz

[5] T. Braun, H. Einsiedler, M. Scheidegger, K. Jonas, H. Stüttgen: A Linux Implementation of a Differentiated Services Router, submitted for publication

[6] http://www.rustcorp.com/linux/ipchains/

[7] http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html

[8] A. Dobreff: Comparison of Simulation and Real Functionality for the Mapping of Differentiated Services to ATM, Diploma Thesis, University of Berne, November 1999, http://www.iam.unibe.ch/~rvs/publications/dobreff.Diplom.pdf