Evaluation of Bandwidth Broker Signaling

Manuel Günter
Torsten Braun

IAM-99-002

May 1999

.

Abstract

The Differentiated Service (DiffServ) architecture for the Internet implements a scalable mechanism for quality-of-service (QoS) provisioning. Bandwidth brokers represent the instances of the architecture, that automate the provisioning of a DiffServ service between network domains. Although several bandwidth broker implementations (e.g. [Bri98]) have been proposed, the alternatives and trade-offs of the different viable approaches of inter-broker communication were not studied up to now.

This paper presents the broker signaling trade-offs considered in the context of a DiffServ scenario used by the Swiss National Science Foundation project CATI [SBGP99], and it presents results gathered by simulations.

CR Categories and Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design; C.2.4 [Computer-Communication Networks]: Distributed Systems; C.2.6 [Computer-Communication Networks]: Internetworking.

General Terms: Design, Experimentation.

Keywords: Signaling, bandwidth broker, service level agreement, simulation, QoS, differentiated services, integrated services.

1 Introduction

The DiffServ architecture[BBC+98] uses automated bandwidth brokers [ROT+98] to negotiate service level agreements (SLA) between different autonomous systems. These agreements describe the volume of DiffServ traffic that can be exchanged between two domains and the price that such traffic will cost. If all the domains between two end users have engineered their networks properly and have established SLAs for the DiffServ volume expected, the DiffServ architecture is said to guarantee end-to-end QoS. However, it is obvious that the local traffic volumes produced by end-users show a dynamic behavior which has to be reflected also in the SLAs between core networks. Concrete numbers do not exist since no DiffServ service is established yet, but the University of Berkeley, California currently processes initial studies on what QoS service the Internet users value [VER98].

One option to cope with the changing user requirements is to signal each change in flow activities through the bandwidth brokers to the core networks. However, this is not desirable, since this would lead to the equivalent scaling problem that the Integrated Services architecture (IntServ) faces, thus undermining the main advantage of DiffServ. Therefore, the signaling between the bandwidth brokers must reflect aggregated changes and should be decoupled to some degree from user flow forwarding. The simulation presented in this paper is used to describe and evaluate pit-falls and trade-offs of such aggregation and decoupling.

In chapter 2 we identify the main trade-off of broker signaling as the trade-off between scalability and cost on one hand and end-to-end QoS guarantees on the other hand. We also describe the terminology and assumptions of the simulation which is used to find a solution to the trade-off. Chapter 3 presents an overview of the simulation implementation. Chapter 4 presents the results of the simulation and chapter 5 concludes.

The rest of this introductory chapter describes the context in which our work has been performed and the DiffServ architecture used.

1.1 Context

The Swiss National Science Foundation project *Charging and Accounting for the Internet* CATI is based on an IntServ [BCS94] Internet architecture for provisioning charging and accounting service on the IP level. The IntServ architecture uses the Resource Reservation Setup Protocol (RSVP) to signal reservation requests on a per flow[1] basis. CATI uses the RSVP signaling to exchange charging information and electronic payments. However, as stated in [MBB+97], core networks cannot

---

[1] A flow is a connection between two peers each identified by an IP Address and a port number.

support the IntServ architecture because of scalability problems. Backbone routers would have to keep state information for millions of flows. Furthermore, many flows (e.g. `http` related ones) are short-lived therefore reservation of resources for them is overhead. Nevertheless, RSVP has been deployed successfully in small networks (host networks) and is used by a growing number of applications (see e.g. [Lei99]).

To address this situation, the CATI project started to evaluate the DiffServ architecture in order to use IntServ in the peripheral (host-) networks and DiffServ in the core networks (IntServ over DiffServ [BYF$^+$98]). As mentioned before, the implications of different signaling mechanisms for the DiffServ architecture are not well understood yet. The performance of the IntServ over DiffServ scenario depends on the DiffServ signaling in the core network which is the signaling between bandwidth brokers. Therefore, CATI needed to further investigate the DiffServ signaling. This paper represents the first results of our research in that area.

The next subsection provides an overview of the DiffServ architecture assumed by CATI.

1.2 Differentiated Service Architecture

The DiffServ architecture uses the IP packet's DiffServ Code Point (DSCP), located in what was formerly called the type of service (TOS) byte. The DSCP describes what kind of forwarding this packet will experience. Once the DSCP is set, all traffic with the same DSCP code is treated in the same way, regardless of its other characteristics (e.g. source/destination address, port). Thus, traffic of many different flows is classified into a small number of traffic classes. This *aggregation* mechanism then easily scales to large core networks, that forward huge numbers of flows.

Two proposed differentiated service classes are *premium* and *assured* service. The premium service is used to provide the characteristics of a virtual leased line (constant bit-rate). The assured service offers less end-to-end guarantees, but allows bursty traffic. Both services are similar in the sense, that the service is expressed in terms of a maximum bit rate. Basically, host networks set up SLAs with their Internet Service Provider (ISP) where they agree on such a rate of DiffServ traffic that the host network can inject into the Internet. ISPs will forward DiffServ packets according to the DSCP marking. ISP networks will queue and schedule DiffServ packets separate from normal ('best-effort') IP packets. Based on the SLAs, the ISPs will engineer their network in order that pure DiffServ traffic cannot congest it, and will setup SLAs with adjacent networks, thus enabling end-to-end QoS for DiffServ traffic. Bandwidth brokers are software agents that automate the SLA negotiation. Upon SLA negotiation for new incoming DiffServ traffic, they have to check if their network is able to support it without congestion and they have to (re-)negotiate SLAs.

The planing and provisioning of a working DiffServ network causes additional work and expenses for an ISP. Therefore, the ISPs will charge for DiffServ traffic. As mentioned before, the ISPs need the customer to commit to a maximum bit-rate in order to provide the service. This is described in the SLAs. If a customer sends more DiffServ packets than agreed upon in the SLA, the ISP will shape that traffic. In the case of the premium service, it will simply drop the exceeding packets. In the case of exceeding assured service traffic, the ISP will reclassify the packets as ’best-effort’ by changing the DSCP marking.

Figure 1 shows an example of a working DiffServ scenario. Here, two host networks (H1 and H2) have established an SLA with an ISP A for 500 Kb per second assured traffic each. They inject that amount of DiffServ traffic plus a large amount of best-effort traffic through fast access links. ISP A forwards all traffic to ISP B. The brokers of the two ISPs have already established a sufficient SLA (1Mbps) between them, thus the DiffServ traffic can continue on its path to the destinations. The link between ISP A and B is only of limited size, thus it can congest. However, this congestion only affects the best-effort traffic.

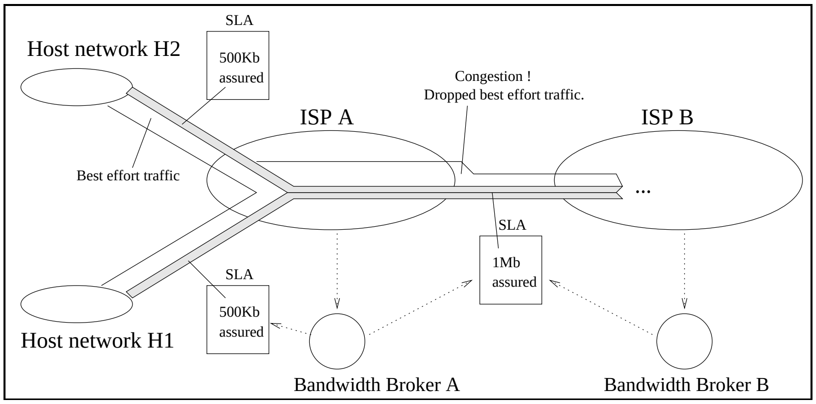

Figure 1: The ideal DiffServ scenario.

Many questions are left open in this scenario: how and when is the SLA between ISP A and B established? How is the bit-rate of this SLA determined? What happens when the DiffServ traffic produced by the hosts splits to reach many destinations and travels through many ISPs? These questions fall back to the question of the signaling between the bandwidth brokers, which is the subject of this paper. The next section will present trade-offs of a broker signaling design.

2 Narrowing the Design Space of the Broker Signaling

The Differentiated Service architecture consists of a data transport level and a control level. The data transport level of DiffServ includes the different kind of DSCP codes and their corresponding per hop behavior. The work in this area is far progressed within the Internet Engineering Task Force (IETF). However, when it comes to evaluate the end-to-end behavior of DiffServ, the control level must be specified. Bandwidth brokers play the main role at the control level of the DiffServ architecture. As mentioned before, the design space of the broker signaling is not explored up to know.

2.1 The Value of a Differentiated Service Architecture

The value of a DiffServ architecture can only be judged, when the broker signaling is specified to more detail. Here is a list of the three main qualities of a DiffServ architecture:

Scalability. IntServ and its RSVP protocol allow for a fine grained end-to-end QoS support. Unfortunately, it does not scale to large networks such as backbone networks in the ever growing Internet. Scalability was the very reason for DiffServ to come to existence, therefore this is a prime quality of a DiffServ architecture.

End-to-end QoS. If the per-hop nature of DiffServ is combined with a per-hop control structure, statements about end-to-end QoS are limited to statistical evidence. Nevertheless, it is end-to-end QoS guarantees that the end-users want, and what they willing to pay for. Therefore, we must evaluate control structures with an end-to-end scope. Such a control structure will be based on a common broker signaling protocol.

Cost. While the data transport level of DiffServ is fairly simple, the control level might add management complexity for the providers. These costs must be paid by the users. The ratio of possible end-to-end QoS value compared to the costs will define the competitiveness of the DiffServ architecture in the data transport market.

The simulations we describe in this paper try to show the way to a good tradeoff between end-to-end QoS on one hand and scalability & costs on the other hand. The focus is on the control level, thus on the level of the signaling between bandwidth brokers.

2.2 Simulation Terms and Assumptions

Our simulation uses a coarse grained model of the Internet. The inter-network is modeled as interconnected autonomous systems. Some of these systems are host networks, which act as traffic sources and sinks, the rest are ISP networks[2] which act as pure transport networks. The following paragraphs describe the simulation terms and assumptions for the different aspects of: business, traffic generation, signaling between host- and core networks, and reservation and notification strategies.

Business Assumptions. Each bandwidth broker represents a business party, namely the ISP of the network that it controls. Business models for traffic forwarding may be complex. We made three basic assumptions:

1. ISPs demand money from other networks that want to reserve for the injection of DiffServ traffic into their networks.
2. Host networks do not demand money for incoming DiffServ traffic.
3. ISPs want to avoid breaking SLAs.
4. Host networks avoid breaking SLAs.

With the assumptions 1) and 2) the simulator is able to simulate the exchange of money between the brokers. Money exchange is based on the broker's business policy and on the amount of reservation as well as on the actual DiffServ usage. However, this is not subject of this paper.

Assumption 4) is supported in the simulator, so each host network is only generating as much traffic as it has negotiated via its SLA. However, assumption 4) is not necessary because the DiffServ architecture copes with non-cooperative behavior of hosts by means of policing.

Assumption 3) is highly important in the context of this paper. This is because the desired end-to-end QoS can only be achieved if the ISPs are collaborative.

Traffic generation. The DiffServ traffic is modeled as aggregated flows. All flows between two distinct host networks are modeled as one aggregated flow. In an inter-network with n host networks, each host network generates $n-1$ aggregated flows which add up to a total of $n(n-1)$ aggregated flows. The simulation allows the flow generation to be parameterized in two ways: (1) The total amount of traffic a single host network generates can randomly vary between a minimum and a maximum value. (2) The percentage of traffic assigned to an aggregated flow can

---

[2] For notation convenience we will often refer to such networks simply as 'ISPs'.

change randomly with a parameterized speed which we call the *fluctuation* of the traffic distribution.

Signaling between host- and core networks. This paper focuses on the signaling between the bandwidth brokers of ISPs. Nevertheless, the host networks (as traffic sources and sinks) initiate the signaling. In the simulation we do not describe what causes the first notification to a bandwidth broker. It is only assumed, that host networks have different upcoming needs for DiffServ traffic. They announce this need or changes in their needs to the appropriate bandwidth brokers. This announcement can be interpreted as the request for a setup of an SLA between a host network and its access network either via the brokers or manually. Another interpretation is an automated IntServ to DiffServ mapping mechanism, that notifies the bandwidth broker.

Reservation and notification strategies. The bandwidth brokers buy and sell reservations of DiffServ bandwidth. Each such purchase is expressed in an SLA. Before a reservation is granted, a bandwidth broker may want to check if it really can grant that request. This includes a check of the capacity of the broker's own network, but it can also include signaling to neighbor brokers to either just inform them that the DiffServ traffic volume will change or to reserve additional bandwidth from them. Both reservation and notification must be handled by a broker signaling protocol. For clarity, we use the term *reservation* for the negotiation of an SLA describing the conditions under which the reserved amount of DiffServ traffic is forwarded with the expected per-hop behavior. Such a reservation may be triggered by an incoming *notification* of a DiffServ reservation request and it may be blocked until a further notification has been issued to the involved neighbors. Such notification decisions are the subject of the simulation described in this paper. The following list summarizes the design decisions that we want to evaluate. The three basic classes include issuing *no notification*, issuing an *end-to-end notification* or issuing a *limited notification*.

No notification. The most simple DiffServ control structure could foresee only reservations between brokers, but no notifications propagated further. This adds no notification costs or scalability problem to the DiffServ architecture, but it would not allow end-to-end guarantees. Especially when DiffServ traffic has to be shaped inside of the core networks, new SLAs need to be established. This is situation shown in figure 2. Note, that if the host network H1 has paid money to the ISP A for injecting DiffServ traffic to the Internet, its users will certainly complain when their traffic is shaped and subsequently dropped or congested in the core network C. In a reasonable scenario, the

ISP networks would *measure* the DiffServ traffic and *overprovision* their networks and their SLAs with the adjacent networks. Such measurement based reservation together with significant overprovisioning is what we call the *adaptive reservation* scenario. It can enable cooperating ISPs to give some statistical end-to-end QoS guarantees. A particular problem in this scenario is the loose cooperation between the ISPs. Overprovisioning causes cost for an ISP. The end-to-end QoS is lost when one ISP is not overprovisioning sufficiently.

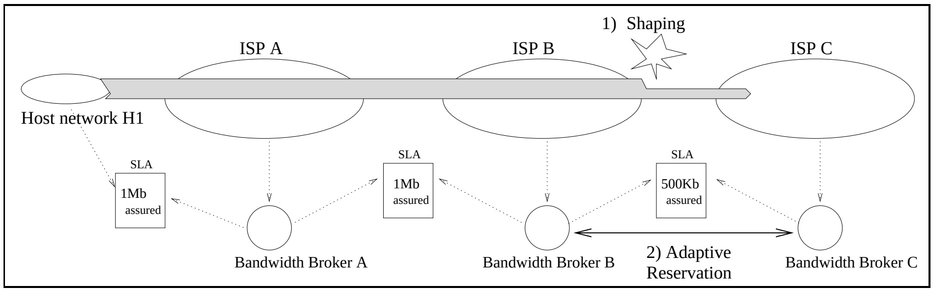

Figure 2: Adaptive reservation triggered by shaping.

End-to-end notifications. Before the establishment of a new SLA, the broker notifies the involved neighbor broker(s), and sets up a new SLA with them if necessary, to accommodate the new DiffServ traffic. The neighbor brokers acts likewise, notifying upstream. Thus, when the originally requested SLA is accepted, all SLAs from traffic source to sink have already been updated. Obviously, this allows for end-to-end QoS. Figure 3 depicts this situation. However, if an SLA would be set up for each flow, this DiffServ architecture would be equivalent to the IntServ architecture and thus suffer form the same scalability problems. Even when only aggregated flows trigger notifications, the number of notification grows with the square of the number of networks. Furthermore, each new aggregated flow would face the delay of the end-to-end notification between the brokers. The bandwidth brokers could then become a bottleneck with negative impact on the DiffServ traffic performance. As mentioned before, it is clearly undesirable that the broker signaling burdens the DiffServ architecture with a scalability problem.

Limited notification. A simple approach to address the scalability problem is to decrease the granularity of the notifications, so that not each flow or change in an aggregated flow triggers notifications and that not each notification is propagated to further brokers. The second option is depicted in figure 4.

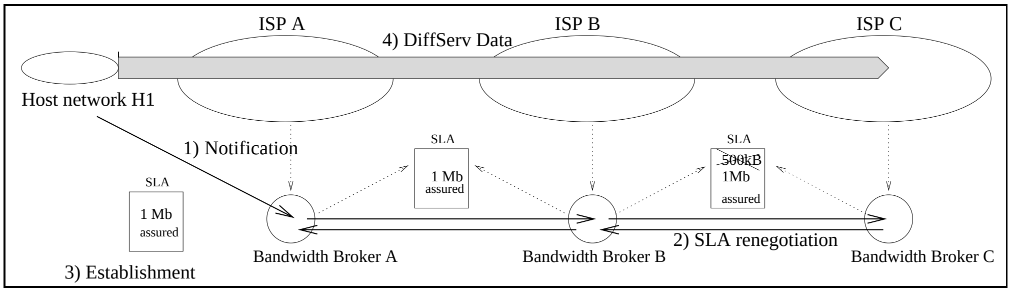

Figure 3: End-to-end notification with end-to-end QoS guarantee.

The obvious problem with such an approach is, that it may lose the end-to-end QoS property. Another problem is that the notification process needs flow destination information. If one notification covers different flow aggregations it is not possible in advance to tell which ISPs will experience an impact.

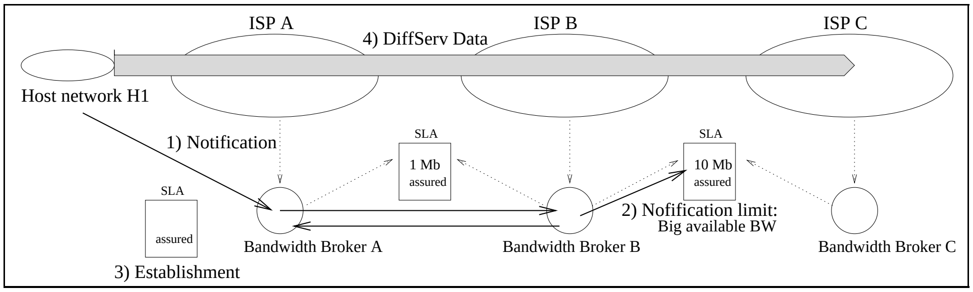

Figure 4: Limited notification.

As mentioned before, in the view of the authors, a fine grained end-to-end notification is not suitable for the DiffServ control level. Therefore, only the adaptive reservation scenario and the limited notification scenario are evaluated by our simulator.

3 Structure of the Simulator

The simulator runs a given number of simulation rounds. A single round has four different phases. Note, that the simulator is build to support the adaptive reservation scenario and different limited notification scenarios. Here are the four phases of a simulation round:

Traffic calculation. According to the fluctuation value, the traffic distribution to the different destinations changes.

Traffic notification and injection. All flows are injected in the network. This can be preluded by notification and reservations between the bandwidth brokers. Furthermore, the traffic is shaped if SLAs are violated. Dynamically, measurements are taken and stored.

Usage based charging. The traffic is charged according to the measured usage.

Adaptive reservation. The ISPs can adapt their SLAs based on the usage measurements. Note, that these reservations do not trigger notifications.

The simulator is written in `Java`. Due to space limitations we can only briefly describe its architecture.

3.1 Simulator Architecture

The main class is `NetworkSimulator`. It controls the program flow and holds the `main` routine. Figure 5 shows the data and control flow of a simulation run.

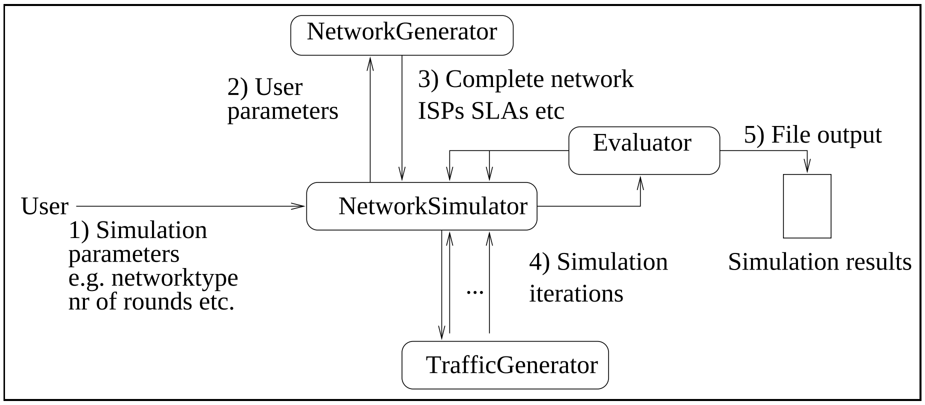

Figure 5: The data flow of the simulation.

1. The user starts the simulation with various parameters describing the reservation and notification options the broker signaling should use, as well as the number of simulation rounds, and the network type to use.
2. The class `NetworkGenerator` can generate different types of parameterizable networks.

3. The simulator iterates for the specified number of simulation rounds.
4. The class `Evaluator` describes what measurements an values to extract in each round.
5. After the simulation, the extracted measurements are written to a log file.

At run time, the ISP objects are interlinked via channel objects and SLA objects. The SLA objects are manipulated by bandwidth brokers. Figure 6 shows two interlinked ISPs objects. For one ISP object, the object relations are described in more detail.

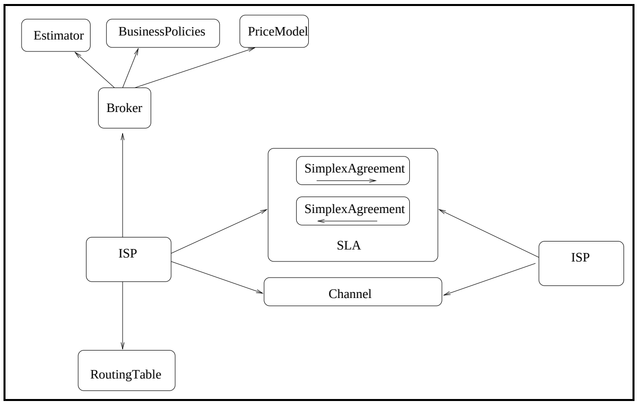

Figure 6: The realization of an ISP-ISP relation.

Without going into detail, each ISP needs a routing instance to forward traffic and a bandwidth broker to renegotiate SLAs. For each connection between ISPs (channels) there is an SLA describing the inbound and outbound differentiated service agreements. The broker uses a price model to individually negotiate prices for DiffServ offerings. Furthermore, it has a business policy, which e.g. describes how to treat notifications, when to request SLA negotiation (when to buy bandwidth) and the chosen level of overprovisioning. An estimator object helps to analyze traffic tendencies in the network.

3.2 Networks Types

The network generator currently features two kinds of customizable networks: the `Dumbbell`- and the `Slalom` networks.

`Dumbbell.` This network has two interconnected backbone networks. As shown in figure 7 there is an equal number of n host networks attached to each of the two core network. Thus, the channel between the two core networks is a possible bottleneck.

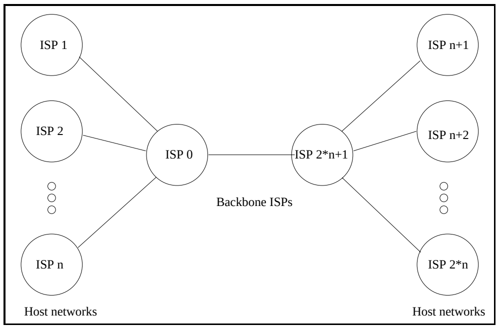

Figure 7: The `Dumbbell` network.

`Slalom.` This network is shown in figure 8. The number of backbone networks is customizable. The purpose of this network is to evaluate the end-to-end QoS behavior, when the DiffServ traffic crosses several autonomous systems.

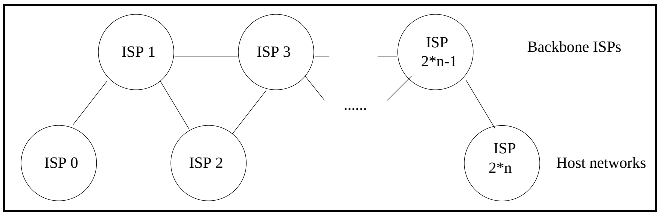

Figure 8: The `Slalom` network.

We have presented the context, terminology, assumptions, and structure of the simulations. In the next section we will present the simulation results.

4 Simulation Results

First we model the adaptive reservation scenario and measure end-to-end QoS. Then we compare the results with a limited notification scenario. We identify the 'dumbbell' problem when using a naive approach, and propose an improved solution.

4.1 The Adaptive Reservation Scenario

As mentioned before, a valuable service in the adaptive reservation scenario can only be achieved using massive overprovisioning. We used concrete numbers for Frame Relay overprovisioning from [FH98b]. There, a Frame Relay provider would conduct network capacity management on a weekly basis. They provision new trunks between Frame Relay switches when trunk utilization exceeds 50 percent. The provider will reimburse a user if the delivery success rate is below 99.8 percent. This maps nicely to a DiffServ simulation where the corresponding overprovisioning is 100 percent. Thus, if a broker measures, that outgoing DiffServ traffic exceeds 50 percent of the agreed value in the appropriate SLA, it will renegotiate the SLA. Using only a medium traffic fluctuation our simulation showed that 99.87 percent of the injected DiffServ traffic reached the destination. This seems to be an encouraging result because it shows that the coarse grained nature of the simulator can still produce appropriate results, and because the end-to-end QoS in this scenario is economically interesting. However, it cannot be assumed, that all ISPs will want to deploy such a high overprovisioning. Furthermore, measurements with larger traffic fluctuation and with more intermediate ISPs showed a poorer end-to-end behavior.

Figure 9 shows a simulation of 100 rounds on the Slalom network with 9 backbone ISPs and 10 host networks. There are therefore 90 different aggregated flows. A total amount of 200 traffic units[3] is injected into the network at each simulation round. The fluctuations of the flows is high here. This means that between two rounds, some aggregated flows will shrink massively, while others will grow. The brokers arrange for an overprovisioning of 20 percent.

At the beginning of the simulation, no SLAs were set up, thus there is no reservation. All DiffServ traffic generated from the host networks is therefore not policy-conform and is shaped. After the 10th round, the content of the SLAs is adapted reasonably and the shaping reaches a stable level. Figure 9 shows the total amount of shaped traffic. Furthermore, the reservation and usage is shown as *average per channel*. Here, they nicely show the 20 percent overprovisioning. As we see in this example, there is a massive loss (shaping) of DiffServ traffic (about 20 percent) in the adaptive reservation scenario, because of an insufficient overprovisioning, heavy traffic fluctuations and a large number of intermediate ISPs.

---

[3] Given the coarse grained structure of the simulator, it would be misleading to use concrete traffic units. Furthermore, the units used here allow a nice integration into the figures.

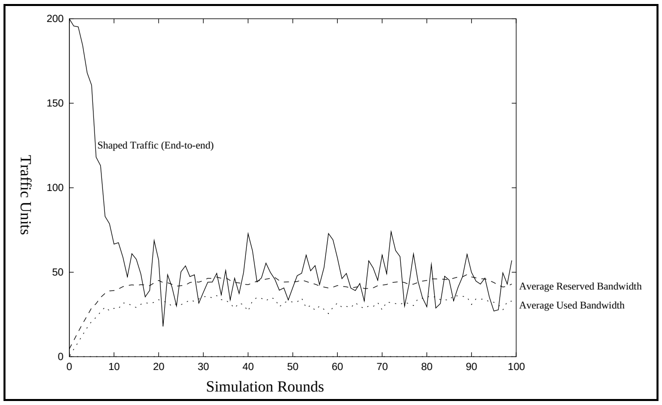

Figure 9: Adaptive reservation with strong fluctuations.

4.2 Limited Notification Scenario

In the limited notification scenario, a broker only notifies and reserves upon significant notifications. There are two kind of problems here. The first is the 'dumbbell' problem, named after the network type that reveals this problem. The other problem is that of the missing destination information in notifications. The next sections describe the problems and propose particular solutions.

4.2.1 The Dumbbell Problem

The first approach for limited notification was to see the notification and reservation as one process. Thus, a broker reacts upon reservation requests by checking its outgoing SLAs and propagating reservation requests, if necessary. In this approach, the broker includes a reservation threshold. If a new inbound reservation causes the reservation on an outbound SLA to exceed this threshold, the broker would issue a new reservation there, before accepting the inbound request. The threshold effectively limits the number of notifications. However, it can have severe impact on the end-to-end QoS as the following simulation run indicates:

In the dumbbell network of the simulator (presented in figure 7), the host networks have only one channel to an access ISP. Using the naive limited notification approach, the host networks reserve a constant amount of DiffServ traffic which suffices all their future needs. Although the weight of the traffic sent for the different destinations changes during the simulation, the total amount of the traffic a host network presents to its backbone ISP stays within the SLA. However, since the traffic distribution scheme of each host network changes, the traffic going through the bottleneck channel between the backbone ISPs may also change. Unfortunately, since the host networks don't reserve new bandwidth, there is no notification sent, and thus no renegotiation of the SLA between the backbone ISP takes place. Consequently traffic is shaped at the bottleneck channel. Figure 10 shows the situation for the `Dumbbell` network with four host networks on each side. Only in the first round, when no reservation is set up at all, notifications are exchanged. Then, no notification is sent at all for the reason mentioned above. Therefore, as reflected in the figure, the reservation stays constant. Subsequently, traffic is shaped without hope for the better.

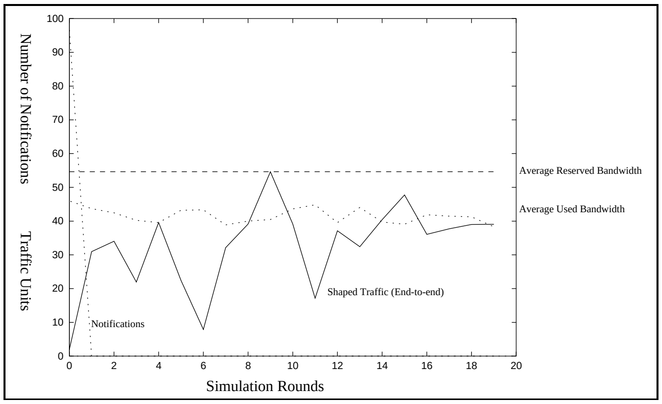

Figure 10: The dumbbell problem.

4.2.2 Lack of Destination Information in Notifications

One approach to limit the notifications is to use one notification to cover several subsequent aggregated flows. Usually, when host networks set up SLAs these SLAs should last some time, thus covering several subsequent flows. However, in that case the notification of such an SLA cannot (in general) include the information of the destination of these flows. There are some special cases however, such as virtual private networks (VPN)[FH98a]. If a host network wants to establish a QoS enabled VPN [BGKK99] it could set up an SLA describing the VPN requested. Usually, the VPN peers are known in advance, such as a company's head-quarters and its branch offices. Therefore, the notification of a new QoS VPN can lead to SLAs that cover several aggregated flows and can include their destination information.

4.2.3 Proposed Solutions

For the two presented problems with limited notification we propose several solutions and show their viability by simulation. The dumbbell problem can be addressed by decoupling notification from reservation. The dumbbell problem occurs, because necessary notifications are not propagated. The notification chain was interrupted, because it did not lead to a reservation in some place. For the problem concerning the lack of destination information we propose the use of exponential estimation based on measurements.

Decoupled Notification Limitation Mechanism. The decoupled notification limitation mechanism is only a small extension to the presented reservation threshold mechanism. Here, the notification is not directly coupled to a reservation. Upon the reception of a notification, that announces DiffServ traffic on an incoming channel, the bandwidth broker reacts according to the following scheme:

- Estimate the impact on the local network.
- Estimate the impact on the outgoing channels. Use destination information if provided.
- Use the estimation and a *reservation threshold* to determine whether to reserve bandwidth (renegotiate the SLA).
- Use the estimation and a *notification threshold* to determine whether to notify other bandwidth brokers. Typically, this threshold is lower than the reservation threshold. Furthermore, the ISPs should all agree on the value of this threshold.
- Use a *minimal notification size* threshold that stops the propagation of notifications concerning only small changes of DiffServ traffic. Such small notifications might occur when estimating the impact of incoming notifications in absence of destination information (see next paragraph).

We also propose to use adaptive reservation and overprovisioning to smooth out the coarse grained nature of the limited notification approach. For the estimation of the size of the needed reservation and notification in case of missing destination information we propose to use the measurements described in the next section.

Exponential Estimation. An ISP with n channels ($n > 1$) can use a distribution matrix D ($n \times n$ matrix). The entry d_{ij} of the matrix D contains the probability that DiffServ traffic coming in on channel i will leave on channel j. Initially, D

contains equal probabilities. However, under the assumption that no routing loops occur, no traffic will leave the ISP the same way it entered it. Furthermore, as mentioned before, the ISPs do not act as traffic sinks. Thus the initial D is:

$$d_{ij} = \begin{cases} 0 & : \quad i = j \\ \frac{1}{n-1} & : \quad i \neq j \end{cases}$$

Periodically, the ISP can compile measurements of DiffServ traffic into the matrix M, where m_{ij} contains the amount of traffic measured, that entered the network from channel i and left it through channel j. The matrix M can be used to update the matrix D in the following way:

$$D_{\text{new}} = \alpha D_{\text{old}} + (1 - \alpha)\text{normRows}(M)$$

Here $\alpha \in [0..1]$ expresses, to what extend the old estimation is still valid after new measurements. In the simulations, α was set to 0.5. The $normRows()$ function normalizes the absolute traffic measurements to relative values:

$$\text{normRows}(m_{ij}) = \frac{m_{ij}}{\sum_{k=1}^{n} m_{ik}}$$

To estimate the impact p on an outbound channel j of a notification about DiffServ traffic of the amount a coming from channel i we can simply calculate $p = ad_{ij}$.

The next section shows, how using such exponential estimation together with the extended limited notification mechanism improved the DiffServ performance in the simulation.

4.2.4 Improved Simulation Results

Without having the destination information of aggregated flows, there are more unknown factors, and there need to be more notifications. However, this more realistic scenario is feasible and reasonable as the following example will show. Figure 11 shows the performance under the same conditions as the example for the adaptive reservation scenario (figure 9). Even though we have up to nine intermediate ISPs for a flow, high traffic fluctuation, little overprovisioning (20 percent), and the destination information is not included in the notifications, the performance is reasonable. The percentage of DiffServ traffic that is shaped is only 11 percent of the total amount of DiffServ traffic presented to the network.

In the first rounds of the simulation, many notifications are necessary to set up the SLAs, but soon the notification limitations restrict the number of notifications to a reasonable level.

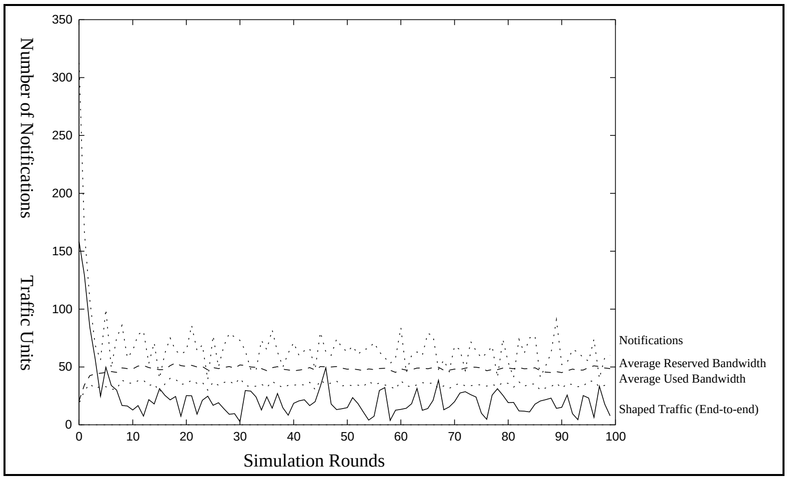

Figure 11: Performance of proposed solution.

If we assume the special case, when the destination information is included in the notifications (e.g. for VPN flows) the result is even improving. Figure 12 depicts the simulation results in this case, using the same harsh network conditions as in the previous example. The shaping decreases to 8 percent of the total DiffServ traffic and there are also less notifications necessary.

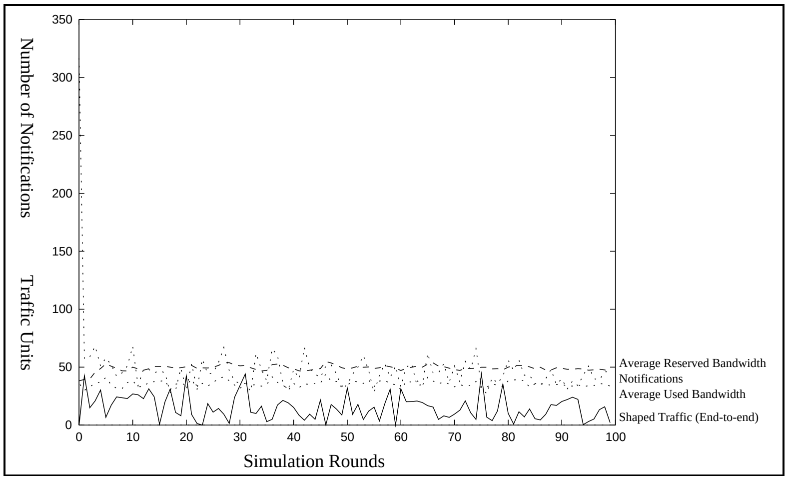

Figure 12: Proposed solution using destination information.

5 Conclusion

Besides of the definition of per-hop behavior of differentiated services in the Internet, there is a need to investigate in the control level of the DiffServ architecture. The control level consists mainly of bandwidth brokers that use signaling between each other, ideally to establish end-to-end quality-of-service. The simulations presented try to explore the main trade-off of the DiffServ control architecture which is between scalability and end-to-end QoS guarantees. We worked out the following conclusions:

- An adaptive reservation mechanism based on measurements is a light weight solution, but cannot be used to provide reasonable end-to-end guarantees. Note, that [SLCL99] shows that even very conservative provisioned DiffServ cannot satisfy user requests under all circumstances.

- A fine grained end-to-end notification strategy between bandwidth brokers breaks the scalability of the DiffServ architecture and is thus not desirable.

- A limited notification scenario can encounter two major pitfalls:
  - The ’dumbbell’ problem, where the notifications do not reach the bottleneck channel, and thus cannot trigger the needed reservations.
  - The ’missing destination information’ problem, where an SLA covers several future flow aggregations in advance.

- Nevertheless, limited notification is a viable way to reduce the number of notifications thus being scalable, but keeping reasonable end-to-end behavior. The decoupling of notifications and reservations, a set of few thresholds as well as a traffic estimation mechanism produce encouraging results in the simulation.

- Services such as a virtual private network service, that allow for the setup of SLAs describing large flow aggregations and that include destination information are beneficial for a limited notification DiffServ control mechanism.

This paper showed, that the limited notification approach to the bandwidth broker signaling is a favorable option. However, the results were only produced by a coarse grained simulation. The next section lists, what future work has to be done in the area of bandwidth broker signaling.

Future Work. The control level of the DiffServ architecture bears many subjects to current research. These topics include the business model, the security architecture necessary [KM99], a design for the monetary transactions involved and SLA routing [Fan99].

We want to deploy the results of this paper in a more general service broker architecture that we proposed in [BGKK99]. An implementation of bandwidth brokers using the limited notification approach is planned. The simulation results of the proposed solution are encouraging but far from perfect. Nevertheless, they seem to indicate that by putting more intelligence to the bandwidth brokers, the end-to-end quality can be further improved without decreasing the scalability of the architecture. Such intelligence could include an optimized set of rules, instead of few simple thresholds. By finding such an optimum, a deployable quality-of-service architecture for the Internet can finally become a reality.

Acknowledgments. This work has been performed in the framework of the project Charging and Accounting Technology for the Internet - CATI (CAPIV 5003-054559/1 and MEDeB 5003-054560/1) which is funded by the Swiss National Science Foundation, SNF, Berne, Switzerland. The authors would like to acknowledge contributions of their project colleagues F. Baumgartner, G. Fankhauser, I. Khalil, S. Leinen and B. Stiller.

References

[BBC$^+$98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services, 1998. RFC 2475.

[BCS94] R. Braden, D. Clark, and S. Shenker. Integrated services in the Internet architecture: an overview, 1994. RFC 1633.

[BGKK99] T. Braun, M. Günter, M. Kasumi, and I. Khalil. Virtual private network architecture. Technical Report IAM-99-01, University of Berne, 1999. http://www.iam.unibe.ch/~rvs/publications/.

[Bri98] British Columbia Institute of Technology. Bandwidth broker high level design, 1998. http://www.merit.edu/working.groups/i2-qbone-bb/.

[BYF$^+$98] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, K. Nichols, and M. Speer. A framework for use of RSVP with diffserv networks, 1998. IETF work in progress.

[Fan99] G. Fankhauser. Service level agreement traders and cost-based routing for differentiated services. Technical Report 59, ETH Zürich, 1999. http://www.tik.ee.ethz.ch/~gfa/.

[FH98a] P. Ferguson and G. Huston. What is a VPN - part I. *The Internet Protocol Journal*, 1(1), 1998.

[FH98b] P. Ferguson and G. Huston. What is a VPN - part II. *The Internet Protocol Journal*, 1(2), 1998.

[KM99] H. Kneer and Ch. Matt. CATI security architecture. to be published, 1999.

[Lei99] S. Leinen. IP resource reservation: Implementations. http://www.switch.ch/tf-tant/rsvp/implementations.html, 1999.

[MBB$^+$97] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang. Resource reservation protocol (RSVP) version 1 applicability statement some guidelines on deployment, 1997. RFC 2208.

[ROT$^+$98] F. Reichmeyer, L. Ong, A. Terzis, L. Zhang, and R. Yavatka. A two-tier resource management model for differentiated services networks, 1998. IETF work in progress.

[SBGP99] B. Stiller, T. Braun, M. Günter, and B. Plattner. Charging and accounting technology for the Internet. 4th European Conference on Multimedia Applications, Services, and Techniques, 1999. to be published.

[SLCL99] N. Semret, R. Liao, A. Campbell, and A Lazar. Peering and provisioning of differentiated Internet services. Technical report, Dept. of Electrical Engineeering, Columbia University, 1999. http://comet.columbia.edu/publications/techreports.html.

[VER98] P. Varaiya, R. Edell, and B. Rupp. Measuring what Internet users value: the INDEX project. http://www.INDEX.Berkeley.EDU/public/index.phtml, 1998.