

Provision of deterministic services for voice over IP using priority queues

J.-A. Müller^{a*}, K. Irmscher^a

^aDepartment of Computer Science,
Leipzig University, Augustusplatz 10-11, 04109 Leipzig, Germany

This paper discusses an approach for resource allocation and management in IP networks, particularly in the context of IP telephony. We show that it is possible to provide deterministic real time services without substantial changes to the current Internet infrastructure using static priority scheduling. All IP telephony traffic is mapped to (unidirectional) virtual channels that allow simple aggregation schemes and subdivision in two parts. We present a calculus to compute the effective bandwidth needed to serve each packet in a service specific time for static priority schedulers. The calculus has been reviewed by simulations. It is used for access control purposes and has the benefit that it can be applied to each node of a network not depending on the other nodes. The virtual channels can logically be subdivided in two parts. Thus, signaling does not need to run from one end to the other, but from both ends simultaneously to the point of aggregation in between. It is shown that the approach can fulfill the requirements of a network build from campus networks connected via a backbone network.

1. Introduction

Today's Internet and its associated Intranets are continuously evolving networks that provide one minimalistic service: the best-effort datagram delivery. While this service allows routers to serve datagrams in a stateless manner, thus making the Internet scalable and robust, it has its cons. It is not possible to guarantee delivery in time or at least delivery at all, as telephone networks use to do. But with the evolvement of the Internet, new applications are developed that need better datagram services to run. IP telephony is one of these applications that require lossless real time delivery.

[1] named IP telephony as application driving the need for substantial changes in the current Internet infrastructure. To address these substantial changes, a wealth of techniques and mechanisms have been developed for packet scheduling, buffer management and signaling. The drawback of most of such approaches is the need for explicit changes of the router hardware or at least their software. Apart from the economical efforts necessary to replace probably all routers, it has to be remarked that these changes add complexity to the network, thus reduce its reliability and scalability. Even if the efforts to overcome

*This work is related to the project 'CoS Konzepte im Gigabit Wissenschaftnetz' (CoS) sponsored by the DFN Verein (Deutsches Forschungsnetz), the german research network provider. The project's objective is the evaluation of a prototypical implementation for providing quality of service by using a class of service concept.

the limits of current techniques look promising, we cannot expect their availability in wider areas of the Internet in the short term. Nevertheless, we expect substantial changes to the Internet infrastructure in the mid- or at least long term.

The aim of this paper is to provide a solution for deterministic services for real time traffic as for example IP telephony applications. We introduce a model of absolute differentiated services that is based on static priority queuing, available in most of the currently used Internet routers and LAN switches. Thus, it does not require changes of existing routers and could be available in the short term.

The paper is organized as follows. An introduction to our model is given together with an analysis of the requirements of IP telephony applications and traffic management techniques. Section 2 provides a calculus that is used as access control mechanism to allow guarantees on upper bounds on queuing time for the better datagram delivery services. This is followed by a presentation of simulation results to emphasize the calculus and a discussion on the applicability. We close with an outlook on advantages in signaling and limitations in scalability.

2. A model based on virtual channels

Since the originality of our work is to develop an approach providing deterministic services *without* changing the network components, we have to define restrictions to the data streams the realtime service is designed for. For the same reason and due to problems with per flow management seen in IntServ, our model has to support out of band management, and more important, it should work with aggregated flows instead of single streams.

Our model is based on (unidirectional) constant packet rate virtual channels similar to virtual channels known from ATM networks. One virtual channel represents a data stream of constant size packets generated with constant inter packet delay, flowing from one source to one destination node. As shown in figure 1, a virtual channel can be divided into parts or can be seen as concatenation of two or more virtual channels. This allows to introduce aggregates of virtual channels using the same type and path (size and rate of packets are equal). Thus, we can reduce the states to keep per queue to a service definition and the number of virtual channels served by this queue. Figure 2 illustrates an example network, where each of the 54 virtual channels is divided into three parts reaching a reduction to 24 virtual channels to be managed.

The model of virtual channels with constant packet rate is motivated by the analysis of common audio codecs. Most of those codecs produce packets of constant size with a constant rate of 25pps, 37.5pps or 50pps. The proposed deterministic services guarantee the end to end delivery of each datagram in a virtual channel with less queuing time than their inter packet delay (Δ). Thus, we avoid aggregation of bursts inside a virtual channel. For each interval $t \leq \Delta$ the amount of datagrams received by any queue in the path of a virtual channel has an upper bound of two packets.

We exclude the aggregation of virtual channels with different service parameters to reduce the keeping of states per queue to a minimum of allowed queuing time and the number of virtual channels to serve. But the calculus, we introduce in the next section, allows to conceive one virtual channel with 50pps and 160 Byte per packet for example as two virtual channels of 25pps with packet size 160 Byte or eight virtual channels of

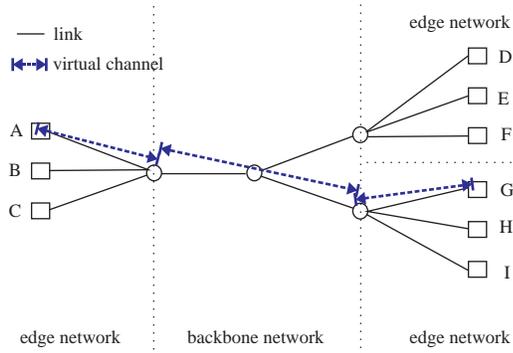


Figure 1. A virtual channel from A to G as concatenation of three virtual channels.

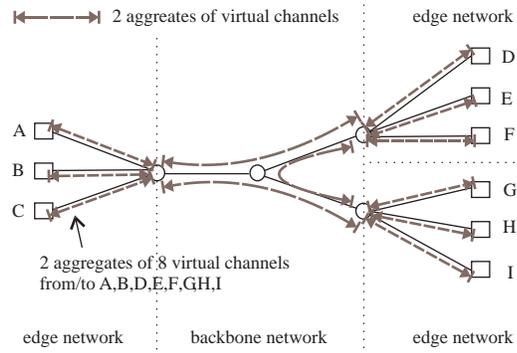


Figure 2. Division and aggregation of virtual channels allows to reduce the states to keep

25pps with a minimum policed unit (MPU) of 40 Byte and an maximum transmission unit (MTU) of 160 Byte. It is also possible to map channel rates and/or packet sizes that are not a multiple of the virtual channel parameters. Those mappings do not match perfect. Thus, the virtual channel parameters should be chosen carefully.

3. Calculus to guarantee an upper bound on queuing time

According to [2], delay is a function of network diameter and use. This can be reduced to a computation of the effective bandwidth R needed to serve each virtual channel's datagrams with a queuing delay having an upper bound (τ). This computation is used to determine, whether the number of virtual channels that can be served by a queue with a delay of τ or less. While the service rate μ of a scheduler/link is typically fix and we defined τ as an upper bound only the use is variable.

Formula 1 provides an example of this computation for one FIFO queue serving two ingress links. We derived this formula from an example in [3] and adapted the notation to our needs. The aggregates of virtual channels streaming in are defined by an average rate r of one virtual channel, the number of virtual channels aggregated n_i ($i = 1 \dots 2$), the packet size U and the peak rate p_i of the aggregate. The maximum burst time of an aggregate is given by $\theta_i = \frac{(n_i-1)*U}{p_i-(r*n_i)}$. θ_i is used as index function. We assume that $\theta_1 \leq \theta_2$.

$$\mu \geq \max \left\{ \frac{n_1U + n_2U}{\tau}, \frac{2U + (p_1 + p_2)\theta_1}{\tau + \theta_1}, \frac{U + n_1U + (n_1r + p_2)\theta_2}{\tau + \theta_2}, r(n_1 + n_2) \right\} \quad (1)$$

We start our discussion using one FIFO queue p serving n input links $I(s_p) = \{i_n(s_p)\}$ with an ingress rate of $C_n(s_p)$ and $i_n(s_p) \in I(s_p)$ channels entering the node to be served by p . The peak rate of each aggregate is limited by the ingress rate, $C_n(s_p)$ is their worst case representation. Since we do not know the history of an aggregate, the worst case has to be assumed. Thus, we assume for all input links $i_n(s_p) \in I(s_p)$ that $\sum_{i_n(s_p) \in I(s_p)} i_n(s_p)$ packets enter the queue p as a burst of packets.

Incoming packets shall be served by a special service s_p with the following parameters: An allowed channel rate of $r(s_p)$ in packets per second, a minimum policed unit of $M(s_p)$ and a maximum transmitted unit of $U(s_p)$. We define the inter-packet delay $\Delta(s_p)$ as $\frac{M(s_p)}{r(s_p)}$ and the upper bound for the maximum transmission time in the whole network for packets with a packet size of $U(s_p)$ as $\delta(s_p)$. The generalized form of formula 1 applicable for FIFO queues with any number of ingress links is shown in formula 2 using our final notation.

$$R(I(s_p)) = \max \left\{ \frac{M * |I(s_p)|}{\tau(s_p)}, r * \sum_{n=1}^{|I(s_p)|} i_n(s_p), \max_{n=1 \dots |I(s_p)|} \frac{b_n^* + r_n^* * \theta_{i_n}(s_p)}{\tau(s_p) + \theta_{i_n}(s_p)} \right\} \quad (2)$$

$$b_n^* = \sum_{k=1}^{|I(s_p)|} (\nu_k^n(U(s_p), M(s_p) * i_k(s_p))) \text{ with } \nu_k^l(x, y) = \begin{cases} x, & \text{if } k \neq l \wedge \theta_{i_k}(s_p) \leq \theta_{i_l}(s_p) \\ y & \text{else} \end{cases}$$

$$r_n^* = \sum_{k=1}^{|I(s_p)|} \nu_k^n(i_k(s_p) * r(s_p), C_k(s_p))$$

If all bursts enter simultaneously and cannot be served in less than $\Delta(s_p) - \delta(s_p)$ the arrival rate is higher than the service rate and bursts get aggregated. Thus, we cannot predict the resulting virtual channels and their queuing times. We therefore define $\tau(s_p) < \Delta(s_p) - \delta(s_p)$ that determines the maximum queuing time for p . $\tau(s_p)$ represents an upper bound for the inter-packet delay variation resulting from queuing in p . If the n -th datagram of one virtual channel is served as last packet of a burst after $\tau(s_p)$ and the n -first datagram of the same virtual channel enters the queue in an empty state, the inter-packet delay of those two packets considered would be $\Delta(s_p) - \tau(s_p) > 0$. As a consequence, queue p is empty for $\Delta(s_p) - \tau(s_p)$ for any interval $\Delta(s)$.

We now extend our discussion to a network, by distributing the maximum queuing time $\tau(s)$ to the queues of the network. For each path through the network the sum of all local upper bounds of queuing times must not rise $\tau(s_p)$. Due to the *pay bursts only one phenomenon* [3] it is ensured that none of the packets will be queued longer than $\tau(s_p)$.

When the maximum queuing time per queue becomes shorter, less virtual channels can be served per queue, thus per network. This is unavoidable without traffic shaping mechanisms. Although we found out that the number of virtual channels that can be provided in networks up to a diameter of 15 nodes is sufficient to provide useful IP telephony services. The diameter of 15 results from allowing roughly 1ms and 2ms resp. maximum queuing time per node with $\tau < \frac{1}{50pps}$ and $\tau < \frac{1}{25pps}$ respectively.

We will now extend the formula 2 from simple FIFO queues to priority based schedulers with $N + 1$ different priorities ($N > 1$) starting from zero as lowest priority value. The best-effort datagram service should be served by the queue of the lowest priority $p = 0$. We assign one service definition (referenced as s_p) per priority level p ($0 < p \leq N$) with the simple rule that a service providing lower bounds on queuing time $\tau(s_p)$ gets a higher priority.

$$B_{I(s_p)} = \begin{cases} m, & \text{if } p = N \\ m + \sum_{k=p+1}^N \left(\left\lceil \frac{\Delta(s_p)}{\Delta(s_k)} \right\rceil * M(s_k) * \sum_{l=1}^{|I(s_k)|} i_l(s_k) \right), & \text{if } p < N \end{cases} \quad (3)$$

$$R^{prio}(I(s_p)) = \max \begin{cases} \frac{M * |I(s_p)| + B_{I(s_p)}}{\tau(s_p)} \\ \sum_{k=p}^N r(s_k) * \sum_{n=1}^{|I(s_k)|} i_n(s_k) \\ \max_{n=1 \dots |I(s_p)|} \left(\frac{B_{I(s_p)} + b_n^* + r_n^* * \theta_{i_n}(s_p)}{\tau(s_p) + \theta_{i_n}(s_p)} \right) \end{cases} \quad (4)$$

Formula 2 is valid for a single FIFO queue scheduler but not for FIFO queues within a priority scheduler because additional dependencies occur. Due to the non-preemptive nature of schedulers in packet switching networks, each queue has to wait for the packet currently scheduled. In the worst case this packet has the size of the maximum transmission unit (referenced as m) allowed by this scheduler and the scheduling of this packet has started as a packet with higher priority was enqueued. Additionally a queue can not be served if there are packets to be served in queues with higher priority. Thus, for a queue with a priority p we have to serve the packets that appear within a given interval $\tau(s_p)$ in queues with higher priority relative to p too. These packets and the packet currently scheduled have to be considered in the calculus as additional burst $B_{I(s_p)}$. Formula 3 allows to calculate $B_{I(s_p)}$, thus we can write formula 4 to calculate the effective bandwidth $R^{prio}(I(s_p))$ for a FIFO queue with priority p within a priority scheduler. The effective bandwidth R needed to serve all datagrams assigned to higher priority queues ($p > 0$) is the maximum of the effective bandwidth needed per priority queue ($R = \max_{p=1..N} R^{prio}(I(s_p))$).

Datagrams in lower priority queues, including the best-effort queue, are allowed to use the spare times of higher priority queues. These spare times have to be relatively long for one reason. Usually the maximum queuing time τ of one queue in a network will have a much smaller value than the inter-packet delay Δ . Since we assume the worst case, a queue is at minimum $\Delta - \tau$ times empty. These spare times are at least used by the best-effort queue. If the average burst time of this queue is known, it is possible to estimate the average queuing times using formula 3 to calculate the burst that is queued before any best-effort datagram (in the worst case).

Finally the number of channels that can be served by a queue depends on the queue's buffer. Each queue should at least have a buffer of the size of the maximum allowed backlog ($\frac{\mu}{\tau(s_p)} + M(s_p)$) that may occur when any queue with higher priority is empty. If that is not possible, the equivalent capacity (to the buffer size) should be computed in addition to the effective bandwidth and the maximum of both has to be used (see [3]).

4. Discussion

The following section gives a discussion of our model. We start with the number of virtual channels that can be served by typical bottleneck links. Afterwards we present results derived from simulations.

Label	Lan	Intra	Inter	BN1	BN2	BN3
Ingress rate	100	1000	2440	2440	1000	1000
Link rate	100	622	2440	622	622	100
Ingress links	5	4	4	1	1	1

Table 1
Bottlenecks used for the discussion (all rates in Mbps)

Codec	GSM	GSM	G.728	G.728	G.726(24)	G.726(24)	G.726(32)	G.726(32)	G.711	G.711
pps	50	25	50	25	50	25	50	25	50	25
Lan	180	290	165	240	135	145	110	120	65	70
Intra	1192	1808	1116	1504	892	908	744	752	444	450
Inter	5304	7980	4976	6648	3980	5988	3316	3340	1988	1994
BN1	1344	2037	1260	1697	1008	1025	840	852	503	508
BN2	2495	3778	2339	3140	1870	1960	1558	1660	933	943
BN3	163	262	153	218	122	262	101	109	60	65

Table 2
The amount of IP telephony data streams that can be served in 1ms

4.1. Bottleneck links

Our discussion is based on the analysis of bottleneck links typical for local networks, backbone links in Intranets, ISP networks and uplinks to those networks. As our work relates to a project with the DFN e.V., we mainly refer to the German Research Network [6]. Table 1 shows the ingress and service rates we use together with the number of ingress links. If there is more than one ingress link, we assume that the number of datagram streams is distributed equally to these links.

Table 2 shows the number of IP telephony data streams for each bottleneck depending on the codec. We have calculated the values for virtual channels with 50pps and 25pps.² A maximum transmission unit allowed by the scheduler of $m = 1500\text{Bytes}$ was assumed. Obviously more virtual channels can be served when the packet size decreases. Not surprisingly we found out, that the ratio between reduction of packet size and grows in number of allowed channels is nearly proportional. Also not surprising is the strong dependency of the ratio between the sum of the ingress rates and the service rate to the number of channels that can be served. While bottleneck BN1 has a ratio of 2440:622 it allows roughly 46 percent less virtual channels than BN2 with a ratio of 1000:622. We got similar results for the other bottlenecks, leading to the general statement, that a fairer ratio allows more virtual channels to be served.

We also discuss the results by value. If a local area network has a bottleneck link of 100Mbps, it is shown that at least 65 calls using G.711 with 50pps can concurrently be served. Usually a local area network including such a bottleneck has not more than 100 users, so it is possible to use the high quality codec G.711 for all calls. Furthermore, it is possible to switch to a codec with lower quality and support up to 290 concurrent calls. As table 1 shows, we got similar high values for a bottleneck link of 100Mbps feeding the

²If the audio codec does not support 25pps (see table 1), we calculated with two virtual channels for one audio stream sending 50pps.

Bottleneck	τ	Priority	Codec	VC	Calls
Intra	1ms	2	GSM	50	1192
	2ms	1	G.726(24)	25	380
Intra	1ms	2	G.726(24)	50	892
	2ms	1	GSM	25	764
BN2	1ms	2	GSM	50	2495(1000)
	2ms	1	G.726(24)	25	0(1007)
BN2	1ms	2	G.726(24)	50	1870(1000)
	2ms	1	GSM	25	0(1394)

Table 3

The amount of IP telephony data streams that can be served in 1ms and 2ms resp. using two better services with different priorities

queue with a peak rate of 1Gbps. Thus, it is not only possible to send 200 calls (GSM codec) via a bottleneck of 100Mbps to a backbone network, but also to receive 200 calls from a 1Gbps backbone link.

Assuming a backbone network of 1Gbps is connected with 622Mbps to the German Research Network backbone. Thus, it is able to transport from 933 calls to 3778 calls. Usually a university, e.g. the Leipzig University, with approximately 2300 staff members, will need so many concurrent outgoing calls. While the German Research Network is a Gigabit backbone network with mostly 2.44Gbps links it does not have a bottleneck in its typical meaning. Thus, we cannot say it allows up to 1988 concurrent high quality calls or up to 7980 concurrent GSM quality calls, since most of the traffic remains local or uses different paths. Since the core network backbone has currently 10 nodes with at maximum one node between two others, we assume a capacity of at minimum five times of the 'Inter' bottleneck link in table 2. This allows at least 9990 concurrent calls. It is more likely that the GSM codec with up to 39900 concurrent calls is used. This allows more than 100 concurrent calls per member³ connected to the German Research Network.

We also investigated the use of two different service definitions instead of one. As already mentioned it is possible to combine virtual channels to meet the requirements of the IP telephony data stream. But for the reasons of shorter delay one might want to define a constant packet rate service using 50pps limiting the queuing time $\tau(s_{p+})$ to less than 20ms and a second constant packet rate service using 25pps, with higher boundary ($\tau(s_{p-}) > \tau(s_{p+})$). We analyzed the effects to the lower priority service using the bottlenecks 'Intra' and 'BN2' and presented our results in table 3. It is not relevant whether the codec with the higher packet size has the higher priority or not. The relations between packet size and the maximum calls possible still remain. Nevertheless the number of channels that can be served by the lower priority service (with 25pps) depends on the high priority service. The results for 'BN2' show that the bursts in a higher priority queue do not allow the lower priority queues to be served in 2ms or less. A 622Mbps link needs 2.408ms to serve 2495 GSM datagrams which a 1Gbps link delivers in 1.497ms. Thus, the higher priority queue is served as supposed. But in the worst case the lower priority queue is blocked for 2.408ms.

³The Deutsches Forschungnetz e.V. has approximately 370 members.

5. Simulation results

In order to verify our analysis of the previous section simulations have been done with the ns2 simulation software[7]. We used the class based scheduler configured to behave like a priority scheduler with strict priority queuing. Each class was assigned a unique priority and it was allowed to borrow unused resources from any other class. None of the classes got an guaranteed allotment of the service rate. The IP telephony datagram streams were simulated by CBR sources using UDP. We assigned a continuous FTP source to each CBR source generating traffic using the best-effort datagram service. A maximum transmission unit of 1500Byte and window sizes between 15 and 50 packets were used for such FTP sources.

Three priority classes were implemented to allow the two better services discussed in the previous section. The queues for these better services were configured to store only the maximum number of packets expected depending on $\tau(s_p)$. The size of the best-effort queue was set to 200 packets. We expected the higher priority queue to serve each packet with not more than 1ms queuing time. The lower priority queue should serve each packet not more than 2ms queuing time

We performed simulation runs for each of the configurations discussed in the previous sections (see table 1, 2, and 3). The CBR sources were configured to send packets simultaneously, thus generated worst case bursts. All CBR packets were served in time and none of those packets were dropped. The queuing times in the highest priority queue were distributed equally from 0ms up to 1ms. Traffic in the lower priority queue, if any, had queuing times between 1ms and 2ms. Due to our worst case scenario, we observed an interval inside those limits where the queuing times again were distributed equally in the lower priority class. The TCP streams using the best-effort datagram delivery service adapted well to the remaining average service rate.

6. Outlook to signaling and scalability

As discussed in short in section 2, it is a benefit of our virtual channel based approach that we can reduce the state keeping by aggregation and partitioning. But we have not presented a solution on signaling and access control to implement our solution. Thus, we give an outlook to the signaling approach chosen by our project CoS mentioned above.

Again, since we do not want to change the behavior of the network components, signaling has to be out of band⁴. Thus, we virtually have to put a management instance, called bandwidth broker, aside each router or switch. Figure 3 shows a reduced view of the example we will use for our discussion. We assume a backbone network with 51 leaf networks, that for instance could be 50 institutes of a university and one gateway to the German Research Network. Each leaf network is managed by its own bandwidth broker, thus we divide each virtual channel into three virtual channels. This allows the backbone to deal only with the aggregated virtual channels. If we had only one real bandwidth broker as supposed for example in [9], it would have to manage 51x50 virtual channels per better datagram service.

⁴We believe that our model would also work with in-band signaling, for instance with RSVP and aggregated RSVP(see [10,11]).

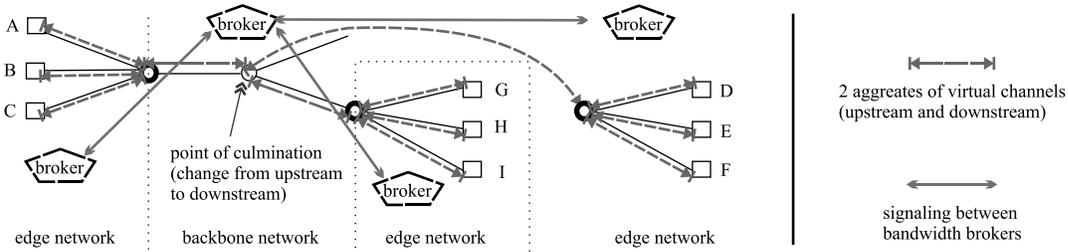


Figure 3. A bandwidth broker managed network divided into edge-networks and backbone

We can simplify that by using another benefit that virtual channels have. It is not necessary to spread signaling from the source to the destination or vice versa. Due to the simple aggregation scheme it is possible to divide the signaling in two parts called upstream and downstream. There is at least one node in the backbone, where both parts of a virtual channel will not be aggregated further. Thus, we can assign one point of culmination to each of the 51x50 virtual channels and again divide them into two. The bandwidth broker in the backbone network therefore has to handle only 102 virtual channels: 51 virtual channels for upstream and 51 virtual channels for downstream.

Apart from the signaling, we have supposed our solution for networks with a diameter of 15 nodes. We propose two approaches to overcome this limitation without changing the implemented infrastructure. Nevertheless we have to add components to the network. Both solutions share the problems of increasing the delay and the need of per call information. Therefore they should be placed only at the gateways between larger networks.

A shaper could be placed in parallel to an uplink between two parts of a network shaping each IP telephony data stream. While the best-effort traffic is transmitted by the usual link, each datagram with priority flag set is routed (using that flag) to the shaper. The shaper could for instance be implemented using a router that supports IntServ techniques.

We could also introduce a proxy at application layer, that acts as a gateway for H.323 and/or SIP conformant calls. It is not uncommon to use a high quality codec for calls within the Intranet and recode the datagrams using a lower quality codec before they are sent to the Internet. This proxy could also reshape the calls. Since this is probably a software solution, it may not scale up very well.

7. Conclusion

We presented an approach to provide deterministic datagram delivery services for IP telephony streams in IP networks using only static priority scheduling. This has been achieved by defining constant packet rate virtual channels with special packet rates. It was shown that data streams generated by common IP telephony applications map well to these virtual channels and we can provide good bounds on queuing delay. Furthermore, it was presented how to use the formula for the effective bandwidth to calculate the

number of virtual channels that can be served by a FIFO queue independent of their path history. We extended the calculus to suit to static priority schedulers and reviewed it by simulations with ns2. A discussion on the usability showed that the number of concurrent calls fits at least the needs of the German Research Network backbone and its connected campus networks. We finally provided an outlook to signaling and scalability to the Internet by example. Additionally the example was used to show that the aggregation of virtual channels allows a remarkable reduction in states to keep.

Future work will include techniques to implement a work conserving reshaper for virtual channels in an aggregate. Thus, reconditioning of virtual channels does not rely on IntServ techniques if scalability is an issue. Secondly, the outlined signaling approach using a bandwidth broker model will be further developed by the CoS project to a applicable prototype.

REFERENCES

1. R. Gurin, V. Peris, *Quality-of-service in packet networks: basic mechanisms and directions*, IEEE Journal on Computer Networks 31 (1999) 169-189
2. A. Charny, J.-Y. Le Boudec, *Delay Bounds in a Network With Aggregate Scheduling*, Proceedings of Quality of Future Internet Services (QofIS 2000), Berlin, Sept. 2000
3. J.-Y. Le Boudec, P. Thiran, *NETWORK CALCULUS: A Theory of Deterministic Queueing Systems for the Internet*, Springer Verlag, LNCS 2050, 2001
4. U. Fiedler, P. Huang, B. Plattner, *Towards Provisioning DiffServ Intra-Nets*, Proceedings of the 9th IEEE/IFIP International Workshop on Quality of Service (IWQoS'01), London, 2001
5. R. Steinmetz, K. Nahrstedt, *Multimedia: Computing, Communications & Applications*, Prentice Hall, 1995
6. DFN Verein, *Der DFN-Verein*, <http://www.dfn.de/dfn/>, Jun 2002
7. K. Fall and K. Varadhan (editors) *ns Notes and Documentation* The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Apr. 2002, <http://www.isi.edu/nsnam/ns/>
8. K. Mochalski, J. Micheel, S. Donnelly *Packet Delay and Loss at the Auckland Internet Access Path* Proceedings of Passive & Active Measurement Workshop (PAM2002), Fort Collins, Colorado, USA, 25-26 March 2002
9. G. Stattenberger, T. Braun *Performance of a Bandwidth Broker for DiffServ Networks*, Proceeding of Kommunikation in verteilten Systemen (KiVS03), Leipzig, Germany, March 25-28, 2003
10. Steven Berson, *Aggregation of Internet Integrated Services State*, Proceedings of the 7th IEEE/IFIP International Workshop on Quality of Service (IWQoS'99), London, 1999
11. J. Schmitt, M. Karsten, L. Wolf, R. Steinmetz *Aggregation of Guaranteed Service Flows*, Proceedings of the 7th IEEE/IFIP International Workshop on Quality of Service (IWQoS'99), London, UK