University of Würzburg
Institute of Computer Science
Research Report Series

A Scalable Protocol Architecture for End-to-End Signaling and Resource Reservation in IP Networks

Michael Menth

Report No. 278 July 2001

University of Würzburg
Department of Computer Science
Am Hubland, D-97074 Würzburg, Germany
menth@informatik.uni-wuerzburg.de

A Scalable Protocol Architecture for End-to-End Signaling and Resource Reservation in IP Networks

Michael Menth
University of Würzburg
Department of Computer Science
Am Hubland, D-97074 Würzburg, Germany
menth@informatik.uni-wuerzburg.de

Abstract

This paper presents a concept of a scalable networking architecture with end-to-end QoS signaling and resource reservation support. It is a synthesis of both the Differentiated Services and the Integrated Services approach. The approach relies on end-to-end resource reservation and takes advantage of traffic aggregation to reduce the number of reservation states in the router MIBs. Signaling costs are further decreased by making overreservation, i.e. bandwidth efficiency is traded for signaling reduction. We give recommendations to dimension the degree of overreservation and compute the resulting tradeoff analytically. The numerical results show that networks running the proposed architecture can be operated efficiently in spite of overreservation.

Keywords: QoS in IP, Signaling, Resource Reservation, Scalability, MPLS, Performance Analysis

1 Introduction

The success of the Internet Protocol (IP) has been overwhelming in the past due to the simplicity of the addressing scheme and the tremendous growth of the world wide web. However, for the support of real-time services like voice over IP (VoIP) or video conference, the shortcomings of best effort IP networks are obvious. The lack of real-time delivery prevents their merge with conventional real-time networks. Therefore, several mechanisms have been introduced to support real-time transport in IP networks but all of them have major drawbacks.

The Integrated Services (IntServ) approach is able to give absolute end-to-end (e2e) quality of service (QoS) guarantees to a micro flow (host-to-host packet flow). For every flow, transmission capacity is reserved in each router along the path from its source to its destination. This requires the allocation of an information state per flow in all intermediate routers. The routers can not handle a tremendous amount of connection states in their management information base (MIB) in real-time and, in addition, they are overloaded with per flow signaling in the presence of many QoS provisioned micro flows. Therefore, IntServ is not suited for networks with many data streams requiring QoS support.

The Differentiated Services (DiffServ) approach defines different treatment for IP packets in a router depending on their DiffServ marking. This scales well because instead of considering many flows only a few traffic classes of relatively differentiated QoS levels are introduced. DiffServ operates on a per packet basis and the absence of the connection concept prevents admission control (AC). Since absolute QoS is a function of available and requested capacity, DiffServ can not provide absolute QoS guarantees.

Many multimedia applications demand for real-time delivery of large media streams but there is no established scalable solution to provide hard e2e QoS guarantees in IP networks. In this paper we present a network architecture that overcomes the scalability problem for e2e signaling and reservation. It is a synthesis of both the IntServ and the DiffServ approach. The architecture relies on e2e resource reservation and uses reservation aggregation to reduce the number of reservation states in the router MIBs. Signaling costs are further decreased by taking overreservation, i.e., bandwidth efficiency is traded for reduction of signaling. The aggregation concept can also be found in [1, 2, 3, 4, 5, 6, 7] and our findings also apply to these protocols.

In the next section, the IntServ and DiffServ approach as well as Multiprotocol Label Switching (MPLS) are described and their drawbacks are pointed out. Section 3 explains the idea for a scalable network architecture and presents two different protocol implementations. We propose an update scheme for aggregate reservations with overreservation. In Section 4 the tradeoff between bandwidth efficiency and signaling cost is analytically computed and a rule of thumb for overreservation is derived. The numerical results of Section 5 illustrate the influence of the update mechanism on the network performance. Finally, the paper concludes that a network running such an architecture can be operated efficiently in spite of overreservation.

2 Protocol Architectures for Qos Support in IP Networks

The Internet Engineering Task Force (IETF) has suggested two main alternatives to equip IP networks with real-time capabilities: the IntServ and the DiffServ approach. In addition, MPLS has been defined in order to facilitate the process of traffic engineering. These concepts are briefly introduced in the following.

2.1 Integrated Services Architecture (IntServ)

The IntServ approach [8] satisfies the QoS demands by making e2e reservations for every micro flow in each router along the path. The Resource Reservation Protocol (RSVP) [9] supports signaling, it performs path discovery, reservation establishment, sender or receiver notification in case of failure, and reservation teardown. When a reservation is set up, each router performs AC and a reservation request only succeeds if the local capacity suffices to serve both the existing micro flows and the new request. This saves the router from being overloaded with priority traffic and QoS can even be enforced during busy hours at the expense of blocked connections. As a result, QoS supported streams see an unloaded network and the remaining bandwidth can be used by best effort (BE) traffic.

The routers need flow specifiers like traffic (T_{spec}) and reservation (R_{spec}) descriptors for every admitted micro flow to record the expected traffic volume and the required QoS. The filter specs help to map IP packets to the respective micro flows and to classify them for the scheduler. The policer uses these data to control the traffic contract and to drop IP packets that are out of profile. The per flow information creates a state in every IntServ router and is stored in the MIB. This has several disadvantages. The IP network loses its stateless property that made it very robust against failures. The administration overhead for setting up, updating, terminating a connection, as well as for forwarding IP packets consumes additional CPU cycles. Lookups have to be done in real-time for every IP packet, so that the MIB is implemented in fast memory. The needed MIB size scales with the number of admitted flows, therefore, IntServ works well if the number of QoS flows is small but it is not likely to run in the core of a network where many streams have to be supported.

2.2 Differentiated Services Architecture (DiffServ)

The DiffServ approach [10] introduces traffic classes. IP packets are classified according to their DiffServ codepoint (DSCP) in the header. They are treated by the routers with a DSCP specific Per-Hop-Behavior (PHB) to realize the service differentiation. At the moment, the Assured Forwarding (AF) and the Expedited Forwarding (EF) PHB groups are defined in addition to normal best effort BE traffic.

DiffServ routers do not perform AC for flows but they drop or recolor IP packets if the preconfigured traffic volume for a certain PHB is exceeded. Traffic conditioners may be used at the network boundaries to limit the traffic volume within the DiffServ domain. In busy hours, the network can either be overloaded with high priority traffic or the transmission of a flow suffers from service degradation at the traffic conditioners that operate on packet level. Unlike in IntServ, blocking some connections in favor of already admitted sessions is not possible and absolute QoS can not be guaranteed.

2.3 Multiprotocol Label Switching (MPLS)

MPLS is a mechanism to allow packet switching instead of routing over any network layer protocol [11]. Packets that share a common attribute create a Forwarding Equivalence Class (FEC) and are forwarded via a label switched path (LSP) by label switching routers (LSR). The first LSR of an LSP puts a label onto the IP packet and the last LSR removes it. A certain capacity can be associated with such a connection to achieve QoS provisioning like in IntServ but a FEC usually consists not of a single micro flow, so we talk about aggregate reservations. The local labels for every LSP are stored in the MIB of the LSRs which introduces also a state per session. Both an extension to RSVP [12] and the Constraint-Based Label Distribution Protocol (CR-LSP) [13] are used for signaling.

MPLS has some features that distinguish it for traffic engineering. Load balancing can be achieved by creating several LSPs with different routes for packets with the same destination. In case of a node failure, fast rerouting repairs the connection within a few milliseconds while the convergence of IP routing algorithms takes in the order of seconds. MPLS is often viewed as modified version of the Asynchronous Transfer Mode (ATM) with variable cell size. But there is a profound difference: ATM exhibits with its virtual connection and virtual path concept two levels of aggregation while MPLS allows for many-fold aggregation using multiple label stacking, i.e. an LSP may be transported over another one.

3 A Scalable Architecture for E2E QoS Signaling and Resource Reservation

IntServ is able to offer hard QoS guarantees because per flow signaling enables AC but the number of reservation states and the amount of signaling do not scale for large networks. DiffServ avoids the scalability problem by providing relative QoS differentiation to a few service classes but due to the lack of signaling, AC is not possible and absolute QoS guarantees can not be given. MPLS itself has hardly any QoS support but it offers capabilities for traffic engineering. In this section we sketch a scalable protocol architecture that gives absolute QoS guarantees and that relies on ideas from IntServ and DiffServ. We discuss the general idea and present two existing protocol solutions.

3.1 Concept

In a large IntServ network, many micro flows share a common subpath of their routes. Their number might be too large to support them on that subpath by IntServ mechanisms. So we grant them an aggregated reservation to support them as a whole and to reduce the states in the intermediate routers to a single one. The RSVP signaling messages for the micro flows are hidden by the aggregating router to prevent them from being processed in the interior nodes. They are recovered by the deaggregating router. The aggregator labels the aggregated packets with a common tag to keep the classification and scheduling mechanisms as simple as in DiffServ. A new flow or a flow update is admitted if the capacity of the aggregate reservation suffices. Otherwise, the size of the aggregate reservation can be increased, if this fails, too, the new request is rejected. Policing is also enforced for aggregate reservations. From the signaling point of view, aggregate reservations do not differ substantially from an e2e reservations. Therefore, they can be aggregated in the same way creating a hierarchical reservation structure (Figure 1). This makes the approach scalable concerning the amount of information states in the router MIBs.

The proposed scheme allows for e2e signaling per micro flow, for AC, and for resource reservation so that absolute QoS guarantees can be granted. Hierarchical reservation aggregation reduces the number of states in intermediate routers and makes the approach scalable even for large networks.

3.2 Protocol Solutions

We present now two different protocol solutions that implement the above explained concept. Both are currently discussed in the IETF.

3.2.1 RSVP Aggregation

In [2], an extension to RSVP is proposed to summarize several RSVP sessions into a new aggregate reservation. The first router changes the IP protocol number in the RSVP control messages of the individual reservations to *RSVP-E2E-IGNORE* such that they are not processed by intermediate routers and the corresponding deaggregator resets the protocol number to *RSVP* (46). The aggregation level is recorded in the router alert option field so that a deaggregator knows which RSVP message has to be set back to *RSVP*. This facilitates the recursive application of that scheme. The aggregator sets the DSCP of the aggregated IP packets to a specific value, such that forwarding within the aggregation region is done only by the corresponding PHB.

3.2.2 MPLS Aggregation

In [1], hierarchical traffic aggregation is achieved using MPLS. An LSP is established with QoS requirements on the way from an aggregating router to a deaggregating router. Both the user and the data plane traffic from aggregated sessions are assigned to the same LSP. Thus, the RSVP control messages are tunneled by MPLS packets, such that they are automatically bypassed at the intermediate routers and no additional mechanisms are required to reveal the RSVP control messages at the end of the tunnel. The traffic may be additionally mapped to DSCPs to reduce the number of classification and scheduling states. Since FECs do not differ substantially from micro flows with regard to signaling, the scheme can be applied recursively.

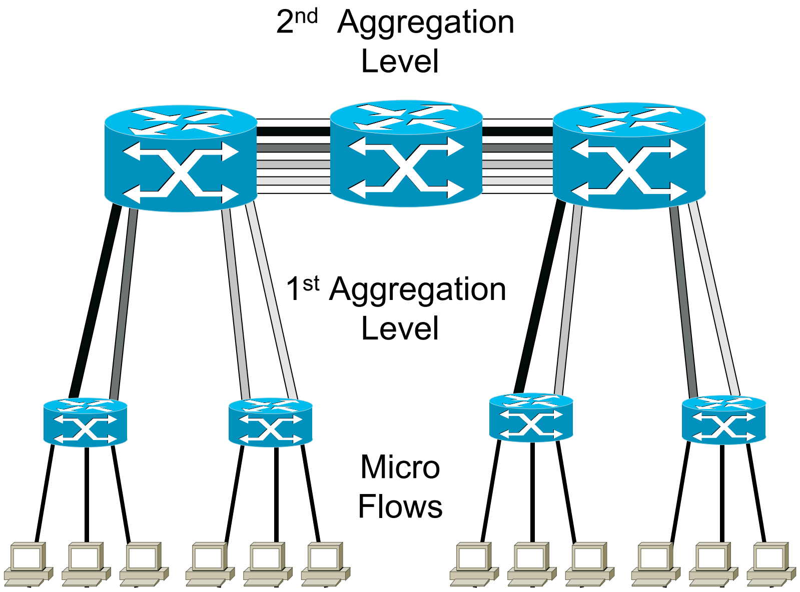

Figure 1: State reduction by hierarchical reservation aggregation.

3.3 Signaling Reduction by Overreservation

Best use is made of the booked capacity for an aggregate if its reservation is tight, i.e. it can not support more than the already aggregated flows. In this case, establishing, updating, and terminating an e2e session entails a change of the aggregate reservation. In a hierarchical structure, these updates are propagated up to the highest level of aggregation, which leads to a session update in each router on the path of the concerned e2e session. Hence, the proposed architecture reduces session states but not the signaling amount. If an aggregate comprises many sessions, their updates can keep all participating routers busy, even if the capacity changes are negligible compared to the overall aggregate reservation. Therefore, an aggregate reservation should outlast at least a few session requests, updates, or teardowns. This can be achieved at the expense of some small capacity overreservation. Here, overreservation is understood in the sense that the AC mechanism can allow some more flows within the aggregate. It does not mean that capacity overbooking is prohibitive.

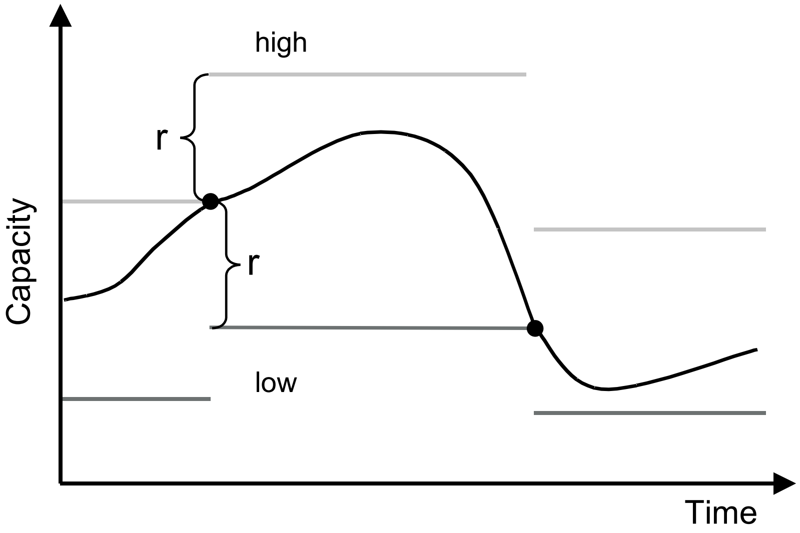

Figure 2: An update scheme with overreservation.

Overreservation decreases network performance, so we propose a simple control mechanism to avoid extensive waste of capacity. It is illustrated in Figure 2. The overall capacity demand for an aggregate reservation is denoted by θ. When an update takes place, θ_{high} capacity is ordered and a lower threshold θ_{low} is defined. The next update for the aggregate is only necessary when $\theta < \theta_{low}$ or $\theta_{high} < \theta$ occurs. Eventually, bandwidth efficiency is traded for signaling costs. This issue will be investigated in the next section.

4 Analysis of the Mean Inter-Update Time

In this section we establish a model for aggregate reservation updates and give simple equations to compute the mean inter-update time analytically. Furthermore, we propose a rule of thumb for overreservation that yields best results.

4.1 Model for Aggregate Reservation Updates

We investigate the previously described update scheme (θ_{low}, θ, θ_{high}) in an IP telephony environment. All voice calls have the same statistical properties and require a toll QoS so that overbooking is not possible. The used capacity $\theta(t)$ is proportional to the number $n(t)$ of admitted e2e sessions. We will use only $n(t)$ in the following and adopt n_{low} and n_{high} for θ_{low} and θ_{high} accordingly. The aggregate reservation is updated when $n(t)$ leaves the tolerance window $[n_{low}; n_{high}]$ for the first time. Since we focus on the update behavior, we assume that bandwidth always suffices so that all requests can be served.

From conventional telephony systems we know that the inter-arrival time between two calls is best described by an exponentially distributed random variable A ($A(t) = 1 - e^{-\lambda \cdot t}$). The mean inter-arrival time is the inverse of the arrival rate ($E[A] = \frac{1}{\lambda}$). The call holding time B is as well exponentially distributed with mean $E[B] = \frac{1}{\mu}$. If there are currently n sessions in place, the overall call termination rate is $n \cdot \mu$. Hence, we can view $n(t)$ as a continuous time Markovian birth-death process.

4.2 Performance Analysis

We compute the mean of that inter-update time in the above model but first we describe the process $n(t)$. The probability for a transition from a state $n(t_*) = i$ to another state $n(t_* + t) = j$ is denoted by $p_{i,j}(t)$. The transition rate from a state i to another state j is defined by

$$q_{i,j}^{w} \quad = \quad \lim_{t \to 0} \frac{p_{i,j}(t_* + t)}{t}. \tag{1}$$

We can easily give the transition rates for the reservation process because A and B are exponentially distributed. A state transition from state i to $i + 1$ happens with the arrival rate $q_{i,i+1} = \lambda$ while a transition to state $i - 1$ occurs with a departure rate $q_{i,i-1} = i \cdot \mu$. As a consequence, the process does not change its state with rate $q_{i,i} = -\lambda + i \cdot \mu$ and all other state transitions are not possible. The state transition rates are accommodated in the state transition matrix

$$(Q)_{i,j} \quad = \quad \begin{cases} \lambda & \text{for } \ j = i + 1, \ \ 0 \le i < \infty \\ i \cdot \mu & \text{for } \ j = i - 1, \ \ 0 < i < \infty \\ -(\lambda + i \cdot \mu) & \text{for } \ j = i, \ \ 0 \le i < \infty \\ 0 & \text{else} \end{cases}. \tag{2}$$

The waiting process for a single update event comprises only the states $\mathcal{W} = \{i | n_{low} \le i \le n_{high}\}$ in the tolerance window for the aggregate reservation. The corresponding rate matrix is given by $Q^w = (Q)_{i,j \in \mathcal{W}}$ and the process stops when $n(t)$ leaves the tolerance window. Its first passage time is the desired inter-update time. In [14], a recursive method, based on taboo sets [15], is proposed to compute the moments of the passage time analytically. This method has also been applied in [16]. In the following we adapt this technique to our problem. The complementary waiting time distribution function $W_i^c(t)$ depends on the initial state $n(t_0) = i$ and can be computed by

$$W_i^c(t) \quad = \quad \sum_{j \in \mathcal{W}} p_{i,j}^{w}(t). \tag{3}$$

The sum of the rates in states n_{low} and n_{high} is negative and the process eventually dies. This entails for the complementary probability density function of the waiting time

$$\lim_{t \to \infty} w_i^c(t) \quad = \quad 0. \tag{4}$$

The analysis is as follows. Starting from the Kolmogorow backward equation, a differential equation for the waiting time distribution is established. Using Laplace transformation (see Appendix for notation), a simple recursion to obtain the n-th moments is derived.

$$\begin{aligned}
\frac{d}{d\,t} p^w_{i,j}(t) &= \sum_{k\in\mathcal{W}} q^w_{i,k}\cdot p^w_{k,j}(t) \;\; \text{sum over } j\in\mathcal{W} \\
\frac{d}{d\,t}\sum_{j\in\mathcal{W}} p^w_{i,j}(t) &= \sum_{k\in\mathcal{W}} q^w_{i,k}\cdot \sum_{j\in\mathcal{W}} p^w_{k,j}(t) \;\; \text{use Equation 3} \\
\frac{d}{d\,t} W^c_i(t) &= \sum_{k\in\mathcal{W}} q^w_{i,k}\cdot W^c_k(t) \;\; \text{use Equation 12 and differentiation} \\
\frac{d}{d\,t} w_i(t) &= \sum_{k\in\mathcal{W}} q^w_{i,k}\cdot w_k(t) \;\; \text{use LT and Equation 14} \\
s\cdot W^*_i(s) + w_i(0) &= \sum_{k\in\mathcal{W}} q^w_{i,k} W^*_k(s) \;\; \text{use differentiation and limit} \\
\lim_{s\to 0}\left(\frac{d^n}{d\,s^n}\big(s\cdot W^*_i(s)\big)\right) &= \lim_{s\to 0}\left(\sum_{k\in\mathcal{W}} q^w_{i,k}\frac{d^n}{d\,s^n} W^*_k(s)\right) \;\; \text{use Equation 15}
\end{aligned}$$

$$\lim_{s\to 0}\left(s\cdot\frac{d^n}{d\,s^n} s\cdot W^*_i(s) + n\cdot\frac{d^{n-1}}{d\,s^{n-1}} W^*_i(s)\right) = \sum_{k\in\mathcal{W}} q^w_{i,k}\lim_{s\to 0}\left(\frac{d^n}{d\,s^n} W^*_k(s)\right)$$

use Equation 16, Equation 12, Equation 4, and Equation 17

$$\begin{aligned}
(-1)^{n-1}\cdot n\cdot E[W^{n-1}_i] &= (-1)^n\cdot\sum_{k\in\mathcal{W}} q^w_{i,k}\cdot E[W^n_j] \\
n\cdot E[W^{n-1}_i] &= -\sum_{k\in\mathcal{W}} q^w_{i,k}\cdot E[W^n_k] \qquad (5)
\end{aligned}$$

The column vector $m^{(n)} = \big(E[W^n_i]\big)_{n_{low}\le i\le n_{high}}$ contains the n-th moments of the inter-update time W_i. Equation 5 can be rewritten as

$$\begin{aligned}
-Q^w\cdot m^{(n)} &= n\cdot m^{(n-1)} \\
m^{(n)} &= -(Q^w)^{-1}\cdot n\cdot m^{(n-1)}. \qquad (6)
\end{aligned}$$

The mean of the inter-update time can be computed by Equation 6 and $m^0 = (1\dots 1)^T$ and the coefficient of variation is calculated by $c_{var}(W_i) = \frac{\sqrt{E[W_i^2]-E[W_i]^2}}{E[W_i]}$.

4.3 A Rule of Thumb for Overreservation

The average aggregate size is $\overline{n} = \frac{\lambda}{\mu}$. We derive a formula that tells the radius of the tolerance window $[\overline{n}-r;\overline{n}+r]$, such that the mean of the inter-update time is approximately constant for different λ and μ. If $\overline{n}$ is large and r is small, we can approximate the call termination rate $i\cdot\lambda$, $i\in[\overline{n}-r;\overline{n}+r]$, in the transition matrix Q^w by $\overline{n}\cdot\mu = \frac{\lambda}{\mu}\cdot\mu = \lambda$ and we get an approximated waiting process given by

$$(Q^w_*)_{i,j} = \begin{cases} \lambda & \text{for } j = i+1,\;\; n_{low} < i \le n_{high} \\ -2\lambda & \text{for } j = i,\;\; n_{low}\le i\le n_{high} \\ \lambda & \text{for } j = i-1,\;\; n_{low}\le i < n_{high} \end{cases}. \qquad (7)$$

For all states $i \in [\overline{n}-r; \overline{n}+r]$, the transition rates towards state $\overline{n}$ in the waiting process Q^w are greater than or equal to the rates in the approximated waiting process Q^w_*. Therefore, the mean inter-update time based on Q^w_* is a lower bound on the real inter-update time. The inverse of the approximated rate matrix Q^w_* is

$$\left((Q^w_*)^{-1}\right)_{i,j} = \begin{cases} -\frac{(i+1)\cdot(2r+1-j)}{\lambda(2r+2)} & \text{for } i \leq j \\ -\frac{(j+1)\cdot(2r+1-i)}{\lambda(2r+2)} & \text{for } i > j \end{cases}. \tag{8}$$

If the waiting process starts in state $n(t_0) = \overline{n} = \frac{\lambda}{\mu}$, the mean inter-update time $\overline{W}$ is computed according to Equation 6 as the negative sum of the elements of the middle row in $(Q^w_*)^{-1}$:

$$E[W_{\overline{n}}] = \frac{1}{(2r+2)\lambda} \cdot \left[(r+1) \cdot \sum_{k=1}^{r+1} k + (r+1) \cdot \sum_{k=2r+1-2r}^{2r+1-(r+1)} k\right] = \frac{1}{2\lambda} \cdot (r+1)^2. \tag{9}$$

If we choose $r = \sqrt{\overline{n}} = \frac{\lambda}{\mu}$, we get

$$\begin{aligned} E[W_{\overline{n}}] &= \frac{1}{2\lambda} \cdot (r+1)^2 = \frac{1}{2\lambda} \cdot \left(\sqrt{\frac{\lambda}{\mu}} + 1\right)^2 = \frac{1}{2\mu} \cdot \left(1 + \sqrt{\frac{\mu}{\lambda}}\right)^2 \\ &= \frac{E[B]}{2} \cdot \left(1 + \sqrt{\frac{1}{\overline{n}}}\right)^2. \end{aligned} \tag{10}$$

This expression converges for large aggregate sizes $\overline{n}$ and we have found a rule of thumb to keep the inter-update time constant for various aggregate sizes $\overline{n}$.

4.4 The Update Process Modeled as a Random Walk

Now, we model the presented update scheme as a random walk. We approximate the mean rate between two changes of $n(t)$ by $\lambda + i \cdot \mu$, $i \in [n_{low}; n_{high}]$, which is about $2 \cdot \overline{n} \cdot \mu$ for the above discussed case. Hence, it takes on average $\overline{n}$ steps until the desired mean inter-update time $\frac{1}{2} \cdot E[B] = \frac{1}{2 \cdot \mu}$ is achieved. We assume that the probabilities p for an increase and $q = 1 - p$ for a decrease of $n(t)$ is equal for every single step.

We compute the first moment and the variance for the deviation X of end position of random walk from its initial position $n(t_0) = \overline{n}$. To that aim, we consider the random variable Y_+ which describes the number of increases in the first $\overline{n}$ transitions. Y_+ is distributed according to a binomial distribution with parameters $\overline{n}$ and p. Hence, we have $E[Y_+] = \overline{n} \cdot p$ and $Var[Y_+] = \overline{n} \cdot p \cdot q$. The number of decreases is exactly $Y_- = \overline{n} - Y_+$ and we compute $X = Y_+ - Y_- = 2 \cdot Y_+ - \overline{n}$. Therefore, we get for $p = \frac{1}{2}$ the first moment $E[X] = 2 \cdot E[Y] - \overline{n} = 0$ and the variance $Var[X] = 4 \cdot Var[Y] = \overline{n}$. Hence, the standard deviation for the random walk X is $\sigma_X = \sqrt{\overline{n}}$.

This means that if the standard deviation of the approximated random walk is taken as the amount of overreservation for the update process based on $n(t)$, the first moment of the inter-update time $E[W_{\overline{n}}]$ remains constant regardless of the mean aggregate size $\overline{n}$. This result is astonishing because the two underlying models are quite different with respect to the observed performance measures. In the above model, we compute the mean of the inter-update time which is limited by a threshold regarding $n(t)$. Here, we compute the standard deviation of a random walk that is limited by an approximated number of steps.

5 Numerical Results

In this section we illustrate the performance behavior of the presented aggregate reservation update model. The state process depends on the arrival rate λ and on the mean call holding time $E[B]$. We prefer studies that are applicable to different parameter sets (λ, μ), so we choose the average aggregate size $\overline{n} = \frac{\lambda}{\mu}$ as the normalized input parameter and measure the time in units of $E[B]$ for the given results.

5.1 Influences on Inter-Update Time

The mean inter-update time $E[W_i]$ depends on the initial state $n(t_0) = i$, in which the process has been updated. Furthermore, it also depends on the position of the tolerance window $[n_{low}, n_{high}]$. In Figure 3 the mean inter-update time is given for an average aggregate size $\overline{n} = 10000$, and a tolerance window of radius 100. The curves for $E[W_i]$ show a clear maximum. If the window is located below $\overline{n}$, the maximum of $E[W_i]$ is in the lower part of the window, if the window is positioned above $\overline{n}$, the maximum of $E[W_i]$ is in the upper part of the window. If we consider a symmetric window around $\overline{n}$, the maximum inter-update time can be expected when the initial state $n(t_0)$ is in the middle of the window.

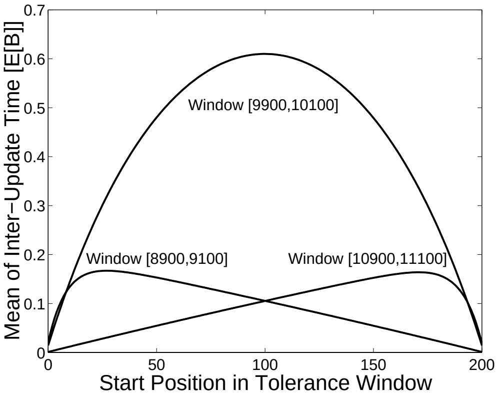

Figure 3: The impact of initial state and tolerance window on the average inter-update time.

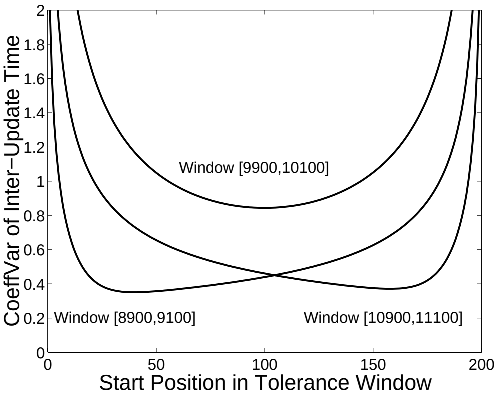

Figure 4: The impact of initial state the tolerance window on the coefficient of variation of the inter-update time.

Figure 4 shows the coefficient of variation of W_i. Note that the variability of W_i is also reduced by optimizing the mean inter-update time. The coefficient of variation is clearly smaller than 1 for $E[W_{\overline{n}}]$, hence, the inter-update time distribution is not Markovian.

The current arrival rate in a communication network is unknown and strongly time dependent. However, it is most probable that the process $n(t)$ resides in state $\overline{n}$, thus, we take the present aggregate size $n(t)$ as an estimate for the present $\overline{n} = \frac{\lambda}{\mu}$. This justifies that we consider in the following only symmetric tolerance windows around the average aggregate size $\overline{n}$.

It is obvious that the inter-update time depends on the radius r of the tolerance window $[\overline{n}-r, \overline{n}+r]$. From Equation 10 it is clear that the inter-update time rises quadraticly with the window radius r as long as the the assumed approximation is good. Figure 5 illustrates that the mean inter-update time increases exponentially as soon as the differences of the call termination rates in Q^w carry weight.

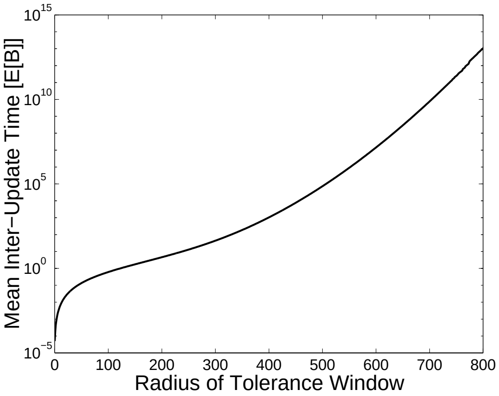

Figure 5: The sensitivity of the mean inter-update time to the window radius.

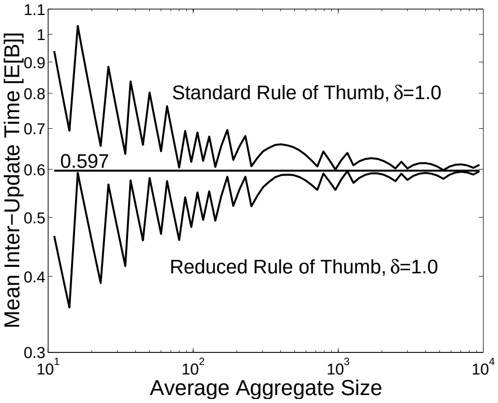

Figure 6: The accuracy of the rule of thumb.

5.2 An Enhanced Rule of Thumb for Adjusting the Radius of the Tolerance Window

The integral size of the tolerance window as well as the quadratic and exponential growth of the mean inter-update time make the dimensioning of the update scheme difficult because there is no analytical formula that computes the required window size. The rule of thumb gives a means to adjust the radius of the tolerance window such that the mean inter-update time is constant. We scale that rule by a linear factor δ such that the radius of the window is computed by

$$r = \lfloor \delta \cdot \sqrt{\overline{n}} \rfloor. \tag{11}$$

Figure 6 depicts the mean inter-update time of an aggregate reservation process depending on the size of the aggregate. $E[W_{\overline{n}}]$ is larger than the asymptotic mean inter-update time and the deviations are considerable for small aggregates. However, if we try a slightly smaller reservation with $r = \lfloor \delta \cdot \sqrt{\overline{n}} \rfloor - 1$ for the radius of the tolerance window, the mean inter-update time will be smaller than the asymptotic value and we get the same strong deviations to the opposite side. This suggests that the proposed rule of thumb is very accurate. The discontinuous shape of the curves just shows that $E[W_{\overline{n}}]$ is hard to control. However, this effect is diminished with larger reservation aggregates. The derivation of the rule of thumb was based on the approximated rate matrix Q^w_*. Therefore, the mean inter-update time converges for $\delta = 1$ with increasing average aggregate size $\overline{n}$ to $0.597 \cdot E[B]$ instead to $0.5 \cdot E[B]$.

Figure 7 illustrates that δ enables the rule of thumb to produce window sizes for different mean inter-update times. This makes the rule of thumb relevant for practical use. A suited δ may be configured in aggregating routers depending on the desired mean inter-update time. Then, the appropriate radius for the tolerance window, that determines the degree of overreservation for an aggregate reservation, can be computed even in real-time.

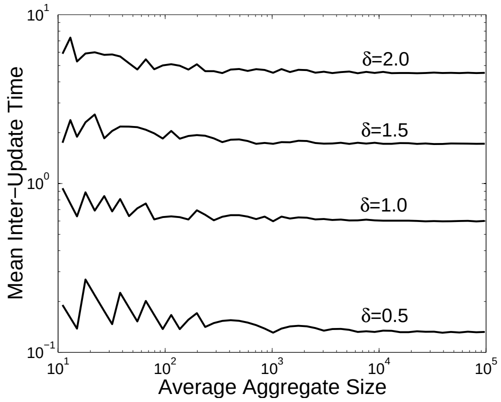

Figure 7: The mean inter-update time can be adjusted by δ.

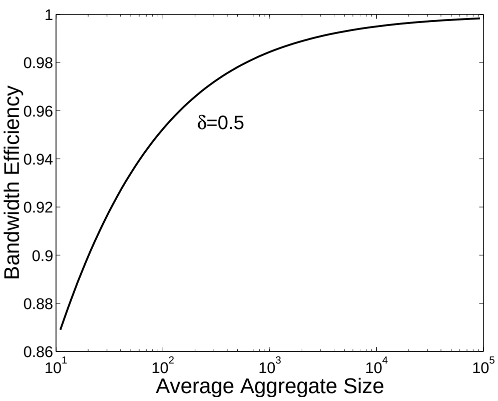

Figure 8: The efficiency of the update scheme with overreservation.

5.3 Overreservation and Efficiency

The proposed update scheme uses overreservation to limit the number of updates for aggregate reservations. This impacts the efficient use of network resources. The rule of thumb is able to adjust the tolerance window in the presented update model such that the mean inter-update time remains constant. The window radius r scales with the square root of the average aggregate size. Therefore, the degree of overreservation converges towards zero for large aggregates: $\lim_{\overline{n}\to\infty} \frac{\lfloor \delta\cdot\sqrt{\overline{n}} \rfloor}{\overline{n}} = 0$. Figure 8 illustrates the efficiency of the update scheme.

Thus, we have proven that the scalable e2e signaling and reservation architecture wastes only little capacity for large aggregates in spite of overreservation and networks running that architecture can be operated efficiently.

6 Conclusion

We presented a concept of a protocol architecture for signaling and resource reservation that overcomes the scaling problem of IntServ. In contrast to DiffServ, it is able to provide hard e2e QoS guarantees. The key idea is reservation state reduction in the router MIBs by hierarchical reservation aggregation. In addition, overreservation for aggregates diminishes the update frequency for their reservations which reduces the signaling amount. This concept is realized in various protocol implementations.

For this scenario we suggested a simple mechanism to control the time between two updates of an aggregate reservation and to limit the waste of capacity due to overreservation. We modeled the process of aggregate reservation updates and investigated the influence of the update scheme on the inter-update times analytically. We found a rule of thumb to determine the appropriate amount of overreservation that keeps the mean inter-update time constant. With this engineering rule, it turns out that the required overreservation for large reservation aggregates is only a small fraction of the allocated capacity. This proves that networks running the proposed architecture can be operated efficiently in spite of overreservation.

Acknowledgements

The author would like to acknowledge the fruitful discussions with Prof. P. Tran-Gia, the valuable comments on random walks by Prof. G. Zimmermann (University of Stuttgart-Hohenheim), and the insightful simulations performed by F. Eckhardt.

Appendix

We give some basic equations from probability theory that we have used in the calculations of Section 4.2.

$$\begin{aligned}
a^c(t) = \frac{d}{d\,t}A^c(t) &= \frac{d}{d\,t}(1 - A(t)) = -a(t) && (12)\\
LT\Big\{a(t)\Big\} = A^*(s) &= \int_0^\infty e^{-st}a(t)\;d\,t && (13)\\
LT\Big\{\frac{d}{d\,t}a(t)\Big\} &= s \cdot A^*(s) - a(0) && (14)\\
\frac{d^n}{d\,s^n}\big(s \cdot A^*(s)\big) &= \lim_{s\to 0} s \cdot \frac{d^n}{d\,s^n}\big(s \cdot A^*(s)\big) + n \cdot \frac{d^{n-1}}{d\,s^{n-1}}A^*(s) && (15)\\
\lim_{s\to 0} s \cdot A^*(s) &= \lim_{t\to\infty} a(t) && (16)\\
E[A^n] &= \lim_{s\to 0}(-1)^k \cdot \frac{d^n}{d\,s^n}A^*(s) && (17)
\end{aligned}$$

References

[1] T. Li and Y. Rekhter, "RFC2430: A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)." ftp://ftp.isi.edu/in-notes/rfc2430.txt, Oct. 1998.

[2] F. Baker *et al.*, "Aggregation of RSVP for IPv4 and IPv6 Reservations." http://www.ietf.org/internet-drafts/draft-ietf-issll-rsvp-aggr-03.txt, April 2001.

[3] P. Pan and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations," *Journal of Communications and Networks*, vol. 2, pp. 157–167, June 2000.

[4] O. Schelén and S. Pink, "Aggregating Resource Reservations over Multiple Routing Domains," in *IFIP* 6^{th} *International Workshop on Quality of Service (IWQoS'98)*, May 1998.

[5] A. Terzis, J. Wang, J. Ogawa, and L. Zhang, “A Two-Tier Resource Management Model for the Internet,” in *Global Internet Symposium’99*, Dec. 1999.

[6] J. Schmitt, M. Karsten, L. Wolf, and R. Steinmetz, “Aggregation of Guaranteed Service Flows,” in *IFIP 7^{th} International Workshop on Quality of Service (IWQoS’99)*, June 1999.

[7] H. Fu and E. W. Knightly, “Aggregation and Scalable QoS: A Performance Study,” in *IFIP 9^{th} International Workshop on Quality of Service (IWQoS’01)*, June 2001.

[8] J. Wroclawski, “RFC2210: The Use of RSVP with IETF Integrated Services.” ftp://ftp.isi.edu/in-notes/rfc2210.txt, Sep. 1997.

[9] B. Braden *et al.*, “RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.” ftp://ftp.isi.edu/in-notes/rfc2205.txt, Sep. 1997.

[10] S. Blake *et al.*, “RFC2475: An Architecture for Differentiated Services.” ftp://ftp.isi.edu/in-notes/rfc2475.txt, Dec. 1998.

[11] E. C. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture.” http://www.ietf.org/rfc/rfc3031.txt, Jan. 2001.

[12] G. Swallow *et al.*, “RSVP-TE: Extension to RSVP for LSP Tunnels.” http://www.ietf.org/internet-drafts/draft-ietf-mpls-rsvp-lsp-tunnel-08.txt, Feb. 2001.

[13] B. Jamoussi *et al.*, “Constraint-Based LSP Setup Using LDP.” http://www.ietf.org/internet-drafts/draft-ietf-mpls-cr-ldp-05.txt, Feb. 2001.

[14] P. Kühn, *Über die Berechung der Wartezeiten in Vermittlungs- und Rechnersystemen*. PhD thesis, University of Stuttgart, 1972.

[15] R. Syski, *Introduction to Congestion Theory in Telephone Systems*. Elsevier, 1986.

[16] M. Mittler, T. Ono-Tesfaye, and A. K. Schömig, “On the Approximation of Higher Moments in Open and Closed Fork/Join Primitives with Limited Buffers,” Technical Report, No. 126, University of Würzburg, Institute of Computer Science, Sep. 1995.