Departmental Papers (ESE)　　　　　　　　Department of Electrical & Systems Engineering

6-27-2005

Enabling Scalable Inter-AS Signaling: a Load Reduction Approach

Rute Sofia
Siemens AG Corporation Technology

Roch A. Guérin
University of Pennsylvania, roch@guerinfamily.org

Pedro Veiga
University of Lisbon

Enabling Scalable Inter-AS Signaling: a Load Reduction Approach *

Rute Sofia
Siemens AG
Corporate Technology
Munich, Germany
rute.sofia@siemens.com

Roch Guérin
ESE, University of Pennsylvania
Philadelphia, U.S.A
guerin@ee.upenn.edu

Pedro Veiga
DI, University of Lisbon
Lisbon, Portugal
pedro.veiga@di.fc.ul.pt

Abstract

In order to achieve better scalability, inter-domain signaling protocols rely on aggregation to reduce the amount of state information that routers are required to maintain. Nonetheless, they do not address another scalability key factor, the signaling load associated with establishing and maintaining reservations. Such load can be reduced if bandwidth is over-reserved. Over-reservation allows accommodating reservations without exchanging signaling messages, but may result in additional blocking. In this paper, we carry out a systematic investigation of the impact of over-reservation in different aggregation approaches, evaluating such impact in terms of the achieved signaling reduction, and blocking.

1. Introduction

In order to deal with loss or delay sensitive services, and in addition to network over-provisioning, a provider may have to support end-to-end and fine-grained *Quality of Service (QoS)* guarantees, both intra and inter-domain. Inter-domain links usually support the largest shares of traffic crossing a provider's domain, which makes fine-grained QoS support a complex process, often resulting in a lack of scalability. To cope with this problem, current proposals for inter-domain QoS signaling protocols perform *aggregation* to significantly reduce the amount of state information maintained by *Boundary Routers (BRs),* i.e., routers that interconnect *Autonomous Systems (ASes)*. Essentially, instead of handling individual reservations, BRs aggregate them according to some criterion, e.g., their destination AS, and then only keep track of *state* at the aggregate level. Despite the fact that state is kept at the aggregate level only, BRs still need to process the signaling messages generated during the individual reservation setup, update, and tear-down phases,

and this obviously severely impacts the scalability of any solution. However, if aggregates are *over-provisioned*, they can accommodate individual requests without requiring the exchange of new signaling messages. The flip-side of over-reservation is the accurate computation of the optimal over-reservation level.

Providing insight into the over-reservation theme is therefore the main goal of this paper, which proposes and evaluates several over-reservation methods, in terms of the achieved *signaling load reduction (SR)* and the associated *blocking probability (BP)*. The proposed methods are explored over a broad range of configurations and parameters, e.g., reservation characteristics, or the type of signaling protocol used. In particular, the methods are evaluated in the context of two proposed inter-domain QoS signaling protocols, namely, the *Border Gateway Reservation Protocol (BGRP)* [5] and the *Shared-segment Inter-domain Control Aggregation Protocol (SICAP)* [9]. These protocols were chosen not simply because they are currently the only inter-domain QoS signaling solutions [3] that perform aggregation, but more importantly, because they are significant examples of different aggregation approaches.

We start by reviewing, in section 2, related work. Section 3 expands on the over-reservation methods proposed, while section 4 reports results of this investigation. Finally, Section 5 summarises the findings of the paper and identifies a few extension directions.

2. Related Work

The works here described share with ours the fundamental aspect of improving the scalability of reservation-based approaches by trading off a minor loss in efficiency. For instance, Schélen et al. [4] introduce an advance resource reservation architecture for real-time events whose agents are fully (statically) aware of the underlying network topology and resources. They consider the use of *sink-tree aggregation*, i.e., aggregation of reservations that share the same destination AS, to reduce global state information, but do

not address the SR.

Chuah et al. [1] introduce a hierarchical overlay infrastructure for scalable network provisioning called *Clearing House*. This static infrastructure applies over-reservation between logical groups of ASes (domains), where the bandwidth to be reserved is computed by means of a Gaussian predictor over a given time interval. Even though the scheme achieves reasonably good bandwidth efficiency if the measurement interval is well adapted to high load arrival patterns, its performance is highly influenced by the choice of such interval, and the SR is again not a parameter taken into consideration in their investigation.

Pan et al. [5] propose BGRP, that relies on the concept of sink-tree aggregation. Given that BGRP attains the same signaling load as RSVP [7], Pan et al. suggest as a future enhancement a static SR approach, *quantization*. Quantization is a form of over-reservation triggered by the arrival of a reservation request that could be merged onto an existing, but under-provisioned aggregate. In such situation, the aggregate, which is assumed to be provisioned with a bandwidth share of $Q \geq 2$, automatically experiences a bandwidth increase multiple of Q. Albeit the goal of SR is present, the paper does not investigate the consequent impact in the BP, nor does it propose functions that dynamically adjust to the aggregate demand, as we do in our investigation.

Nikolouzou et al. [2, 6] specifically aim at reducing the signaling load of BGRP through a *delayed resource release* mechanism. Delaying the release of bandwidth that is no longer needed is another form of over-reservation, where the over-reserved bandwidth share is built upon the recycling of bandwidth allocated to terminated reservations. This investigation is probably the one that is the most relevant to our work, given that it considers the achieved SR. However, it is simply focused on sink-tree aggregation (BGRP), and it does not consider long-term fluctuations of the demand of an aggregate, as do the methods and functions we provide in the next section.

3. Over-Reservation Approaches

This section presents the different over-reservation methods and the functions used to compute how much to over-reserve on behalf of an aggregate. For the sake of conciseness, the examples presented are restricted to SICAP. However, given that the evaluation presented in section 4 considers both the shared-segment and the sink-tree aggregation approaches, we briefly explain the operational procedure for SICAP and BGRP here. Full details about the over-reservation extension and how to apply it for BGRP and SICAP can be found in [10].

SICAP and BGRP are *sender-initiated* protocols in the sense that the first BR on the path triggers the establishment of reservations. BGRP merges requests that have the same destination AS, creating aggregates shaped as sink-trees where the roots are the destination ASes. SICAP merges requests that share segments of a path, according to the criteria specified in [8], and which allows SICAP to perform better than BGRP in terms of state information required [9]. In terms of messaging, both BGRP and SICAP use a similar pair of messages to establish a reservation R_i. First, a *probing* message is originated at the first BR on the path of R_i and is forwarded until the last BR, simply to gather information about resources and routes, thus it requires no state in BRs. Then, as reply, an *allocation* message is originated at the last BR and is used to commit resources on the previously probed path. Additionally, both protocols release resources by explicitly forwarding (between the first and the last BR) termination messages.

For the described messaging scheme there are two situations that can be considered as "natural" over-reservation triggers: the arrival of new reservation requests and the termination of existing reservations. In the former, over-reservation is accomplished by asking for more resources than the amount initially requested, while in the latter it is accomplished by delaying the release of resources allocated to terminated reservations. These are the triggers we consider and describe next, after summarising the used notation in Table 1.

Table 1. Over-reservation notation.

Notation	Meaning
$b_A(x)$	Bandwidth in use by aggregate x
$b_R(x)$	Total bandwidth reserved for aggregate x
$b_S(x)$	Surplus of bandwidth to request on behalf of aggregate x
C	Outgoing link capacity
$B_R = \sum b_R(x)$	Outgoing link total reserved bandwidth
b_i	Bandwidth share of request i

3.1. Reservation Request Based Trigger

The algorithm we use to determine if and by how much to over-reserve is triggered when a BR receives a new request (R_i) asking for b_i bandwidth units. If the link holds at least enough bandwidth to accommodate R_i the algorithm proceeds to check if there is an aggregate onto which the new request can be mapped. If none exists, or if there is a candidate aggregate not adequately provisioned, and if there is at least enough bandwidth to later provision the new request, i.e., $C - B_R \geq b_i$, the request must be forwarded. Note that these first two steps are "conservative" in the sense that a decision is made immediately upon receiving the probing message rather than allowing it to proceed and wait until the eventual return of the corresponding allocation message. We chose to perform this decision earlier to avoid propagating messages concerning reservations with a

minimal likelihood of success.

The most interesting case for our algorithm occurs when the aggregate does not have enough bandwidth to automatically accommodate R_i. The question is then by how much and in particular, whether it is beneficial or not to ask for more than the original b_i share. For such cases the algorithm proceeds to compute the bandwidth surplus the aggregate might over-reserve, $b_S(x)$, computation which can be performed in innumerable ways. One way is the mentioned quantization technique, which simply computes $b_S(x)$ as a multiple of a quantity $Q \geq 2$. While simple, such an approach can easily lead to undesirable results, e.g., resource starvation, since a BR may end up requesting more bandwidth than an aggregate might ever use. A less dramatic alternative is to consider the new request (b_i), the amount of remaining bandwidth on the link ($C - B_R$), as well as the share of the link bandwidth that an aggregate x is actually using ($b_A(x)$), and then compute a new reservation based on those quantities, as expressed in (1).

$$b_S(x) = \begin{cases} 0, & C = 0 \\ \frac{b_A(x)}{C} * (C - B_R - b_i), & otherwise \end{cases} \quad (1)$$

Equation (1) targets some level of fairness in how the link bandwidth is shared between aggregates, given that aggregates are entitled to a share of reserved resources that is proportional to their actual usage. One possible enhancement to (1) is to replace the current bandwidth usage $b_A(x)$ by a bandwidth usage estimate $\tilde{b}_A(x)$ of aggregate x, estimate which we present in section 4. Using such an estimate can smooth out over-reservation decisions performed at an instant when the aggregate usage is very different from its typical value and also, provide greater robustness against demand fluctuation.

3.2. Reservation Release Based Trigger

The counterpart of the over-reservation trigger presented in section 3.1 is a method based on the release of resources due to the explicit termination of individual reservations. Reservation termination is performed through $TEAR$ messages which are propagated from the reservation source to its destination, but are only processed at the BRs along the path. This represents a double opportunity for over-reservation. First, if resources are not released, they become available to automatically accommodate requests. Second, this decreases the global number of $TEAR$ messages: in most cases, it suffices to simply forward the $TEAR$ message between the aggregate starting and ending points.

When a BR receives a request to release b_i units on behalf of R_i, it can decide whether or not to release that bandwidth on behalf of the aggregate x that carries R_i, decision which depends both on the overall load of the link and on the bandwidth usage of x. Specifically, if B_R is lower than the link capacity the decision of whether or not to release

bandwidth is made based on a function, $r(x)$, given in (2). $r(x)$ triggers the release of bandwidth only if the reserved but unused bandwidth of aggregate x ($b_R(x) - b_A(x)$) is a significant share of the free link bandwidth ($C - B_R$), when compared to the link capacity share occupied by x ($\frac{b_A(x)}{C}$). If no bandwidth is released, the $TEAR$ message is stopped, while if a decision is made to release some bandwidth, then not only is the reserved bandwidth of the aggregate updated, but the $TEAR$ message is forwarded to the next BR.

$$r(x) = \begin{cases} 1, & \frac{b_R(x)-b_A(x)}{C-B_R} > \frac{b_A(x)}{C} \\ 0, & otherwise \end{cases} , C \neq 0, B_R < C \quad (2)$$

The rationale behind $r(x)$ is as follows: aggregate x is allowed to retain its over-reserved bandwidth share only when the ratio $\frac{b_R(x)-b_A(x)}{C-B_R}$ is lower than or equal to the ratio $\frac{b_A(x)}{C}$. The first ratio compares the bandwidth margin of the aggregate to the link bandwidth margin, while the second corresponds to the link share occupied by x. For example, an aggregate that is consuming 10% of the link bandwidth is entitled to over-reserve an amount of bandwidth that is also up to 10% of the unused link bandwidth , while an aggregate that consumes 50% of the link bandwidth could over-reserve up to 50%. $r(x)$ is therefore a decision function that is sensitive to the current link load and that allows over-reservation decisions to be taken based on the bandwidth demand of active aggregates.

Once $r(x)$ has determined that the aggregate bandwidth usage calls for releasing some bandwidth, it is still necessary to determine how much to release. One simple option would be to just release the initial b_i value, but this is not necessarily the best option. For example, the link load may have increased significantly since the last $TEAR$ arrived, which may then warrant releasing more bandwidth than the original b_i. Aiming at understanding the impact of different release amounts, we formulated and experimented with several functions [10], settling on the best performing one that is captured in (3).

$$b_f(x) = \frac{b_R(x) - b_A(x)}{B_R} * (b_R(x) - b_A(x)) \quad (3)$$

4. Performance Evaluation

In this section, we present a comprehensive evaluation carried out by means of simulations that rely on a modified version of the *network simulator version 2 (ns2)* [13], extended to implement both SICAP and BGRP. All the simulations rely on a 50 AS-level topology where all inter-AS links are considered to have the same generic capacity. The source and destination of new reservation requests are chosen according to a real mapping of IP addresses to ASes [12]. The arrival of new requests is modelled as a Poisson process, and in order to investigate the impact of the duration of requests two representative session holding times are considered, namely, 20 seconds and 120 seconds, standing for examples of *short-lived* and *long-lived*

requests, respectively. In order to assess the sensitivity of the results to the amount of bandwidth that different requests ask, three types of bandwidth distributions are considered: small bandwidth requests with bandwidth requirements uniformly distributed between 0 and 0.1% of the link capacity; high bandwidth requests assuming bandwidth requirements uniformly distributed between 0.1% and 1% of the link capacity; a mixed environment consisting of an equal percentage of small and large requests, e.g., audio and video sessions. Moreover, to provide a consistent comparison of the performance of both protocols, several *intensities* of requests, i.e., the average number of requests in the system per second, are also considered. The computed statistics rely on the tracking of every change, collected per outgoing link, and are computed for a 95% confidence interval. Values are only considered after an adequate warm-up period, and the simulations are repeated several times with different random seeds, to ensure statistically independent rounds.

We present three main evaluation sets which stand for a representative subset of the full results available in [10], not included in the paper due to space constraints. The first set reports on the performance of the reservation request based trigger, the second set targets the evaluation of the delayed release based trigger, while the third set analyses the combination of both triggers. Before proceeding with the evaluation, we briefly introduce the demand prediction function on which we rely.

4.1. A Predictor for the Aggregate Demand

The estimation of the bandwidth demand of an aggregate should ideally satisfy a number of basic properties: 1) react fast to bandwidth changes, 2) converge fast to the actual demand, 3) smooth out short-term fluctuations, and 4) have small computation complexity. (4) is a variation of the *Exponential Moving Average Estimation (EMA)* [11] which not only fulfils requirements 2), 3), and 4), but which is also able to rapidly adapt to abrupt variations.

$$e_{b_A}(n+1) = \frac{b'(n+1)}{\Delta(n+1)} \quad (4)$$

In (4), $e_{b_A}(n+1)$ represents the $(n+1)th$ estimation of the aggregate demand at the $(n+1)th$ sampling instant $t(n+1)$, and is equal to the ratio of two different weighted exponential averages, computed as represented in (5): $b'(n+1)$ represents the estimated bandwidth, while $\Delta(n+1)$ represents an estimate of the duration of the time intervals for the different bandwidth samples. The introduction of $\Delta(n+1)$ provides the estimate with the means to capture not only demand fluctuations, but also to weight such fluctuations according to their duration.

$$\begin{aligned} b'(n+1) &= (1-w)*b'(n) + w*[t(n+1)-t(n)] \\ \Delta(n+1) &= (1-w)*\Delta(n) + w*[t(n+1)-t(n)] \end{aligned} \quad, w \in [0,1] \quad (5)$$

Results presented in [10] offer evidence that this function is beneficial to over-reservation, especially when demand fluctuates significantly. The function is of low complexity and does not introduce any penalty even in the presence of stable traffic patterns, e.g., long-lived requests. Therefore, the simulations presented throughout the next sections rely on this predictor.

4.2. Reservation Request Based Trigger

(a) long-lived requests, BP vs. intensity

(b) long-lived requests, SR vs. intensity

(c) short-lived requests, BP vs. intensity

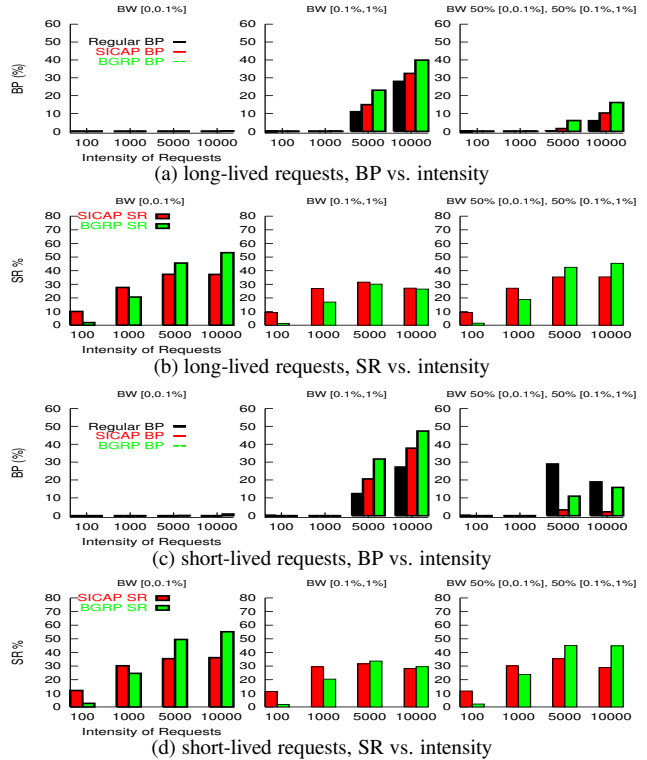(d) short-lived requests, SR vs. intensity

Figure 1. Reservation request based trigger.

To analyse the performance trade-off of the reservation request based trigger, we use a scenario involving long-lived requests. Fig. 1 (a) compares the BP values of both BGRP and SICAP to the no over-reservation case, *Regular BP*, (which obviously is the same for both BGRP and SICAP), while Fig. 1 (b) displays the reduction in SR that BGRP and SICAP are able to achieve. In all figures, the x-axis represents different intensities of requests, while the three sub-figures correspond to the three different bandwidth distributions, as indicated by the notation $BW [x, y]$.

The first main observation is that this over-reservation method translates into noticeable increases in BP only for relatively large (5,000 or 10,000) intensities of requests and primarily for large and mixed bandwidth distributions. Overall, over-reservation consistently results in lower BP values when applied to SICAP than when applied to BGRP but this comes at the cost of a slightly lower reduction in

SR for some of the scenarios where the intensity of requests is higher. This behaviour is partially a consequence of the aggregation approaches followed by BGRP and SICAP. SICAP aggregates based on shared-segments of a path, providing better bandwidth "sharing". As for the SR, it is a function of the likelihood of finding an adequate and at the same time well-provisioned aggregate. Higher intensities of requests give rise to more nodes holding over-reserved resources, meaning that after a while most BRs on a path hold enough resources to merge reservations without the need to propagate signaling messages. While the sink-tree aggregation method followed by BGRP automatically profits from this situation, SICAP still requires all aggregate segments on a path to be checked for available resources, which means sending messages at least to the starting point of each aggregate segment on the reservation path.

While these results illustrate how over-reservation can reduce the signaling load for different aggregation approaches there is still the need to understand the impact of other parameters such as, for instance, the duration of requests. Given that short-lived requests provide greater variability in how requests come and go, we analyse next their impact on the methods discussed so far, presenting results in Figs. 1 (c) and (d) for the BP and the SR, respectively.

The BP values now obtained are noticeably not uniform, and there are even two cases (cf. Fig. 1 (c), right-hand chart, intensities of 5,000 and 10,000 requests) where over-reservation actually decreases the regular (obtained without over-reservation) BP, being this phenomenon enhanced when over-reservation is applied to SICAP. This occurs for high intensity scenarios involving a mixture of requests with very different bandwidth requirements, and the underlying reasons are as follows. First, over-reservation ends up allowing more "small" requests to be accommodated, independently of the aggregation approach followed. Second, the shared-segment approach on which SICAP relies introduces more flexibility in the way bandwidth is shared, and this flexibility enhances the observed phenomenon.

In terms of state variability, the over-reservation relevance is amplified for high variability scenarios (short-lived requests) as corroborated by the significant decrease in BP visible for higher intensities of requests. As for the SR, the percentage attained in high variability scenarios is similar to the one attained in low variability scenarios. However, these are relative values, not absolute. Which means that the signaling reduction is much more relevant for high variability scenarios, given that they attain higher signaling loads.

4.3. Reservation Release Based Trigger

The previous simulations showed that the duration of requests does not have a major qualitative impact on the behaviour of different over-reservation methods, and there-
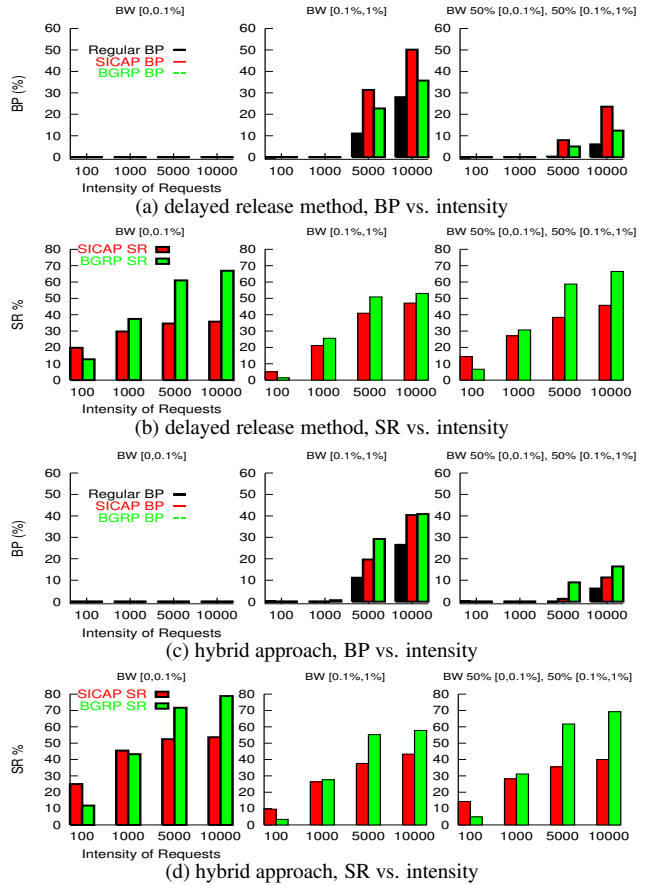

Figure 2. Reservation release and hybrid based triggers .

fore, the next simulations simply rely on long-lived requests, and on the delayed release method described in section 3.2. Results for such scenario are provided in Figs. 2 (a) and (b).

A striking difference from the previous trigger is that the one we now explore results in a much higher overall BP and a lesser SR for SICAP across all scenarios. In contrast, this method is extremely advantageous for BGRP, given that it is the method that resulted in a lesser BP and a higher SR for all the cases depicted. We believe that this behaviour is a direct consequence of the fact that BGRP only creates one aggregate per path. This trigger allows the earlier availability of over-reserved resources along an entire segment of a sink-tree, i.e., from the branch where the source BR is attached up to the sink, and therefore, TEAR messages stop in a point closer to the source BR, thus decreasing the overall signaling. SICAP however, creates more than one aggregate per path. Resources released at the termination of reservations become available on the aggregate that is closer to the BR source, i.e., the first aggregate on the reservation path, but it is also necessary to propagate the release request to other (possible) aggregates on the path, which

not only takes both additional time and messages, but also means that subsequent requests might only find available resources in some of the aggregates they traverse, which contributes to a higher BP.

4.4. A Hybrid Approach

The final set of results are presented in Figs. 2 (c) and (d) and are a consequence of combining the previous over-reservation triggers into a hybrid approach. They reveal that, in general and independently of the signaling protocol used, the SR achieved is now better than when either simply using one of the two other methods. This method is particularly beneficial for SICAP, which attains a SR often in the order of 50%, while the BP achieved by this method is the lowest, only occurring in highly congested scenarios, i.e., in scenarios where high blocking values is present when either SICAP or BGRP operate without over-reservation. The picture is, however, slightly different for BGRP that exhibits consistently higher BP values across all scenarios. While this is the best method to apply to SICAP, the best over-reservation method to apply to BGRP is the delayed release one, as it translates into lower BP, and quite high (reaching 60%) SR values.

5. Conclusions and Future Work

This paper presented an investigation covering the evaluation of several over-reservation methods that aim at lowering the signaling load of inter-domain QoS signaling proposals such as SICAP or BGRP, being the findings as follows. Firstly, the blocking increase penalty associated with over-reserving is tolerable especially in the light of the substantial signaling load reduction achieved. Furthermore, in terms of scalability, the signaling reduction is the most significant for high intensity scenarios, given that these type of scenarios attain higher signaling loads. Secondly, the use of an aggregate demand predictor as an input to the over-reservation process proved useful to better assess how much more to reserve, especially in high variability scenarios. Thirdly, the choice of a specific over-reservation mechanism is relatively insensitive to the duration of requests. Fourthly, because of the different aggregation approaches they follow, different protocols benefit most from different over-reservation methods. Specifically, while the hybrid over-reservation method provides the best performance trade-off for SICAP, the delayed resource release is the best option to apply to BGRP.

This evaluation, while reasonably extensive, is a first step towards the full understanding of the over-reservation theme. Additional steps include not only exploring additional and more generic traffic scenarios, but also and most importantly, to carry out a detailed implementation of an operational protocol with a built-in over-reservation capability. These are topics we are pursuing.

References

[1] Chen-Nee Chuah. *A Scalable Framework for IP-Network Resource Provisioning Through Aggregation and Hierarchical Control*. PhD thesis, University of California at Berkeley, 2001.

[2] Eugenia Nikolouzou, Peter Sampatakos, Lila Dimopoulou, Iakovos S. Venieris, Martin Winter, Bert F. Koch, Thomas Engel, Stefano Salsano, Vincenzo Genova Fabio Ricciato, and Gerald Eichler. BGRPP: Performance evaluation of the proposed Quiet Grafting mechanisms. *Internet Engineering Task Force, draft*, July 2002.

[3] J. Manner (ed.), X. Fu, and P. Pan. Analysis of Existing Quality of Service Signaling Protocols. *Internet Draft (Work in Progress)*, May 2004.

[4] O. Schelen and S. Pink. Aggregating Resource Reservations over Multiple Routing Domains. *IFIP Sixth International Workshop on Quality of Service (IWQoS'98), California*, May 1998.

[5] P. Pan, E. Hahne, and H. Schulzrinne. The Border Gateway Reservation Protocol (BGRP) for Tree-Based Aggregation of Inter-Domain Reservations. *Journal of Communications and Networks*, June 2000.

[6] P. Sampatakos, L. Dimopoulou, E. Nikolouzou, I. S. Venieris, T. Engel, and M. Winter. BGRP: Quiet Grafting Mechanisms for Providing a Scalable End-to-End QoS solution. *Elsevier Computer Communications, Volume 27 (5), pages 423-433*, Mar. 2004.

[7] R. Braden, L. Zhang, and S. Jamin. Resource Reservation Protocol (RSVP) - version 1, Functional Specification. *Request for Comments 2205, Internet Engineering Task Force*, Sept. 1997.

[8] R. Sofia, R. Guérin, and P. Veiga. An Investigation of Inter-Domain Control Aggregation Procedures. *International Conference on Networking Protocols, ICNP'02, Paris, France*, Nov. 2002.

[9] R. Sofia, R. Guérin, and P. Veiga. SICAP, a Shared-segment Inter-domain Control Aggregation Protocol. *High Performance Switching and Routing, HPSR 2003, Turin, Italy*, June 2003.

[10] R. Sofia, R. Guérin, and P. Veiga. A Study of Over-reservation for Inter-Domain Control Aggregation Protocols. Technical report, University of Pennsylvania, May 2003. Available at http://einstein.seas.upenn.edu/mnlab/publications.html.

[11] S. Floyd. Comments on Measurement-based Admissions Control for Controled-Load Service. *Computer Communication Review*, 1996.

[12] S. Uhling and O. Bonaventure. Implications of Interdomain Traffic Characteristics on Traffic Engineering. Technical report, University of Namur, June 2001.

[13] VINT Project. *The ns Manual*. UC Berkeley, LBL, USC/ISI, Xerox Parc, Sept. 2001.