The Border Gateway Reservation Protocol (BGRP) for Tree-Based Aggregation of Inter-Domain Reservations

Ping Pan and Ellen Hahne (Bell Labs) Henning Schulzrinne (Columbia U.)

Abstract

We describe Border Gateway Reservation Protocol (BGRP), work-in-progress, for *inter*-domain resource reservation that can scale in terms of state storage, message processing load and bandwidth. Each stub or transit domain may use its own intra-domain resource reservation protocol. BGRP builds a sink tree for each of the stub domains. Each sink tree aggregates bandwidth reservations from all data sources in the network. Since backbone routers only maintain the sink tree information, the total number of reservation states at each router scales, in the worst case, linearly with the number of domains in the Internet.

BGRP relies on differentiated services for data forwarding. As a result, the number of packet classifier entries is equal to the number of traffic classes, not the number of flows. To reduce the protocol message traffic, routers may reserve domain bandwidth beyond the current load so that sources can join or leave the tree or change their reservation without having to send messages all the way to the root for every such change. We use "soft state" to maintain reservations.

1 Introduction

The Internet is growing at a rate faster than Moore's law as measured in the number of end systems, with the number of networks (or announced routes) and the number of the carrier domains growing much more slowly[1] Recently, several papers have addressed scalable resource reservation. They can be categorized as router-based and server-based. Awduche *et al.*[1] and Guerin *et al.*[2] have proposed a router-based approach by modifying RSVP [3, 4] to support scalable reservation. Both proposals allow routers to aggregate individual reservation requests. However, they do not address the state storage scaling issues: aggregated reservations come in RSVP Sender Template and Session pairs. Instead of carrying application level information in the form of 5-tuples, a Sender Template and/or a Session object can carry a CIDR block or an Autonomous System (AS) number.

These proposals set up reservations between every pair of domains that communicate via reserved flows. This means that state storage scales with the square of the number of communicating domains. Given that there are currently more than 4,500 domains, the state storage may not be significantly lower than for per-flow reservations.

Variations of the server-based approach have been described by Zhang *et al.* [5], Schelen and Pink [6], Berson *et al.* [7] and Ong *et al.*[8]. In these proposals, in each domain, a bandwidth broker (or agent) is responsible for selecting and setting up the aggregated reservation sessions. This approach has the advantage of removing the message processing and storage burden from routers. However, synchronizing reservation information among the bandwidth brokers and the border routers may be complex. Furthermore, the processing overhead remains, and is simply moved to a different network entity. Also, care has to be taken not to introduce a new single point of failure for the whole domain.

2 Terminology and Assumptions

The Internet is composed of a number of *domains* that exchange user traffic among each other. Each domain is under a common administration, and has a unique Autonomous System (AS) number to identify itself. A domain in the Internet can be classified as either a *stub* domain or a *transit* domain. A domain is a stub domain if the path connecting any two routers i and j goes through that domain only if either i or j is in that domain; transit domains do not have this restriction. A domain can connect to a number of domains via *border routers*. We only consider the delivery and processing of reservation messages between border routers. We assume that each domain can manage its own network resources via static resource provisioning or intra-domain reservation signaling protocols. Since we are focusing on

[1] See, for example, http://www.telstra.net/ops, http://www.employees.org/˜tbates/cidr-report.html, http://www.nw.com

providing inter-domain level resource reservation to existing ISP backbones, we do not consider multicast support for now.

3 BGRP Operation

To set up a reservation over multiple domains, our approach requires two control messages, PROBE and GRAFT. The PROBE messages are used to select reservation path; the GRAFT messages set up the actual reservations. The PROBE and GRAFT messages specify the desired reservation parameters in terms of rate and traffic class. This approach is very similar to RSVP, where senders and receivers periodically generate PATH and RESV messages to set up and maintain reservations. However the key difference is the following: the RSVP PATH was intended mainly to install the appropriate routing state at intermediate routers to pin-down reservation path. Path pin-down requires the routers to keep both reservation sender and destination information. In a network that has N nodes, this could result into of maintaining up to $O(N^2)$ states within the network.

In our approach, the PROBE messages travel downstream from stub domain border routers and are delivered between border routers. They consist of an aggregated reservation request and destination network information. Each border router interfaces with BGP [9] and QoS bilateral agreement to forward the PROBE messages. Instead of installing routing state at routers, the PROBE messages collect routing information along the path. This is equivalent of gathering source routing information for the data traffic.

At reservation set-up time, the reservation sinks send GRAFT messages upstream. The GRAFT messages use the previously collected routing information and traverse exactly the reverse direction of the PROBE messages. The routers on the path aggregate the reservations along a sink-based tree, where the root is the reservation sink, and the leaves are all the reservation sources. Each traversed router can interface with intra-domain protocols to set up transit reservations within each domain.

Routers only need to store the reservation information for the sinks, but not sources. Hence it can potentially reduce the maximum number of reservation flows at each router from $O(N^2)$ in RSVP to $O(N)$, where N is the total number of reservation nodes in the network.

4 Discussion

Message Maintenance: We use "soft state" to manage aggregated reservation state among routers. This requires a new control message, which we call a REFRESH message. Each BGRP router periodically refreshes its adjacent upstream and downstream border routers. If reservations are not refreshed, the BGRP routers will remove the corresponding resource. Since all BGRP messages are delivered reliably (via TCP, for example), the refresh period can be relatively large.

Sink Selection: In BGRP, reservation sinks are BGP border routers. They are chosen dynamically. For example, each PROBE message carries the reservation destination information. When a border router intercepts the message and realizes (by reading own IGP and BGP routing tables) the destination is within its own domain, it terminates the PROBE message. At the reservation time, the border router sends a GRAFT message toward the reservation originator with its own IP address as the reservation sink.

Loop Prevention: Since we are only dealing with unicast, the potential reservation looping problems can be solved within the context of underlying unicast routing protocols. On the other hand, reservation looping problem being observed in RSVP was due to the nature of multicasting.

Reservation with Hysteresis: It's unrealistic to trigger BGRP messages for every end-user's small reservation request. To reduce the control message volume, we allow at least some tree nodes to routinely reserve more bandwidth between themselves and the tree root than is currently required. One can think of this as aggregated advance reservations on behalf of unknown parties.

5 Scaling Benefits

At worst, the total number of reservations that a router has to maintain is in proportion to the number of domains. Currently, the Internet encompasses approximately 4,500 domains. Assuming that, as in RSVP, each reservation state requires 350 bytes[10], BGRP state consumes approximately 1.5 MB at each border router. This is far less than the storage space for routing protocols; BGP routing tables, for example, may occupy hundreds of megabytes of memory in a backbone router.

The reduction in the number of states causes a proportional reduction in message processing. First, since we allow routers to over-reserve bandwidth, small reservations can join and leave the sink tree without disturbing the entire tree. Second, the total number of refresh messages has been reduced. In RSVP, refresh messages are unreliable and thus need to be repeated at about three times the state-timeout interval. In BGRP, all refresh messages are transferred reliably hop-by-hop, reducing the needed refresh frequency. We only need the refresh mechanism to remove state orphaned due to route changes and failures. As in RSVP, the refresh frequency can be tuned according to the stability of the network. Finally, since reservations are aggregated, the total amount of bandwidth required is small. For a backbone link with 4,500 trees and messages of 350 bytes sent every 30 seconds, the refresh bandwidth is only 420 kb/s.

6 Conclusions and Future Work

We believe that BGRP scales well in terms of state storage, processing and bandwidth, making it suitable as an inter-domain resource reservation protocol. The use of soft state provides robustness. Since sink trees are created in the opposite direction as routing announcements, the reservation aggregation of BGRP improves as route aggregation becomes more effective and popular. Since BGRP does not interfere with the local resource management in each domain, it can be deployed without requiring changes to local resource management.

This paper presented the basic architecture and protocol design. We are planning to analyze BGRP's performance in terms of scaling and bandwidth utilization, develop the interface with various inter-domain routing protocols, evaluate the use of source trees instead of sink trees and investigate how BGRP behaves with different amounts of route flapping.

References

[1] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan, "Extensions to RSVP for LSP tunnels," Internet Draft, Internet Engineering Task Force, Feb. 1999. Work in progress.

[2] R. Guerin, S. Herzog, and S. Blake, "Aggregating RSVP-based QoS requests," Internet Draft, Internet Engineering Task Force, Nov. 1997. Work in progress.

[3] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: a new resource ReSerVation protocol," *IEEE Network*, vol. 7, pp. 8–18, Sept. 1993.

[4] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation protocol (RSVP) – version 1 functional specification," Request for Comments (Proposed Standard) 2205, Internet Engineering Task Force, Sept. 1997.

[5] L. Zhang, V. Jacobson, and K. Nichols, "A two-bit differentiated services architecture for the internet," Internet Draft, Internet Engineering Task Force, Dec. 1997. Work in progress.

[6] O. Schelen and S. Pink, "Aggregating resource reservation over multiple routing domains," in *Proc. of Fifth IFIP International Workshop on Quality of Service (IwQOS)*, 1998.

[7] S. Berson, R. Lindell, and R. Braden, "An architecture for advance reservations in the internet," technical report, ISI, Marina del Rey, California, July 1998.

[8] F. Reichmeyer, L. Ong, A. Terzis, L. Zhang, and R. Yavatkar, "A two-tier resource management model for differentiated services networks," Internet Draft, Internet Engineering Task Force, Nov. 1998. Work in progress.

[9] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," Request for Comments (Draft Standard) 1771, Internet Engineering Task Force, Mar. 1995.

[10] P. P. Pan and H. Schulzrinne, "YESSIR: A simple reservation mechanism for the Internet," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Cambridge, England), July 1998. also IBM Research Technical Report TC20967.