S. Berson
S. Vincent
USC/ISI
August 1998

Aggregation of Internet Integrated Services State

August 7, 1998

Status of Memo

Abstract

The Internet Integrated Services (IIS) architecture[4] has a fundamental scaling problem in that per flow state is maintained at all routers and end-systems supporting a flow. This draft examines the use of aggregation as a technique to reduce the amount of state needed to provide IIS, and describes the modifications to RSVP to support aggregation. In our approach, routers at the edge of a region doing aggregation keep detailed IIS state, while in the interior of this region, routers keep a greatly reduced amount of state. Packets will be tagged at the edge with scheduling information that will be used in place of the detailed IIS state. The aggregation scheme described will allow large scale deployment of IIS without overloading routers with state and associated processing.

1 Introduction

In order to deploy Internet Integrated Services (IIS), additional resources are needed in the routers to store state and to process packets. As currently defined [3, 4], this state is on a per session basis, where a session is a unicast or multicast destination and optionally a port. IIS state is stored at all routers between the source(s) and destination(s) of a session. Thus, widespread use of IIS would mean a large amount of state at routers in the core of the Internet. This large amount of state and related processing could overwhelm routers, so IIS state models that do not require per-session state at all routers need to be explored.

Several types of per-session state are used with integrated services. First, there is scheduling state that consists of the different traffic service queues for packet forwarding. Second, there is packet classifier state. Classifier state is used to assign each arriving packet to a packet scheduling queue for forwarding. Third, there is the state for the setup protocol, e.g. state carried in RSVP PATH and RESV messages. Finally, current multicast routing algorithms also store state, either per source and session (e.g. DVMRP[9], PIM-DM[5]) or per session shared-tree (e.g. CBT[1], PIM-SM[5]). While approaches to aggregation of routing state are similar to aggregation of IIS state, they are beyond the scope of this draft.

Per session state in a router has two detrimental effects. First, additional state consumes memory, and second, additional state requires additional CPU cycles to process the state. Depending on the speed with which the memory needs to be accessed, different types of memory can be used. State that is required in order to process each packet (e.g. classifier state, scheduling state, and multicast routing state) needs to be stored in fast (i.e. cache) memory, while other state (e.g. setup protocol state) can be stored in slower memory. Since fast memory is substantially more expensive than slow memory, and since accesses to fast memory are on the router fast path, our primary goal is to reduce the classifier and scheduler state. Reducing the amount of setup protocol state is an important, but secondary goal.

Any aggregation scheme entails some tradeoffs. Keeping full integrated services state at all routers allows the resources dedicated to the flow of packets from each admitted session to be isolated from the resources for packets from all other sessions. This isolation means that resources reserved and needed for one session will be used only on the reserving session. The disadvantage of doing aggregation is the loss of some portion of this flow isolation, since with aggregation many flows would share the same traffic service class.

This draft describes how to provide Internet Integrated Services for both unicast and multicast traffic with greatly reduced state requirements. We concentrate primarily on Controlled Load type service[10], which reserves bandwidth but not delay or jitter, though our approach should be more generally applicable (see [7] for a related approach). With our approach, the amount of scheduler state is bounded at all nodes for both unicast and multicast traffic. For unicast traffic, full classifier state and setup protocol state are stored only at the edges of the network. In the interior of the network, no unicast setup protocol state is stored, and only a bounded amount of classifier state is used. For multicast traffic, a minimal amount of additional classifier state and additional setup protocol state is used in the network interior.

We make several assumptions:

1. We assume that an explicitly definable region (e.g. one or more contiguous autonomous systems), called an *aggregating region*, will aggregate and will implement supporting mechanisms at its boundary points. We define an *ingress* of an aggregating region as a router where a packet for a specific unicast or multicast session enters the region, and an *egress* as a router where a packet for a specific session leaves the region. Also, we define an *interior router* for a session as any router in the aggregating region that is on the data path for the session and is not the ingress or egress.

2. We assume that the choice to aggregate and the mechanism used to do so is an intra-domain issue, not an inter-domain issue and therefore there can be variability across regions so long as at the boundaries routers continue to pass along adequate messages to support the setup protocol upstream and downstream.

3. We assume current multicast and unicast routing within the aggregating region; but with no other enhancements.

2 Framework

Since each network service provider will want to make decisions on how to provide integrated services based on their provisioning and traffic patterns, a brief review is presented of how IIS state is used.

2.1 Integrated services state

Scheduler classes are the unit of assignment of network resources. Scheduler state typically takes the form of different traffic classes. These traffic classes are assigned different priorities, link shares[6], and policing policies to provide a certain level of service. In the basic IIS model, each traffic flow gets its own traffic class, meaning that each flow has certain resources dedicated to it. By reducing the number of traffic classes, a coarser granularity of resource assignment is done in that several flows will be assigned to the same scheduler queue. A consequence of the coarser granularity is that the aggregated traffic gets a certain level of resource usage, but it is impossible inside an aggregating region to isolate flows in the same traffic class from each other. This lack of isolation can be mitigated by adequate provisioning of network resources and by policing at aggregating region borders.

Classifier state is used to assign packets to traffic classes. Each arriving packet is classified into a traffic class at each node according to state established by the setup protocol. In the complete absense of classifier state, no service differentiation is possible. However a packet can be classified at the ingress, and the results of that classification can be encoded in each packet. This reduced classifier state encoded in a packet can be as simple as one or more bits segregating different classes of traffic, or alternatively, an encapsulation header. By encoding a small amount of classifier state in the packet, only a very simple classification is needed at interior routers.

Finally setup protocol (e.g. RSVP) state is used to do admission control, provide policing, allow reclassification, and allow merging of reservation messages. The first two of these issues, admission control and policing, are general issues involving both unicast and multicast traffic, while the last two, reclassification and reservation merging, are only issues with multicast traffic.

2.1.1 General setup protocol state issues

Setup protocol state is used in admission control. Setup protocol state contains the amount of resources reserved for each admitted flow at a router. Admission control with full IIS state involves computing how much network resources are already committed to existing reservations and whether there are sufficient resources to admit a new flow with certain resource requirements. In the absence of setup protocol state, admission control decisions can be made only on the basis of the reservation request control message and load measurements on the router. We call this type of admission control *measurement based admission control*[8].

Setup protocol state is also used for traffic policing. Policing is done in an IIS network at traffic merge points and at traffic split points. A traffic merge point (see Figure 1a) is where the traffic from two or more sources for the same session merge. For an RSVP shared style reservation, the traffic from all the sources needs to be policed to the reservation parameters at traffic merge points. This means that though the traffic arriving on links L1 and L2 is within the reservation parameters, the sum of the traffic exiting on link L3 may exceed the reservation parameters. In this case, policing is needed on link L3.

Policing is also done at traffic split points (see Figure 1b). A traffic split point is where multicast traffic has multiple outgoing interfaces. Because of RSVP reservation merging and because some outgoing interfaces may have differing reservations, traffic at a split point needs to be policed on each outgoing interface to the proper reserved values for that interface. In the figure, link L5 has a reservation Q1 while interface L6 has a reservation Q2. Policing is needed to make the traffic on the outgoing links conform to the link reservation.

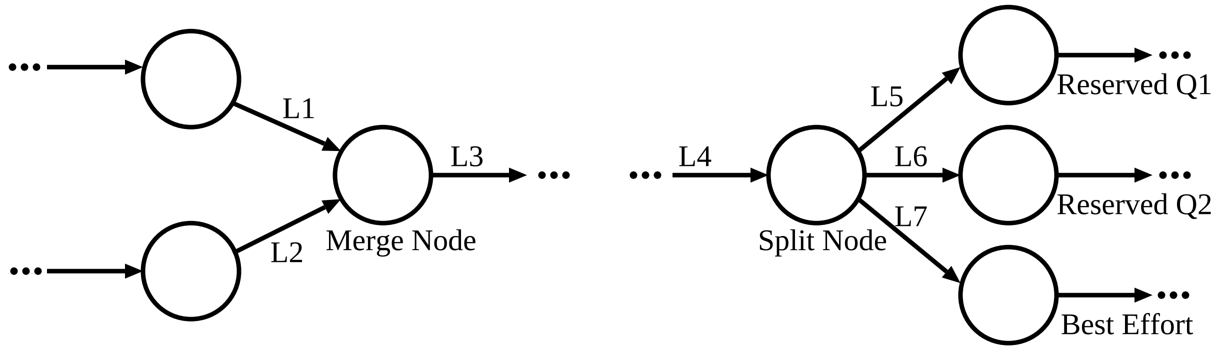

Figure 1: Traffic split and merge nodes

2.1.2 Multicast setup protocol state issues

Setup protocol state is used to allow reclassification of packets from reserved to best effort. This reclassification is necessary at traffic flow split points (see Figure 1b) where there are one (or more) outgoing interfaces with reservations and one (or more) outgoing interfaces without reservations. In the figure, links L5 and L6 have reservations, while link L7 has no reservation and is forwarding best effort. Since data packets are marked with their with their service class, the packets may need to be remarked at traffic split points. At these points packets forwarded to outgoing interfaces with no reservations such as link L7 need to be reclassified as best effort.

Finally, setup protocol state is used to merge reservation messages coming from the egresses to the ingress. With setup protocol state for each session, and hop by hop processing of reservation messages, each router can limit the number of reservation refreshes it sends. Without setup protocol state, each reservation refresh message from each egress would be sent to the ingress, requiring a large amount of processing for any large multicast groups.

2.2 Basic aggregation architecture

We propose a scheme for unicast traffic with a constant amount of scheduler state at all routers, a constant amount of classifier state at interior routers, and no setup protocol state at interior routers. For multicast traffic, we propose a scheme with bounded scheduler state, and greatly reduced classifier and setup protocol state. In our approach, a fixed number of traffic service classes are defined and configured at all routers in an aggregating region. Each traffic service class is subject to admission control at the ingress to the region. Traffic policing (and reclassification) is done at the ingress for unicast traffic, and at the ingress plus at traffic split points for multicast traffic. Since only a fixed number of traffic classes are available, the scheduler state at all routers is constant.

Data packets are classified and *tagged* on entry to the aggregating region with some identifier indicating to which aggregated traffic class the packet should belong. This identifier can consist of some bits in the packet header (e.g. type of service bits) or the packet could be encapsulated. The tag (or encapsulation) encodes the traffic service class that this packet will receive across the aggregating region. Tagging or encapsulating each packet bounds the amount of classifier state in the interior of the aggregating region.

Full setup protocol state is stored at the borders of the aggregating region, but only a small amount of multicast setup protocol state is needed in the interior of the region. When a reservation message arrives at an ingress to the region (from an egress), and that reservation passes admission control, then packets from the flow associated with the new reservation are assigned to one of the configured traffic service classes.

Admission control will be measurement-based, and, since there is no unicast and minimal multicast setup protocol state in the interior, admission control will be done at the ingress for a session. But the decision will be based on congestion measurements within the aggregating region. Thus each node in the aggregating region, upon receiving an admission control setup protocol message, will make a local congestion decision. This local congestion decision will be made on a threshold basis where new reservations are not admitted if the existing measured load plus the expected additional load from the new flow is greater than a (possibly dynamic) threshold.

To implement the local congestion decision, each node keeps an estimate of its current load per traffic class. As an admission control message traverses the path between ingress and egress, each interior router must admit that reservation. Each router that provisionally admits the reservation should factor the new reservation into its currect traffic estimate. To admit the flow, an estimate of the amount of the reserved traffic due to this reservation must be made, e.g. based on peak rate specified in the admission control message. If some router does not admit the reservation, the admission control message is marked appropriately by the rejecting router and the message is forwarded. When the admission control message arrives at the egress, the egress node forwards the control message back to the ingress. The ingress checks the message to see if any router rejected the message. If some router rejected the reservation, then the ingress will reject the reservation and send a reservation error message.

In summary, our approach provides for reduction of packet scheduler state, packet classifier state, and setup protocol (e.g. RSVP) state in the following ways. A constant amount of packet scheduler state at all routers in an aggregating region is achieved by using preconfigured traffic service classes. A constant amount of packet classifier state for unicast traffic at all interior routers in the aggregating region is accomplished by tagging or encapsulating each packet with its traffic service class. No setup protocol state at interior routers for unicast traffic is achieved by doing admission control and policing only at the edges of the aggregating region. For multicast traffic, only a small amount of RSVP state is stored at routers, while classifier state is stored only at traffic split points and only on those outgoing interfaces with no reservation.

2.3 Multicast setup protocol state

We would like to apply the above approach to multicast traffic to eliminate all of the setup protocol state in the network interior. However, there are some fundamental differences between unicast and multicast traffic that results in additional setup protocol state being desirable. Since traffic with multiple destinations is being classified and tagged at the ingress router, a tagged multicast packet will receive reserved service everywhere in the cloud, including branches for which there are no reservations. This all-or-nothing reservation property, caused by tagging at the ingress, complicates aggregating state for multicast sessions. The complications derive from two features of multicast sessions, *heterogeneity* and *dynamic group membership*. Heterogeneity refers to the situation where different branches of a multicast distribution tree have different reservations, including the case where some branches have no reservation. Unreserved packets receiving reserved service can adversely affect packet flows that properly have a reservation. Without additional state, there is no way to determine which packets have a legitimate reservation.

The other multicast feature causing complications is dynamic group membership. Dynamic group membership means that the members of a multicast session can be changing over time by having new receivers join, and old receivers leave. Due to the all-or-nothing property, a new best effort receiver joining a multicast session with a reservation in place will be receiving reserved service. Admission control will not have been done for that new receiver, and so those reserved packets to the new receiver can cause congestion for other properly reserved packets. Thus a best effort receiver can disrupt reserved traffic.

Our approach to aggregation and multicast is to keep some additional setup protocol state for multicast sessions but only at split nodes. Since setup protocol state is not on the router fast path, this option is reasonable since scheduler state and classifier state are still minimized. Some additional classifier state is needed, but only for sessions with heterogeneous branches, and only at those nodes with heterogeneous outgoing interfaces. This classifier state can be established dynamically from setup protocol state and routing state. When a node discovers that it has heterogeneous outgoing interfaces, classifier entries are established on the best effort branches. The packets that the classifier matches have the reserved markings removed and new markings inserted indicating that the packets are to be forwarded as best effort.

2.4 Policing in the interior

There is still an issue with state aggregation and shared (e.g. RSVP wildcard or shared explicit) style reservations for unicast packets at traffic merge points. At a merge point, the traffic entering the node may conform to the reservation, but the traffic exiting the node may be non-conforming due to multiple senders. With setup protocol state, the traffic would be policed at the merge node. Since there is no unicast setup protocol state in the interior of the network, it is impossible to police the traffic of an individual flow in the interior. The traffic will eventually be policed at the next ingress, but may interfere with actual reserved traffic between the merge point and the egress. Since the traffic will eventually be policed, there is no "free bandwidth" for a user trying to exploit this feature. However there is a security problem where a malicious user could tie up reserved bandwidth traffic in a transit network. The simplest solution to this problem is to limit or prohibit unicast shared style reservations, which are of limited usefulness. Other approaches involve detection at the egress of abusive behavior, and cancelling the offending reservation.

3 RSVP operations/extensions

In this section we describe how this aggregation scheme would be applied to RSVP. We assume that a measurement based admission control algorithm[8] is defined, and we describe how to utilize this system to implement aggregation. The basic issues are gathering the measurement data, keeping track of which reservations are new, and describing the appropriate data structures.

3.1 RSVP protocol changes for aggregation

In order for RSVP to support aggregation of IIS state, two new RSVP message types are needed, admission request (ADREQ) and admission reply (ADREP). The ADREQ is used by an ingress to set up the reservation across the aggregating region, while the ADREP is used by the egress to report that the reservation has succeeded or not.

In addition to new message types, an admission control object (ADMISSION), and an admission control status (STATUS) object types are needed. The ADMISSION object and zero or more STATUS objects are included as part of a ADREQ or ADREP message. The ADMISSION object contains a common object header, a handle from the ingress node, and some flags. The handle is included by the ingress node on initiating aggregated admission control and is used to uniquely identify reservation state on the ingress. The IP address of the ingress is stored in an RSVP_HOP object in the ADREQ message. The STATUS object has a common header, an RSVP_HOP, and a load status. The RSVP_HOP object contains the IP address of the node supplying the load report, while the load status is the measured load from a node.

The format of the ADREQ and ADREP messages are as follows:

```
<ADREQ> ::= <Common Header> <FLOWSPEC> <ADMISSION> <RSVP_HOP>
               <status_report>

status_report ::= <empty> | <status_report> <STATUS>

<ADREP> ::= <Common Header> <ADMISSION> <status_report>

struct {
        Object_header   header;
        Handle          handle;
        int             flags;
} ADMISSION;

#define ADMITTED 1
#define REJECTED 0

struct {
        Object_header   header;
        RSVP_HOP        sess;
        Load            load;
} STATUS;

#define RSVP_ADREQ      11
#define RSVP_ADREP      12
```

3.2 RSVP behavior - unicast

Our approach to collecting measurement information for admission control is to originate the state collection from the ingress upon receiving an RSVP reservation message (RESV). When an ingress receives a new RSVP RESV message (see figure 2) and local admission control is successful, the ingress initiates aggregated admission control by sending an ADREQ message. The FLOWSPEC from the RESV message is used in the ADREQ message, as well as an ADMISSION object containing a handle chosen by the ingress. The basic exchange of messages is shown in Figure 2. Note that no processing is needed in the interior of the aggregating region on the RESV or the preceding PATH messages. But when the ingress receives a new RESV message (i.e. a message for which

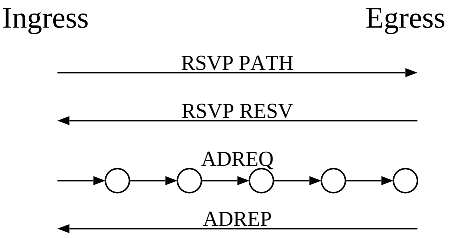

Figure 2: Admission control exchange of messages

it has no traffic control state), the ingress initiates admission control. For admission control, the ingress sends an ADmission REQuest (ADREQ) message hop-by-hop through the interior of the aggregating region to the egress (known from the previous RSVP hop field in the RESV message). At each hop through the region, the node attempts to admit the reservation. If admission control succeeds at a node, the ADREQ message is forwarded with an optional status object to the next hop. If admission control fails at a node, a failure status object is appended to the ADREQ message and then the message is forwarded. When the egress receives the ADREQ message, an ADREP message is generated and sent back to the ingress containing all the status objects. The ingress checks if any interior node appended a failure status object to the message. If there is no rejection information in the ADREP message, then the reservation is accepted and the packets for this flow will be marked as reserved. If there is a failure status object in the message, the reservation is rejected and a reservation error message is generated.

Note that if either an ADREQ or an ADREP message is lost, then the ingress behaves as a router that lost an RSVP reservation message. The ingress will wait until the reservation message is refreshed and then send a new ADREQ.

3.3 RSVP behavior - multicast

Multicast behavior is more complicated due to the dynamic nature of receivers joining and leaving groups and due to the possibility that some hosts request reservations while other hosts do not. In figure 3, there is a sender $s1$ upstream of router $R1$ and receivers $r4$ and $r5$ downstream of routers $R4$ and $R5$. If $r4$ has a reservation for a session and $r5$ does not, it is important that traffic for that session with reserved markings do not travel on the R3-R5 link (as the traffic commitments to other sessions over that link might be disrupted). To prevent marked traffic from transitting links without a reservation, a *reclassifier* needs to be installed on the unreserved link (e.g. R3-R5) to remove the reserved traffic bits in the data packets. With the reserved bits removed, the traffic will travel as ordinary best effort.

In order to reclassify the traffic, RSVP state will be needed at traffic split points (such as $R3$). The

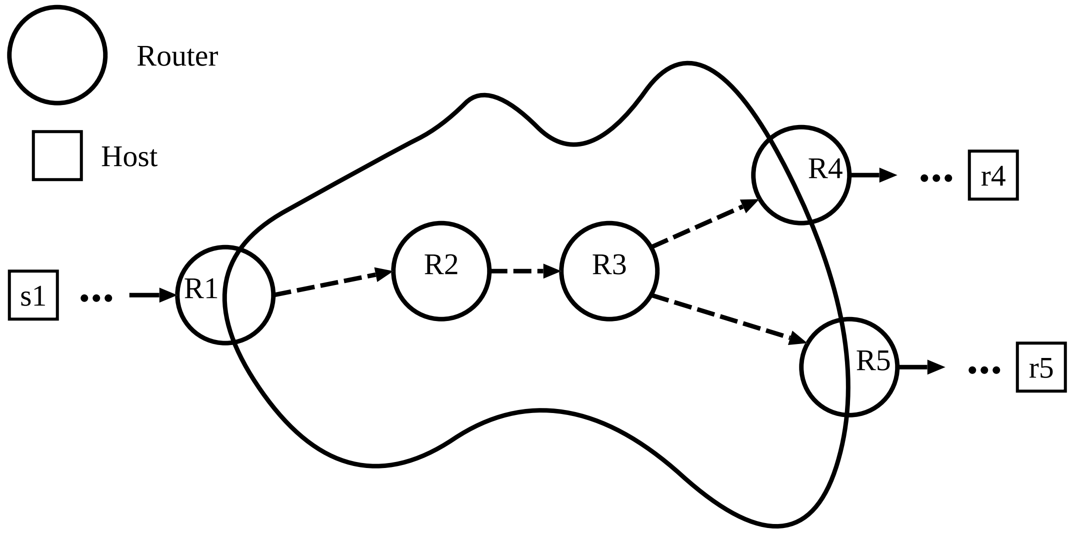

Figure 3: Internet aggregating region

accumulating of RSVP state is triggered by the formation of a split point in data traffic. When a second (or higher) multicast session join (e.g. over link R3-R5) reaches a router (e.g. $R3$), the router needs to activate RSVP state management for that router for that session. In addition, the router needs to establish a reclassifier on the newly joined link for the session.

The next PATH message for this session will be received and processed by $R3$ (and state stored). If a RESV message is then received on R3-R5, an ADREQ message will be sent out the link. When an ADREP message is received from the egress (or next downstream split point), the reservation is in place. When the reservation is in place on all outgoing links at a node for a session, any reclassifiers are removed. Also, if $R3$ receives an ADREQ from $R2$ for this session, an ADREP is generated in response.

When the penultimate outgoing link is removed from the multicast session at a node, the RSVP state accumulation may end, and the existing state for this session may be purged.

3.4 Other approaches

There are three other possible approaches to aggregated admission control that may be useful in certain types of network environments, but which have serious limitations in general. One approach is to put the admission control request in RSVP reservation messages. This approach is conceptually simpler than the above scheme, but assumes that the reverse route of the data (i.e. egress to ingress) through the aggregating region is known. There are cases where this holds, most notably in networks using link state routing protocols, but in general this is not the case.

The second approach to collecting admisson control information is an RSVP path message which travels on the forward route from data ingress to data egress. When this message is processed at each interior node, admission control information is collected. The accumulated admission control information is then included in a corresponding RSVP reservation message from egress to ingress. The disadvantage with this scheme is that each PATH message would need to be processed by interior routers. ADREQand ADREP messages are only generated on a new reservation.

The third approach to collecting admission control information is by out-of-band communication, e.g. as part of routing protocols. In this case, the ingress site can do the admission control based on the out-of-band communication. Further work is needed in this area.

4 Reserved traffic congestion

One last issue is traffic congestion in reserved traffic classes. There are two ways that admitted load can cause high levels of congestion on a router, (1) measurement based admission control failure and (2) link failure. Measurement based admission control failure occurs due to past load not being representative of the future. For example, if many idle reserved sources become very active at the same time, some links may become congested. Link failure is caused by a link being declared down and a new set of routes are generated. The traffic that was traversing the failed link may now be added to an already heavily loaded link. In both of these cases, links may become overcommitted.

For small amounts of congestion, reserved traffic can preempt best effort traffic, i.e. best effort traffic packets are dropped, but new admission control requests are rejected. For larger amounts of congestion, reserved packets need to be dropped or reservations need to be preempted. For link failure in a per-flow IIS network, the setup protocol would typically use reservation preempting (while admission control mistakes are expected not to happen). With aggregated IIS state, the point at which the routing changes may not have setup protocol state. Thus a node typically cannot distinguish an overload caused by a link failure from one caused by transient heavy congestion. The response is the same for either cause of congestion; the ingress nodes in this case will need to preempt reservations.

5 Conclusions

This draft has described a scheme for providing Internet Integrated Services with greatly reduced state requirements. This scheme bounds the scheduler state at all routers in an aggregating region. For unicast traffic, the classifier state and setup protocol state is stored only at the border routers. For multicast traffic, full classifier state is stored at the border routers, while a limited amount of additional classifier and additional setup protocol state is stored at traffic split points.

6 Acknowledgements

This draft is the result of discussions with many people particularly Bob Braden and Bob Lindell.

References

[1] Ballardie, T., Francis, P., and Crowcroft, J., "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing," SIGCOMM '93.

[2] Berson, S., and Vincent, S., "Aggregation of Internet Integrated Services State," IWQoS '98.

[3] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S., "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205.

[4] Braden, R., Clark, D., and Shenker, S., "Integrated Services in the Internet Architecture: an Overview," RFC 1633.

[5] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, C., and Wei, L., "An Architecture for Wide-Area Multicast Routing," SIGCOMM '94.

[6] Floyd, S., and Jacobson, V., "Link-sharing and Resource Management Models for Packet Networks," IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp. 365-386, August 1995.

[7] Guerin, R., Blake, S., and Herzog, S., "Aggregating RSVP-based QoS Requests," Internet Draft.

[8] Jamin, S., Shenker, S., and Danzig, P., "Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service," Infocomm '97.

[9] Waitzman, D., Partridge, C., and Deering, S., "Distance Vector Multicast Routing Protocol," RFC 1075.

[10] Wroclawski, J., "Specification of the Controlled-Load Network Element Service," RFC 2211.

Security Considerations

Security considerations have not been addressed in this draft.

Author's Address

Steven Berson
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

Phone: +1 310 822 1511
EMail: berson@isi.edu

Subramaniam Vincent
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

Phone: +1 310 822 1511
EMail: svincent@isi.edu