SRP – an alternative approach

Tony Jeffree

Summary

The existing SRP draft has been a useful starting point for discussions on the reservation process and how it can be made to work. However, I believe that the current approach can be simplified – in terms not only of how SRP is documented, but also how it is implemented in L2 devices, and how the separation between L2 and higher layer functions is successfully maintained. I believe the key to this simplification is to recognize what is, and what is not, the business of the L2 devices that participate in the process, and to leverage existing technology in the form of the recently completed MMRP.

1. Introduction

My starting assumption is that as far as the SRP standard is concerned, the problem that we are trying to solve is how we associate reservation information (in the form of some kind of bandwith/traffic class specification) with stream information (in the form of MAC addresses) contained in the Filtering Databases of Bridges, in order to allow the Bridge to make meaningful forwarding decisions based on that data[1].

I don't believe that, at least in this project, we are in the business of figuring out how the end stations and stream servers might negotiate between themselves as to what streams exist, what high-level tags or names they might use to communicate about them, and how that high level data might get translated into stream identifiers that are going to be meaningful to Bridges, i.e., MAC addresses. I also don't believe that we are in the business of providing some kind of transport protocol at L2 for communicating higher level data about streams and reservations.

That is not to say that the other set of problems isn't interesting, and likely form a necessary part of the overall solution to AV; just that it fundamentally isn't the job of SRP to solve them. That being the case, there may well be one or more further projects needed in order to complete the jigsaw; alternatively, it might well be our conclusion that getting into what amounts to an Application layer protocol development isn't 802.1's job. Either way, that particular discussion hasn't (as far as I can tell) really taken place yet.

In other words, at least part of what I am trying to achieve with this proposal is to take what has proved to be an extremely successful approach over the history of 802.1 – that is, to focus on a relatively limited scope problem that can be solved simply, and avoiding the potential pitfalls of widening the scope to the point where the project can't complete.

2. What are streams anyway?

The idea of data streams isn't new to the 802.1 Bridge standards; providing support for data (video, voice…etc.) streamed to multicast destination addresses dates back to the mid 90's, and was the driving force behind the development of GARP and GMRP (originally standardised in 802.1D:1998, now re-incarnated as MRP and MMRP in 802.1ak). The only real difference, in terms of the starting assumptions stated in 1), between what we are looking at now and what we were looking at then, is the decision to associate bandwidth reservation information with a stream, and apply traffic shaping to it in the forwarding process accordingly.

Hence, the first and most obvious simplification that can be made in SRP is simply to leverage the support for streams that already exists in our standards – namely, MMRP. It is there, and it can already be used, without modification, to create a subtree of the Spanning Tree that provides a forwarding path between a stream data source and any

[1] The job of specifying those forwarding decisions is part of P802.1Qav; obviously there will be some level of interaction needed between these two projects to get it right.

registered end stations; as can be seen later on, the detail of whether or not a stream becomes "active" at the point where MMRP is used to register an address is something that can be handled by the data source and sink, if it is important to them.

The first consequence of going down this route is that SRP becomes a single protocol, one that is concerned only with making and maintaining reservation decisions, rather than two protocols. There will still be some need for interaction between MMRP and the reservation mechanism; however, this can successfully be achieved without affecting the specification of MMRP in any way.

The second consequence is that, as streams are (as far as SRP is concerned) identified only by a MAC address, we lose one of the optimizations of the original proposed mechanism – that stream registrations propagate only in the direction of the stream data source. In the target networks (AV networks in a home or studio environment) I don't believe that this particular optimization (or its absence) is an issue – the whole point of the 802.1ak project was to improve the scaling properties of MRP-based applications, so that their use made sense in small networks, and so they could successfully scale for use in large networks. I will grant the fact that all Bridges in an AV network (or at least, all that exist on the path between stream data sources and sinks) will need to implement the mechanism; however, given that they have to support it, the extra overhead of processing registrations on Ports that don't reach a data source is insignificant, as is the bandwidth consumed by the registration PDUs. Given that the MRP PDU structure can represent registration state information for 4K streams in a single PDU (assuming the stream addresses are consecutive), the real protocol overhead, and consequently, the real optimisation opportunity, is small[2].

[2] Certainly too small to justify inventing a new MRP application when one already exists that can do the job.

3. Registering and de-registering for a stream

A listener can register for a stream, using MMRP, at any time. There are two cases to consider:

- The listener is the first in the network to register for the stream.
- The listener is not the first in the network to register for the stream.

In the first case, the talker for that stream can make use of the MMRP state information to detect the fact that there is now a listener registered for that stream where none existed before. The concept of "source pruning" is already described in MMRP; the presence/absence of a registered recipient is used to forward/filter frames destined for the registered address – the data source effectively behaving like a single port Bridge. That same mechanism can be used to trigger the reservation protocol to establish reservation information for that stream along the path from the talker to the listener.

The degree to which the listener cares about the success or failure of the reservation process seems to be an application choice. If the application concerned is tolerant of degraded service, then it would seem to be a perfectly reasonable choice for the listener to completely ignore the reservation protocol. At the other extreme, an application that is sensitive to the provided QoS might listen in to reservation messages, and determine whether or not to de-register for the stream (or take some other action) depending on success or failure of the reservation. There is probably a middle ground here too, where the listener might accept degraded service on a stream, but might listen in to the reservation protocol in order to be able to report the service level to its user[3].

From these observations, it would seem that the reservation protocol can be a one way (source to sink) transmission of data; the source can determine whether anyone is still

[3] This effectively means that there can be two types of streams in the world; those that have reservations, and those that don't. It is up to the application concerned to decide which of these it prefers and act accordingly. In some cases, it could be possible to successfully operate stream-based applications in a bridged LAN that didn't support SRP at all, but did support MMRP.

interested in the stream, at the available QoS, simply by monitoring the registration state of the stream.

In the second case, where a new listener joins an already established stream, all that is required over and above the first case is for one or more of the intervening Bridge(s) to recognize that the stream has become registered on a new Port, and to propagate the reservation information that it already knows for that stream in that direction.

De-registration is simply the reverse of the above; as registered listeners go away, Bridges will stop propagating reservations for that stream on Ports that no longer support listeners, and eventually, when the last one goes away, the talker will stop originating the reservation information for the stream and will stop transmitting the stream data.

4. Stream reservation

As observed in 3), it looks as if this is a simple “declarative” protocol, where reservation information originates from, and is refreshed by, the stream data source, and is modified and propagated by Bridges only along the path(s) towards any registered listeners.

The protocol needs to convey:

- The stream ID (a MAC address);
- Reservation information, as required for establishing the parameters needed for the operation of Qav[4];
- Not a lot else. For example, given the assumption stated in 1) that some higher layer negotiation mechanism exists that allows the establishment of the right L2 stream identifiers to be used by an application, there probably isn’t even a need to tie the talker address to the reservation information; the stream ID is sufficient, as the higher layer mechanisms can ensure there would be no more than one talker using any one stream ID[5].

The result is, I believe, certainly no more complex, and maybe a little simpler than the mechanism in the current draft. In particular, it doesn’t carry higher layer information in the protocol that is of no concern to the L2 operation, and therefore removes any temptation for this protocol to become some kind of unspecified transport mechanism for higher layer information that it doesn’t itself understand[6].

5. The filtering database

The purpose of the reservation protocol is to allow us to associate reservation information with a given stream; there will therefore be a need to define reservation data entries of some form as part of the filtering database, in order for the forwarding process to perform whatever de-queuing mechanisms we may decide upon in P802.1Qav.

I’ve used the words “…as part of the filtering database” above deliberately. The filtering database already contains a number of different types of entry – static filtering entries, dynamic filtering entries,…etc. etc., all of which interact in ways that are clearly specified in the 802.1D and 802.1Q standard. However, it is nonetheless conceptually a single database, and adding a new type of entry doesn’t change that. I.e., you might just as easily view the additional reservation information as an extension to the definition of the Group Registration Entry. If there is reservation information there, then the forwarding process uses it; if there isn’t, then it behaves exactly like an existing D or Q Bridge with respect to its forwarding behaviour[7].

---

[4] This should be equivalent to the reservation information described in the current draft.

[5] I guess there’s an open question (at least, in my mind!) as to whether there are potential applications where there are multiple talkers for a given stream, or whether that simply doesn’t figure in the anticipated uses.

[6] Putting this more strongly: The high level tag information included in the current draft isn’t needed by this L2 protocol, its use is not specified at all in the draft, and therefore, I believe it has no business being included in the specification.

[7] I would strongly dispute the assertion in http://www.ieee802.org/1/files/public/docs2007/at-feng-SRP-MMRP-070205.pdf that this approach results in a more complex forwarding engine. I believe that as far as Qav is concerned, this proposal and the one in the current draft should be exactly equivalent.

6. Overall architecture

To summarise the above, the overall structure that needs to be documented in the SRP standard looks like this:

- Talkers and listeners (sources and sinks of stream data) that use some higher layer mechanism (not specified in the SRP standard, but possibly the basis of some future projects) to determine what stream identifiers, in the form of MAC addresses, they will use;
- Stream registrations and de-registrations based on the use of MAC addresses registered by the existing MMRP mechanisms;
- A stream reservation protocol, supported by protocol entities in the talker, the Bridges, and (optionally) the listener, that carries only L2 reservation information for a given stream[8]:
  - The Talker uses MMRP registration/de-registration events to trigger the transmission of reservation information and the transmission of the stream on a regular basis, and to cease transmission when no listener exists;
  - The Bridges use the reservation information received from the Talker (or from an upstream Bridge) to update their FDB with the information needed by P802.1Qav for each stream, and forward the reservation information (modified as appropriate) on any paths where the stream is registered;
  - The Bridges also recognize stream registration/de-registration events, and update/propagate FDB and reservation information accordingly.
  - The listeners take as much notice of the received registration information as is appropriate for the application concerned; consequently, listener support of the reservation protocol is optional.
- Extensions to the existing FDB definition that allow reservation information to be associated with a registered stream, and in a form that allows P802.1Qav to perform its job of traffic shaping for the stream.

[8] I guess we might talk about optimizations that would allow the packing of reservation information for multiple streams into a single PDU, but that is a detail.