Leakage of unreserved streams
Tony Jeffree

Summary

Kevin Stanton's presentation at the November 2010 meeting in Dallas (http://www.ieee802.org/1/files/public/docs2010/ba-kbstanton-multicast-treatment-with-no-reservation-1110-.pdf) identified a problem with the current specification in 802.1Q that can result in disruption of valid stream transmission if frames are queued on a credit-based shaper outbound queue for a destination address that doesn't correspond to a stream that is active on that Port. This paper examines the potential ways that this problem can occur, and what solutions might be implemented in 802.1Q to fix the problem.

Background

In normal operation, a stream is advertised by a Talker, and subscribed to by one or more Listeners. In all Bridges along the path from Talker to Listener(s), the outbound queue(s) that will carry the stream are provisioned with a suitable increment to their idleSlope that allows the stream's bandwidth to be accommodated, and a dynamic reservation entry is created in the filtering database that details the outbound Port(s) that will carry the stream. The reservation mechanism enforces uniqueness of the destination addresses used for stream transmission within a given VLAN; i.e., there is a 1:1 correspondence between streams and destination MAC addresses. Hence, although the idleSlope assigned to a given shaper queue reflects the reservation for multiple streams, the filtering database ensures that a given stream is transmitted only on those Ports on which it has been provisioned.

Failure cases

There seem to be a number of potential failure cases that can result in frames that are not part of a valid stream for a given Port being queued on a credit-based shaper queue, and hence interfering with validly provisioned streams. There seem to be the following cases to consider:

1. Frames that are part of a valid stream on Port X are queued on Port Y because of mis-configuration of the filtering database. This could either be that the filtering database has no entries for the destination address/VID concerned, and therefore the decision is to flood the frame, or it could be that there is explicit information in the FDB, and it has been mis-configured to forward the frame on Port Y. As these frames have not been provisioned on Port Y, they will potentially disrupt normal stream operation for the streams that are provisioned on Port Y.
2. Frames that are NOT part of a valid stream (i.e., have been generated by some other "legacy" application that resides within the SRP domain boundary, and happens to use a priority value associated with an SR class). As these frames are not subject to the priority re-mapping that occurs at the SRP domain boundary (see 6.9.4 of Q-REV), they will be queued on a shaper queue and will potentially disrupt normal stream operation.
3. The idleSlope for the shaper queue is zero; i.e., there is no current bandwidth provision on that queue for stream traffic, but frames are being queued as a result of case 1 or case 2. In this case, the current specification will result in one frame being transmitted, which will cause credit to go negative, and all subsequent frames will be queued until the Bridge times them out.
4. The idleSlope for the shaper queue is non-zero, indicating that there is some stream provisioning on the Port. Frames queued and transmitted as a result of case 1 or case 2 effectively "steal" bandwidth from the streams that are validly provisioned on that

queue; the consequence is potential disruption of those streams and any streams that are provisioned on Ports downstream of this Port.

Potential fixes

The 4 failure cases described above, in combination, could be dealt with as shown in the following table:

	idleSlope = 0 (case 3)	**idleSlope != 0 (case 4)**
Mis-configured stream (case 1)	If the idleSlope is zero, then the shaper is inactive. All queued frames for that traffic class should be discarded.	Ideally, any frames for the mis-configured stream should be discarded from the queue and not transmitted; however, this may not be achievable.
Legacy traffic sourced inside the SRP domain (case 2)	Legacy traffic should not be queued on a shaper queue. Two approaches: (a) discard all such legacy frames, or (b) re-map the priority associated with the frame so that it is queued on a non-shaper queue[1].	Legacy traffic should not be queued on a shaper queue. Two approaches: (a) discard all such legacy frames, or (b) re-map the priority associated with the frame so that it is queued on a non-shaper queue.

For cases 1/3 and 2/3, if the decision is to discard frames for case and 2/3, the fix is trivial. This involves adding a couple of lines of text to the definition of the credit-based shaper, indicating that frames are never made available for transmission, and are discarded from the queue, in the case of a zero idleSlope. This change would have a very minor impact on existing implementations; however, given that the existing behaviour isn't particularly harmful (one frame transmitted per affected queue, then nothing more), it would also have a very minor effect on the undesirable behaviours associated with the other two combinations. This is a change that could be made as part of current 802.1Q revision activity.

If the decision is to re-map frames for case 2/3, then the fix is not so trivial; first, you have to determine whether the frame is part of a valid stream. If it is (i.e., there exists a dynamic reservation entry for the frame's MAC/VID) then the frame is discarded from (or never queued in) that queue; if not, then handling the frame is essentially the same as for case 2/4 below.

For case 1/4, which amounts to valid stream data for a currently active stream being queued on the wrong outbound Port, the ideal behaviour would be for the frames associated with the mis-configured stream not to be placed in that queue for transmission. However, some, but not all, of the sub-cases can be detected from available FDB data:

1. If the mis-configuration is that a dynamic reservation entry in the FDB erroneously identifies this Port as "forwarding" for the VID/MAC address concerned, then it is not possible to use FDB information to detect that this is actually a mis-configuration; it is essentially no different from the case of a Talker transmitting at a higher data rate than allowed by its reservation for the stream.
2. If the mis-configuration is that there is no dynamic reservation entry in the FDB for the VID/MAC address concerned, then this case could be detected by interrogating the FDB, and action taken accordingly. However, there are practical considerations that would affect the ability of existing implementations to do this – see "implementation considerations" below.
3. If the mis-configuration is that there is some other entry in the FDB for the VID/MAC address concerned, for example, a static filtering entry, or a dynamic filtering entry created by the

[1] The no-brainer would be to discard the frames; however, there is a perfectly reasonable argument that if you re-map the priority of legacy frames coming into an SRP domain, then re-mapping the priority of legacy frames generated within an SRP domain is OK too, as long as you are confident that they **are** legacy frames. The advantage is that you potentially break fewer legacy protocols that way.

operation of MMRP, then this case could be detected by interrogating the FDB, and action taken accordingly[2]. However, again, there are practical considerations that would affect the ability of existing implementations to do this – see "implementation considerations" below.

For case 2/4, which happens when "legacy" non-stream traffic that uses SR class priorities is generated within the SRP domain boundary, the detection possibilities are closely similar (but subtly different) to those outlined above for case 1/4. It is conceivable that a dynamic reservation entry could be present in the FDB for the VID/MAC address concerned, either because the legacy app is using a stream destination address, or because the FDB has been mis-configured; again, this is indistinguishable from the case of a Talker exceeding its bandwidth allocation for a stream, and is not detectable using available FDB information. If the frame is being forwarded on that Port because either there is no FDB entry, or there is an FDB entry (other than a dynamic reservation entry) that indicates "forwarding", then this case is detectable using currently available FDB information. The question would then be whether the appropriate action would be to re-map the priority associated with the frame in order to ensure that it does not use a shaper queue, or to not queue the frame on that Port. However, again, there are practical considerations that would affect the ability of existing implementations to do this – see "implementation considerations" below.

Implementation considerations

With the exception of handling the stream frame/zero idle slope case, which is trivial, the degree to which the solutions outlined above can be achieved will depend on the way that "real" forwarding tables are implemented in products. As a generalization, the filtering database (FDB) structure that is described in Clause 8 of 802.1Q really only describes the management database in the Bridge; the forwarding table (FT) that appears in the "fast path" of a Bridge implementation is a very much simpler structure. The reason for this is that accessing the full FDB structure in the fast path of the Bridge would be impractical at wire speeds; the FT therefore contains a condensed version of the FDB information.

In its simplest form, the FT is a database that is indexed using a tuple of VID (or FID) and MAC address, and if there is a "hit", returns a Port map indicating forward or filter for each Port. There will be a "hit" if there is **any** explicit data in the FDB for that tuple; this could be any combination of:

- A static filtering entry;
- A dynamic filtering entry;
- A MAC address registration entry; or
- A dynamic reservation entry.

The information contained in the set of entries in the FDB for a given tuple is distilled into the Port map, indicating forward or filter for each Port, that is held in the FT. The consequence of this is that while the FT still tells you whether a frame should be forwarded on a given Port, it doesn't tell you whether this is because there is a reservation on that Port for that tuple, or because there is some other FDB entry that specifies forwarding for that tuple. So interrogating the FT can tell you:

- Whether or not there is a "hit" for the tuple;
- If there is a hit for the tuple, whether the frame should be forwarded or filtered on a given Port, but not why (i.e., does not tell you what type(s) of FDB entry said "forward").

Given the simple FT structure outlined above, the only real possibility for dealing with any aspects of the harder cases (1/4 and 2/4 above) is, if the priority associated with a frame corresponds to one of the SR classes supported on that Port, and there is no FT "hit" (i.e., there are no FDB entries for the VID/MAC address tuple), then the frame is not queued on that Port. Unfortunately, this only deals with the case where the frame

[2] Note that there could be a dynamic reservation entry as well; the effect of these entries is additive according to current 802.1Q.

would otherwise be "flooded" because there is no FDB hit; it doesn't deal with any of the other FDB mis-configuration cases.

To deal more effectively with the other cases, it would be necessary for the FT to store more information; a simple flag associated with each FT entry, indicating whether or not there is a dynamic reservation entry in the FDB for that tuple, would allow a better job to be done. The logic then could be:

IF
<priority == stream priority>
AND <FT-hit for VID/MAC tuple>
AND <port-map indicates Forward for this Port>
AND <FT flag indicates dynamic reservation entry exists for this tuple>
THEN
<queue the frame on the relevant shaper queue on this Port>
ELSE IF
<priority != stream priority>
AND <FT-hit>
AND <port-map indicates Forward for this Port>
THEN
<queue the frame on the relevant non-shaper queue on this Port>
ELSE IF
<priority != stream priority>
AND <No FT-hit>
AND <defaults indicate forward unknown addresses >
THEN
<queue the frame on the relevant non-shaper queue on this Port>
ELSE
<do not queue the frame on this Port>

The above logic discards legacy frames if there is no reservation on the Port and the legacy frame uses a stream priority. An alternative logic, which re-maps the priorities of legacy frames that use stream priorities, looks like this:

IF
<priority == stream priority>
AND <FT-hit for VID/MAC tuple>
AND <port-map indicates Forward for this Port>
AND <FT flag indicates dynamic reservation entry exists for this tuple>
THEN
<queue the frame on the relevant shaper queue on this Port>
ELSE IF
<priority == stream priority>
AND <FT-hit for VID/MAC tuple>
AND <port-map indicates Forward for this Port>
AND <FT flag indicates dynamic reservation entry does not exist for this tuple>
THEN
<re-map the priority and queue the frame on the relevant non-shaper queue on this Port>
ELSE IF
<priority != stream priority>
AND <FT-hit>
AND <port-map indicates Forward for this Port>
THEN
<queue the frame on the relevant non-shaper queue on this Port>
ELSE IF
<priority != stream priority>
AND <No FT-hit>
AND <defaults indicate forward unknown addresses >
THEN
<queue the frame on the relevant non-shaper queue on this Port>
ELSE
<do not queue the frame on this Port>

As mentioned earlier, the FDB mis-configuration cases include the possibility that there is a dynamic reservation entry as well as one or more other types of entry; as the effect of these entries is currently additive, we may wish to consider changing the specification in the FDB such that a dynamic filtering entry overrides any other type of FDB entry for a given tuple. This would remove the possibility of streams leaking out of Ports where they have not been provisioned, but where the VID/MAC address is specified as Forwarding because there is some other type of FDB entry in play.

What can be fixed now?

Clearly, if we decide not to re-map legacy frames, dealing with the zero idleSlope case is an easy fix, and could be implemented in current Q-REV. Similarly, it would be relatively simple to specify and implement logic of the form "if it is a stream priority and there is no FT hit, then don't queue the frame on a shaper queue" (which would address part of the unknown flooding case), so this could also be done in current Q-REV.
We might also consider changing the way the FDB is specified such that stream reservation entries override any other FDB entries for that tuple, rather than entries always being additive as they are now.

What should be fixed later?

Doing a more complete job would likely involve some combination of the following:

- Refining the "If no FT hit then…" approach to "If no dynamic reservation entry then…" and changing the combining rules for FDB entries such that the dynamic reservation entry overrides anything else in the FDB;
- Policing what source address(es) can be valid sources for a given stream, and discarding the stream if the SA is incorrect;
- Policing actual bandwidth used vs. bandwidth reserved, potentially on a stream-by-stream basis, and limiting actual bandwidth accordingly;
- Probably lots of other policing stuff that I haven't considered.

However, dealing with any of these items is likely to involve significantly more thought and development than is practical at this stage of the Q-REV project, so should be held over until we start work on version 2 of the AVB standards.