DRAFT DISPOSITION OF BALLOT COMMENTS on P802.1p/D4: Draft Standard for Local and Metropolitan Area Networks - Supplement to Media Access Control (MAC) Bridges: Traffic Class Expediting and Dynamic Multicast Filtering

Sponsor

LAN MAN Standards Committee
of the
IEEE Computer Society

Prepared by:

Tony Jeffree, Project Editor

11A Poplar Grove
Sale, Cheshire
M33 3AX, UK
Tel: +44-161-282-3824
Fax: +44-161-973-6534
Email: 100271.522@compuserve.com

Commentary:

This Draft Disposition of Ballot Comments has been prepared to document the ballot comments received in the ballot on P802.1p/D4, and to prepare the issues for discussion at the editing meeting. The document contains:

1) A table of responses received.
2) A listing of comments received, by author, each accompanied by a proposed disposition or a reference to the list of issues;

Note that for Editorial comments, the default disposition is that the Editor will take the comments into account in preparation of the next Draft. Explicit statements of how editorial comments have been resolved are only included for exceptions to this default.

3) A list of issues that were raised in ballot comments, and which require a solution to be agreed at the editing meeting.

This document, once agreed and completed, will constitute a record of the Instructions to the Editor for the preparation of P802.1p/D5.

<<Until agreed by the Editing Meeting, any resolutions or proposed resolutions contained in this document are simply the opinion of the Editor; the contents of those resolutions is therefore subject to change.>>

CONTENTS

1. Table of Responses

The following table indicates the status of each ballot response received on the P802.1p/D3 ballot. Where comments have been received without an accompanying ballot, this is indicated by a reference only in the COMMENTS column.

Table 1—P802.1p Ballot Analysis

NAME	STATUS	YES	NO	ABSTAIN			COMMENTS
				(Time)	(Exp.)	(Other)	(Section, Page)
Floyd Backes	Voting			X			-
John Boal	Voting			X			-
Paul Carroll	Voting			X			-
Jeff Catlin	Voting			X			-
Alan Chambers	Voting		NO				S22, P51
Steve Chan	Voting	X					-
Hon Wah Chin	Voting					X	-
Steve Cooper	Voting		X				S8, P16
David Delaney	Voting	X					-
Peter Ecclesine	Voting					X	-
JJ Ekstrom	Voting				X		-
Norm Finn	Voting		X				S13, P22
Rick Foster	Nonvoting			X			-
Paul Frantz	Voting		X				S19, P34
John Grinham	Voting		X				S24, P65
Steve Haddock	Voting		X				S7, P15
John Hart	Voting			X			-
Richard Hausman	Voting		X				S11, P20
Ariel Hendel	Voting					X	-
Steve Horowitz	Voting		X				S5, P10
Raj Jain	Nonvoting			X			-
Vipin Jain	Nonvoting						S26, P69

Table 1—P802.1p Ballot Analysis

NAME	STATUS	YES	NO	ABSTAIN			COMMENTS
				(Time)	(Exp.)	(Other)	(Section, Page)
Tony Jeffree	Voting		X				S23, P58
Hal Keen	Voting		X				S18, P31
Keith Klamm	Voting		X				S9, P17
Paul Lachapelle	Voting			X			-
Gadi Lahat	Nonvoting		X				S6, P14
Paul Langille	Voting			X			-
Johann Lindmeyr	Voting			X			-
Peter Martini	Voting			X			-
John Messenger	Liaison		X				S15, P26
Milan Merhar	Voting	X					-
Yaron Nachman	Voting		X				S2, P7
Krishna Narayanaswami	Voting			X			-
Joerg Ottensmeyer	Voting		X				S3, P8
Anand Padmanabhan	Nonvoting					X	-
Luc Pariseau	Voting			X			-
Yondav Perry	Voting			X			-
Gideon Prat	Voting					X	-
Kirk Preiss	Liaison			X			-
Anil Rijsinghani	Voting		X				S4, P9
Doug Ruby	Voting				X		
Ayman Sayed	Voting				X		-
Mick Seaman	Voting		X				S20, P38
Rich Seifert	Liaison			X			-
Lee Sendelbach	Voting		X				S14, P25
Himanshu Shah	Voting			X			-
Kark Shimada	Voting	X					-

Table 1—P802.1p Ballot Analysis

NAME	STATUS	YES	NO	ABSTAIN			COMMENTS
				(Time)	(Exp.)	(Other)	(Section, Page)
C. Fred Shu	Voting			X			-
PJ Singh	Voting			X			-
Stuart Soloway	Voting		X				S12, P21
Robin Tasker	Voting		X				S16, P27
Don van-Mierop	Voting			X			-
John Wakerly	Voting		X				S10, P19
Peter Wang	Voting		X				S21, P47
Trevor Warwick	Liaison		X				S17, P30
Alan Weissberger	Potential voting			X			-
Mike Witkowski	Voting			X			-
Edward Wong	Potential voting			X			-
Michael Wright	Voting			X			-
Robert Wu	Nonvoting	X					-
John Yang	Voting			X			
Wayne Zakowski	Voting		X				S25, P66
TOTALS (Voters & Liaison only)	***60 voting members of 802.1 (at start of ballot)***	***3***	***23***	***21***	***3***	***4***	
PERCENTAGES (Voters & Liaisons only)	***90% response (Yes, No or Abs)***	***12%***	***88%***				

2. Yaron Nachman

Comment 1

I think that priority tagging should be kept separated from the VLAN tagging. User might very much use priority on Ethernet without VLANs. So, we should give each "feature" a different "VLAN Ethertype" - VLAN tagged only, Priority tagged only, VLAN & Priority tagged.

Disposition of Comment 1

See Issue 1.

Comment 2

I still don't see the use of more than two levels of priority. I think that it could be also the conclusion out of Fouad Tobagi presentation in the Twente meeting.

Disposition of Comment 2

See Issue 1.

3. Joerg Ottensmeyer

Comment 3

The priority fields need to be taken out of the VLAN header In the presentation of Norm Finn on Service Classes, both solutions (priority signaling via GARP and in the VLAN header) turned out to be equal solutions. However, the VLAN approach has some mayor drawbacks:

a) priority mechanisms can only be implemented together with VLANs. If no VLANs are needed no priorities can be supported.
b) the priority of frame can not change on its way though a network. The main benefit of this feature would be that priorities could be overwritten at bridges (by network management procedures) if needed.

Disposition of Comment 3

See Issue 1.

Comment 4

The GARP mechanism as currently described in D4 is by far too complex. I propose to use the IGMP mechanism for IP multicast as a guide. Moreover, since IP multicast will be the major application of MAC multicast mechanism, the interworking of these has to studied more thoroughly than currently outlined in annex E2.

Disposition of Comment 4

See Issue 2.

4. Anil Rijsinghani

Comment 5

Mechanisms in the GARP protocol appear quite a bit more complex than similar functionality for IP as documented in RFC1112. A number of simplifications may be worth evaluating. In describing some of these to Tony, it appears that some simplifications are indeed being considered and will to be presented to the WG.

Disposition of Comment 5

See Issue 2.

5. Steve Horowitz

Comment 6

This is really coming together. Tony, you have a real talent for pulling this complex work together. ('Course, I don't know how you are going to resolve this comment.)

Disposition of Comment 6

That’s easy Steve...I agree!

Comment 7

(2.6.6.2: 50) What is the motivation for having Mode C be the default port filtering mode? That mode would not be backward compatible with existing stations. It would seem to me that Mode B or even Mode A would make the most sense for default (factory) configuration for plug N play reasons and compatibility with existing equipment that is not GARP aware. General: What is the motivation for having GARP be able to change the mode? This also is not clear with regards to its practical application.

Disposition of Comment 7

See Issue 3.

Comment 8

(2.6.9.4) I don't understand the value of changing the priority of the frame depending on the outbound port. I am reading this section as: a frame arrives high priority and is forwarded out a port that essentially ignores the priority and sends it as normal priority. If I am reading this correctly, then I don't understand how this applies in a real network or how it would be used. Hmmmm, just thinking. Are you trying to support bridges where some ports have priority queueing and some don't? Then, do you really want to change the frame priority or just how it is queued?

Disposition of Comment 8

See Issue 1.

Comment 9

(3.7.5 - Note) I think that it suffices to state the all high priority traffic is forwarded before low priority traffic, but allowing for vendor differentiation for other options. That is, I agree with the way it is currently written.

Disposition of Comment 9

See Issue 4.

Comment 10

(3.7.3:30) It is not clear to me what exactly is optional and what exactly is mandatory. There is also some related chart in one of your annexes (A?), but I am not clear on how to read it. Can the key for the chart be made more explicit? I read this line as that I must support the capability to set the Outbound User Priority per port. Is this true?

Disposition of Comment 10

No. There is a choice (signalled by the word “May”) of two options:

a) Provide management capability;
b) Don’t provide management capability.

If you choose a), then it is mandatory (signalled by the word “Shall”) to allow independent management of this parameter per Port, and for the full range of values...etc. to be settable per Port.

If you choose b), then it is mandatory that the implementation uses the defaults specified in the table.

Yes, the Annex A tables are tough to read, but that’s the standard format. No substitute for learning the notation, Steve.

Comment 11

(3.9.3.1:44) I assume entries learned by GARP are considered "dynamic". If this is the case, then dynamic entries are also created and updated by the GARP process (or the learning process section needs to be updated to include GARP).

Disposition of Comment 11

No...GARP is responsible for generating some dynamic entries, the Learning Process is responsible for generating others. GARP has nothing to do with learning unicast addresses.

Comment 12

(3.9.3.2: 27) This may be a cut and paste error - Did you really mean to include individual and broadcast addresses on this line? It appears that in other places that you specifically exclude them. This paragraph needs work.

Disposition of Comment 12

You’re right. Will get fixed in next draft to line up with bullets a) through e); i.e., delete the second sentence of the paragraph.

Comment 13

Style nit: Some tables are specified with one dash, and others with two dashes. Table 3-5 vs. Table 9--1.

Disposition of Comment 13

Will fix it.

Comment 14

(6.3 h) - microseconds or centiseconds? Your later tables are in centiseconds. What is SNMP's time interval (10ths of seconds)? Isn't that more convenient? (centiseconds is OK, micro seems <word I should not use in public>).

Disposition of Comment 14

Should be centiseconds. Will be fixed in next draft.

Comment 15

(9.5.1.4: 22) I still don't understand why the timer is restarted if you see another Leave. Doesn't this cause the time for a segment to be de-registered to be linear to the number of GARP switches on that segment? Since leave time is 3 seconds, then the time that a segment de-registers with 3 switches can be up to ~13 seconds (assuming your 1.5 multiplier). I had a similar comment (I think) on the previous draft. Why isn't it good enough just to ignore subsequent leave requests for the same group?

Disposition of Comment 15

See Issue 2.

Comment 16

Style nit: State machine pictures did not print very well. States don't fit in the circles.

Disposition of Comment 16

Will get fixed.

Comment 17

Clarity note: Why not just say "sendJoin" and "receiveJoin" (or rcvJoin) instead of "sJoin" and "rJoin", etc.? It would be clearer and does not require the reader to search for the legend. Re: State machine diagrams section needs some work.... Off the top of my head (and not being an IP multicast expert):

1) IP Router sends join to IGMP well-known address (as do the IP end-stations hanging off of mode C ports).
2) IP Router determines which IP multicast sessions it is serving and sends GARP message joining each one, followed by IGMP advertisement.
3) End-stations must join any group desired, followed by IGMP report for the router.

Isn't it something like that?

Disposition of Comment 17

See Issue 5.

Comment 18

Nit: F.2: Running a GARP state machine for every port for every group is not a lot different than running a STP for every VLAN...The difference being is that the GARP state machines don't do a lot when in a steady state, only when changes are occurring, where STP does require regular calculations for "best port". The steady state is what saves it.

Disposition of Comment 18

Er...not clear what you want me to do with this, Steve...

Comment 19

Issue 5 (AZ): Define what is manageable, but let IETF define the MIB. (my vote). We define the instrumentation necessary, IETF defines the behavior and structure of the managed objects. If GDMO is necessary, make is a separate work item so as not to hold this effort up.

Disposition of Comment 19

See Issue 6.

Comment 20

H-11 comment - Why not differentiate between a join from a bridge and a join from an end-station? For all states except less anxious, the rJoin and the rbJoin (bridge join), are treated the same. The rbJoin is ignored if in the Less Anxious state. Does this fix the problem? (Note, this is off the top of my head without any real thought - perhaps a conversation over beer at Ottawa will help.)

Disposition of Comment 20

See Issue 2.

Comment 21

Figure 9-7. Transition from leaveImminent State to out State. Should read:

"leavetimer expired: perform Member Left Action"

Which timer expired is inconsistent with notation on fig9-6. And if it expired, there is no need to stop it.

Disposition of Comment 21

Will be fixed in next draft.

Comment 22

9.6 rLeave description should say: receive Leave or LeaveAll PDU.

Disposition of Comment 22

Will be fixed in next draft.

Comment 23

9.9.1.4 Forwarding delay --- Is this the same as the Bridge Forward Delay as defined in 802.1D Table 4.3? If so, a reference is needed.

Disposition of Comment 23

Yes. Add reference as proposed.

6. Gadi Lahat

Comment 24

I think that priority tagging should be kept separated from the VLAN tagging. User might very much use priority on Ethernet without VLANs. So, we should give each "feature" a different "VLAN Ethertype" - VLAN tagged only, Priority tagged only, VLAN & Priority tagged.

Disposition of Comment 24

See Issue 1.

Comment 25

I still don't see the use of more than two levels of priority. I think that it could be also the conclusion out of Fouad Tobagi presentation in the Twente meeting.

Disposition of Comment 25

See Issue 1.

7. Steve Haddock

Comment 26

Section 3.7.4: If there is no recommendation on the number of traffic classes supported in a.1p bridge, then it leaves an implied suggestion to take the all or none approach: either don't support expedited traffic classes or implement eight classes. The advantage of a recommendation is that it explicitly acknowledges that there is value to supporting more that one traffic class even if the number supported is less than eight. I don't think this is controversial. The controversy comes in when trying to pick a specific number to recommend. Therefore I suggest a compromise of changing the recommendation from "two" to "at least two."

Disposition of Comment 26

See Issue 7.

Comment 27

Section 3.7.5: The strict priority scheduling algorithm is not the most appropriate for all applications, however I suspect it will be very difficult to find any one algorithm that is appropriate for all applications. I concur with the spirit of the footnote (not the main text) that it is useful to have one consistent mode of operation for predictable and interoperable behavior, while still permitting implementation-specific variations. I suggest that the standard specify the strict priority algorithm as a default scheduling algorithm to be supported by all.1p bridges, but acknowledge that bridges may be administratively configured to support other (unspecified) algorithms.

Disposition of Comment 27

See Issue 4.

8. Steve Cooper

Comment 28

The GARP mechanism is too complex.

Disposition of Comment 28

See Issue 2.

Comment 29

There have been a number of studies which indicate that priorities should be able to be changed within the "mechanism". A farfetched example, is that you might not be real concerned with an incoming missile when it is far away and not locked on but may change your mind as it gets a bit closer.

Disposition of Comment 29

See Issue 1.

Comment 30

Section 3.7.4: I support a compromise of changing the recommendation from "two" to "at least two levels of priority." While I believe that two is inadequate, there seems to be great difficulty in selecting the right number. The compromise might help.

Disposition of Comment 30

See Issue 7.

Comment 31

Section 3.7.5: The standard should specify the strict priority algorithm as the default scheduling algorithm to be supported by all.1p bridges, but acknowledge that bridges may be configured to support other algorithms.

Disposition of Comment 31

See Issue 4.

9. Keith Klamm

My vote to disapprove this ballot is based in part on the many open issues and on the following comments:

Comment 32

Page 23, line 25; Concerning traffic class number recommendation: Recommendations in standards tend to set both the minimum and maximum implementation compliance levels. Since this is not what we are trying to do here I recommend that this standard make no recommendations regarding the number of traffic class levels. Annex F adequately explains the rationale behind multi-level traffic classes and could be referenced here to guide implementors.

Disposition of Comment 32

See Issue 7.

Comment 33

Page 24, line 43; Concerning de-queueing algorithms: If we intend to allow implementor to select their own de-queueing algorithms (pg. 24, line 38) then there is not much point in providing a standard algorithm. I recommend making a very general statement in section 3.7.5 along the lines of:

"frames queued on higher priority queues are serviced in preference to frames queued on lower priority queues but not necessarily to the exclusion of frames queued on lower priority queues".

(Actually the existing material in 3.7.5 is acceptable.) This whole issue is best left to product differentiation.

Disposition of Comment 33

See Issue 4.

Comment 34

Page 29, line 27; strike "individual addresses" and "and the broadcast address" from this sentence.

Disposition of Comment 34

Amend the text as specified.

Comment 35

Page 32, line 37; replace "microseconds" with "centiseconds".

Disposition of Comment 35

Amend the text as specified.

Comment 36

Page 34, line 7 and page 35, line 30; Consider this comment a nit. I don't follow the reasoning behind having the "Set Port Inbound Priority Handling To Default" and the "Set Bridge Filtering Mode to Default" commands output information that is already accessible via the "Read" command. Can these commands be simplified by making their output "none"?

Disposition of Comment 36

See Issue 6.

Comment 37

Page 41, line 3; Concerning bridge management: Ultimately the IETF will define the official SNMP MIB for managing 802.1p bridges. However what could be of real use to the IETF effort is a central definition of the objects that this IEEE working group feels are appropriate for 802.1p bridge management. I recommend that we define a pseudo-MIB (i.e. written in SNMP ASN.1) that would serve as a model for future MIB work.

Disposition of Comment 37

See Issue 6.

Comment 38

Page 75, line 49; Concerning fixed timer values: Fixed timer values are allright with me- it is one less thing to go wrong.

Disposition of Comment 38

See Issue 8.

10. John Wakerly

It's getting close...

Comment 39

p. 17. It seems to me that we may need additional priority management/policing capabilities. For example, the ability to "clamp" the priority coming in (or going out) on a given port at a certain value. Perhaps even on a per-port-per-VLAN basis. This could be conveniently stored in a matrix which contained other per-port-per-VLAN information (such as tagging method -- implicit or explicit). So, does this belong in P or Q or both?

Disposition of Comment 39

See Issue 6.

Comment 40

p. 24, Table 3-2. Which is the "highest" priority, 0 or 7? Even though this is defined in Sec. 3.7.5, the table would be a convenient place to reiterate which queue gets serviced first. (E.g., add a column titled "Urgency" and place the word "least" in row 0 and "greatest" in row 7.)

Disposition of Comment 40

See Issue 1.

Comment 41

p. 24, lines 43ff. The dequeueing algorithm need to be settled. I would readily agree with using pure priority as a default, and having the selection of other algorithms being an implementation management option (as long as pure priority or SOME agreed-upon baseline was a selectable option). Among these comments, my most overriding concern is for the ability to manage priority on a per-port-per-VLAN basis. Which standard (P or Q) should specify this capability or, if different pieces of it are in different standards, how do we make them consistent? The answer to my question may be as simple as saying that the priority configuration mechanisms remain as they are in this draft, and that any finer grained per-port-per-VLAN has precedence when implemented.

Disposition of Comment 41

See Issue 4.

11. Richard Hausman

While I've not had time for much review, I note the following:

Comment 42

This cannot become a standard with a forward reference to 802.1Q (regarding priority signalling), unless 802.1Q has first become a standard.

Disposition of Comment 42

See Issue 1.

Comment 43

The discussion regarding the dequeueing algorithm must be resolved (3.7.5). I suggest the strict priority scheme be specified as optional (perhaps recommended), and a hook be left allowing implementations to do otherwise in a visible manner. In particular, have a read-only bit available through the management interface which indicates whether the bridge supports the strict priority (value = 0) or an alternate, implementation specific scheme (value = 1). The idea is, a system seeking to utilize priorities can be designed to examine bridges to determine if the expected algorithm is available, if that is an important question for its operation.

Disposition of Comment 43

See Issue 4.

12. Stuart Soloway

Comment 44

I discussed with Tony my concern that the GARP applicant and registrar protocols and state machines were more complex than necessary. Tony said that a newer, more correct, and simpler version would be forthcoming, and, low and behold, one was presented at the meeting. I have not had a chance to study it, but will do so soon. At that point, I will probably drop my objection for the next draft, but if not I will make a contribution outlining possible improvements.

***Disposition of Comment 44*5**

See Issue 2.

13. Norm Finn

13.1 Technical comments

Comment 45

MINOR. I think that somewhere in parallel to section 2.6 there should be a section that describes the priority tagging feature (mentioned in section 2.3.9 as a service provided by 802.1p). The only service mentioned in 2.6 is the multicast MAC filtering service.

Disposition of Comment 45

See Issue 9.

Comment 46

MAJOR. Section 3.7.3 lines 22-25. This paragraph does not specify what happens if the user_priority in the 802.1p tag differs from the user_priority in the M-UNITDATA.indication. I would suggest that the 802.1p tag overrides the M-UNITDATA.indication, and M-UNITDATA. indication be used only if no 802.1p tag is present. ISSLL can use 8 levels of end-to-end (in MAC/bridge layer terms) priority that does not change as a frame is forwarded. These 8 priority levels may or may not match the physical priorities on media that carry them.

Disposition of Comment 46

See Issue 1.

Comment 47

MAJOR. Section 3.7.4 (2nd page thereof) lines 25-38. Two are sufficient for many purposes. Fouad Tobagi’s excellent presentation makes it clear that more than two levels of priority are pointless when sending data to a CSMA/CD medium. However, more than two levels of priority can be quite useful when point-to-point full-duplex links are employed. 802.1p should say nothing about the number of priority levels that one should implement.

Disposition of Comment 47

See Issue 7.

Comment 48

MAJOR. It would be easier for a two-queue switch to use one bit of the 802.1p priority to make its decisions. Perhaps the table would be more easily decoded if it were built from the center out:

Output queue selection:

user priority	Available classes 2 3 4 5 6 7 8	
		(Principle: decode most-significant
0	0 0 0 0 0 0 0	bits of priority field. 2 queues
1	0 0 0 0 0 0 1	use high bit, 4 queues high 2 bits.
2	0 0 1 1 1 1 2	On intermediate boundaries, differ-
3*	0 0 1 1 2 2 3	initiate priorities closest to 3/4
4	1 1 2 2 3 3 4	boundary, the "center" of the table,
5	1 1 2 3 4 4 5	before differentiating the ends.
6	1 2 3 4 5 5 6	Differentiate high priority levels
7	1 2 3 4 5 6 7	before low priority levels.)

* This gets rid of the notion that 0 == "unspecified." There would be no such thing as a packet with a 802.1p tag that makes no choice of priority. If an 802.1p/Q tag has no preference for priority, it uses "3" as the priority.

Disposition of Comment 48

See Issue 1.

Comment 49

MAJOR. Section 3.7.6. I believe that we have a problem, here. Were we bridging between like media, and were the 802.1p tag required to match the M_UNITDATA priority, then simply setting the M_UNITDATA.request priority to the bridge-determined user_priority would work. Unfortunately, this is not the case. I think the only answer is a per-medium (802.5, 802.6,...) set of tables mapping 802.1p priority to M_UNITDATA priority. For example, on a medium with only two levels of priority, the table would be:

802.1p priority:	0 1 2 3 4 5 6 7
M_UNITDATA.request priority:	0 0 0 0 1 1 1 1

Disposition of Comment 49

See Issue 1.

Comment 50

MAJOR. This all leads to an algorithm: An 802.1p-compliant bridge carries the 802.1p priority through the bridging process. On input, if an 802.1p tag is present, that is the priority carried through. If not, then yet another table is used to map the M_UNITDATA.indication to 802.1p priority. For example, for a two-level medium, the input table would be:

M_UNITDATA.indication priority:	0 1
802.1p priority:	3 4

On the output side, the table in point 6, above, maps the 802.1p priority to queue selection. The appropriate table in point 7 translates from 802.1p priority to M_UNITDATA.request. This is rather complex, but the decision that 802.1p priority has global (to the bridged [V]LAN) meaning seems to drive us toward this model. 802.1D dodges the question of mapping priorities between media. This faces the problem.

Disposition of Comment 50

See Issue 1.

Comment 51

(These comments on revising the priority tables are suggestions, not demands), I'm open to arguments as to why the current tables are better. My main concern is allowing ISSLL to use all 8 levels of priority in the 802.1pQ tag.

Disposition of Comment 51

See Issue 1.

13.2 Editorial comments

Comment 52

a) Typo in section 3.7.3, line 22. c) should be a)?
b) Is this document formatted for A4 paper? The page numbers at the bottom of the document are cut off by my printer. If the document could be formatted for the minimum width (A4) and minimum length (11") of each format, then everyone could print it.

Disposition of Comment 52

a) IEEE formatting rules indicate that numbered bullets are sequentially numbered within a subclause, in order to disambiguate references to them. This is, in any case, in the (broad) category of things that get fixed between the Editor & the IEEE Standards Office prior to publication.
b) The Editor will attempt to fix the problem.

14. Lee Sendelbach

Comment 53

Standard is obviously not ready for prime time with all of the questions and open issues as detailed in the rear of the standard.

Disposition of Comment 53

True.

Comment 54

Issue 3 - traffic classes - I don't think the standard should limit the classes to only two at this time. This makes a lot of assumptions about future applications. I would restrain it to approximately 8 levels to be compatible with FDDI/Token-Ring.

Disposition of Comment 54

See Issue 7.

Comment 55

Issue 5 - I would opt to let the IETF define the SNMP MIB. If 802.1 were to define it, the IETF would choose a smaller subset of what 802.1 had done anyway, thus rendering a certain percentage of our work meaningless.

Disposition of Comment 55

See Issue 6.

Comment 56

Issue 8 - Centiseconds seems too fine to me. We use tenths of a second on our products and it seems to be adequate. I do not think the standard should mandate this due to the potentially large overhead required to get that much accuracy. There may be some parameters which need this accuracy, but the vast majority should not need it.

Disposition of Comment 56

See Issue 8.

15. John Messenger

Both of these comments are primarily issues to do with VLANs, but at the Ottawa Interim it was decided that priority issues were in fact 802.1p issues instead, so here they are.

Comment 57

Where a MAC provides a method of signalling the User Priority across a link, then that method should be used to convey the priority when the frame is transmitted over that MAC. There may be a mapping to translate the tag priority to the User Priority. In the case of 802.5, it should be a one-to-one mapping, with the 3-bit priority field in the tag being transferred directly to the 3-bit YYY field in the Frame Control octet (the 3 rightmost bits of the FC). Doing this allows existing bridges to honor the priority designation.

Disposition of Comment 57

See Issue 1.

Comment 58

Similarly, there should be a mapping to translate the tag priority into an Access Priority when transmitting on a MAC that supports access priorities. The appropriate mapping to choose in this case is less trivial. For 802.5, perhaps that recommended for use when forwarding 802.5 frames in a 802.1d bridge is appropriate (0...4 -> 4; 5->5; 6->6; 7->7, I believe).

Disposition of Comment 58

See Issue 1.

16. Robin Tasker

16.1 Technical comments

Comment 59

802.1p/D4 is incomplete in a number of areas noted by the editor and until these have been resolved the draft cannot progress.

Disposition of Comment 59

This is true.

Comment 60

p11 2.3.9 (b) "by use of the user priority field carried in the VLAN header" I am unhappy that two distinct services - VLANs and priority - are being tied together by this statement. They should stand separately in their formal definitions. This is tied to my equivalent comment to 802.1Q - both services are distinct extensions to what currently exists, and there may well be others to be added in the future. So rather than tying priority to VLANs just have (the existing) field that currently separately defines VLANs and priority. In this way one can define one without the other and if required add new services in the future.

Disposition of Comment 60

See Issue 1.

Comment 61

p12 2.6 (a)(b)(c) Will there be a conformance clause associated with this Section and if so which items will be included.

Disposition of Comment 61

There are already conformance clauses in p/D4 related to GARP conformance and management conformance.

Comment 62

p22 line 16 "Both these delays can be managed...." This is not true for Ethernet so either don't say it or make it more precise.

Disposition of Comment 62

Replace the offending text as follows:

“Queueing delays can be managed using user_priority. Access delays can be managed using user_priority in media access methods that support more than one access priority.”

Comment 63

p22 lines 40 - 42 I would support the Editor's Note in D4

Disposition of Comment 63

OK

Comment 64

p22 lines 53 onwards This paragraph still sounds too much like an implementation.

Disposition of Comment 64

In the absence of proposed replacement text, it's likely to stay that way.

Comment 65

p23 Number of Traffic Classes I think it's unreasonable for this Standard to recommend 2 traffic classes. The text describes the user priority levels and how they are mapped to 802.1p priorities; it should be an implementation matter as to how that is then modeled into number of queues.

Disposition of Comment 65

See Issue 7.

Comment 66

p24 Note 2 This Note should probably be removed. It is vital that the Standard defines a de-queuing algorithm which is the default operation on all bridges claiming conformance. One of the purposes of a Standard is to allow inter-operating products to be manufactured. To leave the Note as currently worded appears to encourage precisely the opposite.

Disposition of Comment 66

See Issue 4.

Comment 67

p27 line 9 and 28 "by local or private means" At the very least some re-ordering of text is required, e.g. "carried out by use of the remote management capability provided by Bridge management, or by local or private means" but better still just remove the words.

Disposition of Comment 67

Remove the words “by local or private means, or”.

Comment 68

p28 and 29 bullet items Just for consistency change the item notations (c) and (d) to (a) and (b); and, (e) and (f) to (a) and (b) and (d) - (g) to (a) - (d); and, (h) - (k) to (a) - (d)

Disposition of Comment 68

Consistency with what? Bullet numbering is not a fruitful topic for ballot comments - it is one of many boring issues that are resolved between the Editor & the standards office prior to publication, in accordance with the formatting rules prevailing at the time. As a matter of fact, those rules currently disallow multiple bullet sequences within a single (sub)clause, of the kind proposed by this comment, as it is then not possible to unambiguously reference a bullet item. (For example, if we were to adopt the proposal in this comment, a reference to “3.9.3.2 a)” would be ambiguous.) In the context of that rule, I think you will find that the text is entirely consistent.

Comment 69

Section 7 I have no text to offer but suggest that (d) is the best option. In reality until 802.1p is technically stable MO definitions are difficult but a PAR to express the intent is probably worthwhile.

Disposition of Comment 69

See Issue 6.

Comment 70

p45 line 20 I assume "at any moment in time" refers to the Very Anxious state entered just after last orders have been called and before the Landlord requests drinking up. How about "at any instant" to avoid this worrying misunderstanding.

Disposition of Comment 70

No...it refers to the Very Anxious State entered just after the Editor reads the Email header that says “Ballot comments on P802.1p/D4 from R Tasker”.

OK - I give in - change wording to “at any instant”.

Comment 71

p92 Annex G I still believe Goal (a) and Non-Goal (a) are mutually exclusive. Clearly "to satisfy QoS requirements of time-critical" is very different from the "provision of guaranteed QoS levels"; but is the former a whole lot of use without the latter in providing the services expected from this work.

Disposition of Comment 71

Not clear to the Editor what changes are required to the text in the light of this observation.

16.2 Editorial comments

Comment 72

p44 line 28 "Port MAC Bridges" -> "Ports of MAC Bridges"

p84 line 26 "However, such may" -> "However, such benefit may"

17. Trevor Warwick

Comment 73

I would prefer 802.5 and FDDI to continue to use their native mechanisms for signalling requested priority. 802.5/FDDI devices should do priority queueing based on received FC values only. Devices that bridge between tagged 802.3 and another medium would translate priority from 802.3 into the FC value as part of the tagging/translation process.

Disposition of Comment 73

See Issue 1.

18. Hal Keen

As there are a lot of unfilled gaps and open issues, there can be no question of actually approving this draft. I have therefore neglected to declare particular comments editorial, or designate which ones constitute a basis for disapproval. This omission would not be appropriate in ballots on a draft which is close to completion. Some of these comments are contributions to the discussion on specific open issues; they do not necessarily offer specific changes, and may therefore be disposed of as the committee chooses.

Comment 74

I have noticed enumerations of possible MAC types, with the apparent implication that the lists are exhaustive. Even if these lists cover all cases presently expected, an 802.1 standard should be written to cover all providers of the MAC service, including any future MAC types that may be introduced. The lists on page 11, lines 35-36, and page 22, lines 24-25 should therefore be qualified with "e.g." (replacing "i.e." in the latter instance).

Disposition of Comment 74

Adopt rewording as proposed.

Comment 75

Issue 3 arises from our desire to accommodate as many priority classes as any known MAC service could distinguish, combined with an awareness that we should not mislead the occasional unwary implementor into overly ambitious approaches where it is evident there are diminishing returns. A survey of available material, including the draft Annex F, the 802.5 priority classes, and Gideon Prat's material presented in Wakefield, yields a long list of different traffic classes with distinct QoS requirements. For our purposes, it is more useful to examine the possible handling in a bridge. The following assumes the simple queueing approach, i.e. higher priorities transmitted first. (This is an attempt to extend notions I picked up in a conversation with Mick last June. I'll take all blame, but cannot claim all credit, associated with these ideas.) High-priority traffic with a limited bandwidth can be presumed not to overwhelm the throughput capacity of the bridge at any time. (If this is not so, more capacity is needed!) It therefore suffers negligible queueing delay; whenever a high-priority frame is to be forwarded, the previous one should already have been transmitted.

"High priority" in this context refers to any priority classes such that the maximum aggregate received burst never results in any queueing delays for any of the traffic in those classes. That being the case, all such traffic may be regarded as constituting a single priority class.

Lower classes of traffic are subjected to queueing delays. If it is desirable to guarantee a lower latency to some of this queued traffic, further class divisions may be introduced. (Bounding latency is also our only effective means, in a MAC service provider, for limiting jitter.) The lower the class, the greater the delay. When traffic bursts overwhelm capacity, and frames are discarded, the same lower classes which suffer the greatest delay also suffer the greatest rate of discard. This is not necessarily bad; many protocols reduce the transmission rate to compensate for data loss, increasing the effectiveness of the overall use of available capacity. Again, if the lowest traffic class is completely shut out, acquisition of greater capacity is the indicated remedy.

The division into two classes allows one class with minimal delay and discard probability effectively zero, and another which may be subjected to both delay and discards. Each additional division is either meaningless (because it simply adds another high-priority class, indistinguishable from the first) or it allocates the delay and discard probability unequally among different portions of the lower class. This is a process yielding diminishing returns; eventually, it will be more effective, much more predictable, and simpler, to purchase increased capacity than to add more traffic classes in the bridge.

We need to make two things clear to implementors:

a) the fundamental separation of traffic into two categories,
   1) a single, high-priority class which suffers negligible delay and negligible discard rate, subject to limits in burst rate and total bandwidth consumed, and
   2) a continuum of lower-priority classes, such that any preference (lower delay and lower probability of discard) obtained by one class is at the expense of another;
b) the diminishing returns involved in adding subdivisions within category 2) (I should note that if the limits on category 1) are violated, that category, or some portion of it, becomes part of category 2). The point of category 1) is that it contains the portion of the traffic for which we don't want that to happen. If it does, it's definitely time to shop for higher capacity.)

I was happy with recommending two classes, as the default implementation strategy for anyone who didn't know how many classes they needed, just so long as we didn't make it look like a requirement. I thought we had achieved that, before the discussion I missed in Twente.

Disposition of Comment 75

See Issue 7.

Comment 76

With respect to Issue 4, I see little need for alternatives to the simple rule of highest priority transmitted first. In particular, there seems to be no reason to complain that some lower-priority frames may be discarded. In the absence of sufficient capacity to forward all traffic, that's the desired solution. (See previous comment.) In fact, existing single-class bridges do the same thing; we're simply adding a means of more intelligently selecting suitable frames for discard.

Disposition of Comment 76

See Issue 4.

Comment 77

Would lines 37 and 40 on page 42 read better with "in consequent" changed to "consequent"?

Disposition of Comment 77

Change wording as proposed.

Comment 78

The multicast router problem discussed in E.3.1 (pages 86-87) is very similar to a problem that can occur with GARP bridges. The router is operating Applicant state machines triggered by requests from its surroundings. In the bridges, this problem is addressed by periodically transmitting Leave All (as the router does, once, at startup). I'm not sure it works--sorry, I'm short of time to work out details--but could a periodic Leave All approach be adapted to the router case?

Disposition of Comment 78

See Issue 1.

Comment 79

The network monitor situation discussed in E.3.2 (page 87) does not seem entirely real. In ordinary Bridged LAN environments, filtering limits the propagation of individually addressed frames; I don't expect to be able to attach a monitor to an arbitrary LAN segment and pick up all the unicast traffic. GARP merely extends the same filtering characteristics to multicast. If network monitors routinely force Mode A filtering, that could seriously disrupt, or at least distort, the normal functioning of the network as long as the monitor is running. This elegant replication of the Heisenberg Uncertainty Principle notwithstanding, some sort of health warning belongs in this section.

Disposition of Comment 79

Add a suitable health warming.

Comment 80

19. Paul Frantz

19.1 Technical comments

Comment 81

General - To deal with the problem of referring to 802.1Q which does not yet exist, the priority handling section should only refer to the priority associated with the received MAC data indication. An informational note can indicate that if other ways of associating a priority with a frame are standardized in the future, e.g. the proposed 802.1Q standard, those priority values may be used when no priority information is available in the MAC data indication.

Disposition of Comment 81

See Issue 1.

Comment 82

Section 2.6.8.1 - Why is there any reference to unicast in 802.1p? It does not seem necessary.

Disposition of Comment 82

See Issue 9.

Comment 83

Section 2.6.9.2. - The ability to specify ranges of MAC addresses should not have been removed. It doesn't matter as far as the data path is concerned (a range is equivalent to a whole bunch of individual entries) but it does matter for management. If you want to say that all addresses in the range used by IP multicast require registration, and then a management process tries to read out the table of group entries, with the current draft text the switch would transmit to the management process an entry for every address in the range. This is not practical.

Disposition of Comment 83

See Issue 6.

Comment 84

Section 3.7.4 - there is no need to recommend a specific number of traffic classes. The draft text is fine. Whether the extra cost of additional classes is justified by the potential for improved performance is surely an issue for vendors to decide.

Disposition of Comment 84

See Issue 7.

Comment 85

Section 3.7.5 - De-queueing algorithms are best left up to vendors. What the specification should say is that for any packet traversing the switch, the latency and probability of loss when the packet is sent at a higher priority shall be no worse and may be better than would be observed if the packet were sent at a lower priority with all other conditions being equal.

Disposition of Comment 85

See Issue 4.

Comment 86

Section 3.7.7 We should not preclude implementations which appropriately adjust the received CRC to account for the changes made to the packet rather than recomputing the entire CRC. Vendors may choose to do this to provide error detection on the switch's internal data paths.

Disposition of Comment 86

See Issue 10.

Comment 87

Section 3.9.3.2, lines 27, 31 - references to individual addresses should be removed.

Disposition of Comment 87

Editor to remove these references.

Comment 88

Section 3.9.4, figure 3-8 and related areas - the specification should not specify redundant data structures. If a static filter is configured for a multicast MAC address, this appears as a static filtering entry. If Joins are received, this entry is converted to a group registration entry. But when the dynamic membership ages out or is deleted as a result of Leaves being processed, even though the data being held is exactly the same as before the first Join was received, it's still considered a group registration entry, not a static filtering entry as before. Either the entry should revert back to a static filtering entry, or filtering entries should apply only to unicast addresses with group registration entries used for all multicast control. This may seem like a documentation nit but it's not - it changes the information which will be provided over the management interface.

Disposition of Comment 88

See Issue 11.

Comment 89

Section 3.9.4 item e) is not needed - learning does not apply to broadcast or multicast addresses.

Disposition of Comment 89

Remove bullet e).

Comment 90

Figure 3-8 - transition from bottom right to bottom left is possible if management deletes all static elements.

Disposition of Comment 90

Not true. According to the rules as stated, this can only happen if the entire entry is deleted (an entry that has any static elements has a statically defined address - there is no means of removing that element without removing the entry in total). That transition is already described in the diagram.

Comment 91

Section 6 needs completion. Also, on line 8, "Use GARP Priority" no longer applies.

Disposition of Comment 91

Remove “Use GARP Priority”.

Comment 92

6.6.5.2 - The ability to change the port traffic class table should not be required for switches which implement our recommended default table. At most we should say that if a vendor chooses to make the table changeable, then they must use our standardized mechanism for communicating the change to the switch. There should be a means for management to retrieve entries for only those groups on which active dynamic entries exist (same problem as above with having to walk through the entire IP multicast address space).

Disposition of Comment 92

See Issue 6.

Comment 93

Section 7. A MIB is required. The best solution is to get IETF to create it. If they won't, 802.1 needs to do it.

Disposition of Comment 93

See Issue 6.

Comment 94

Annex F fails to draw a distinction between delay-sensitive and loss-sensitive traffic. These are very different requirements. They should be discussed individually.

Disposition of Comment 94

See Issue 12.

19.2 Editorial comments

Comment 95

(Minor) - typo, section 2.6.9.3, line 15

(Suggestion) - I second the suggestion made on the e-mail list that the document should be formatted so that it will print on either A4 or 8.5x11 paper without cutting off page numbers. Not all PDF readers/printers provide easy rescaling.

20. Mick Seaman

20.1 Technical comments

Comment 96

Replace the protocol definition of GARP with a simplified form as proposed in Ottawa. See 20.2 for text of "Simplifying GARP".

Disposition of Comment 96

See Issue 2.

Comment 97

Remove the handling of legacy equipment, currently embedded in the GARP protocol, to an instance of GARP which makes its own separate registration for "there are legacy devices here" just as it would register for a group, plus the set of forwarding rules which take into account such registrations.

Disposition of Comment 97

See Issue 13.

20.2 Simplifying GARP

20.2.1 Introduction

This note proposes and explains a revision to the GARP protocol being standardized in 802.1p. The revised protocol is simpler than the protocol of P802.1p/D3 [1] in four senses:

a) It is explainable in a common sense, easy to understand way (or so the author hopes!).
b) The timer value dependencies of [1], which require join timers to expire in the range 0.5 to 1.0 of their maximum value, and which make it a matter of protocol correctness that all join timer maximum values be (very close to) the same, have been removed.
c) Conflicts between the Applicant and Registrar state machines have been removed by having a single state machine send all required messages. In [1], when another participant sends a leave, an Applicant sends a join message in response, then its own Registrar sends a further leave, which finally forces the Applicant to send a further join (if no other participant does). The revised protocol uses a much simplified Registrar.
d) Implementation of the state machines for a GARP participant or Group can be reduced to very few bits while still permitting simple processing: two bits for the simplest end station.

In a sense the minimum number of recorded bits provides a hard metric of simplicity. Nine or ten suffices for the most complex case of a bridge port (potentially both receiving from and transmitting to the Group members), inclusive of:

a) Administrative controls:
    1) Always Join (whatever the user instructs)
    2) Never Join
    3) Always In (tell user there are group members)
    4) Never In (always tell user there are no group members)
    5) Do Not Participate (send no messages)

b) Timer support (!).

c) Pending transmission handling to support:

   1) packing of transmit messages for multiple groups into a single PDU for protocol efficiency (important because the most important factor in protocol reliability in general is input buffer overrun, not loss or data corruption on the wire)

   2) rate limiting of PDU transmission, through a holding time as for 802.1D

   3) the exception case of no transmission buffer available - which has to be handled in real implementations.

d) Support for the total suppression of join and leave messages in cases where they are unnecessary (see below).

The revised protocol provides two important enhancements:

e) It fixes the delayed join after single message loss bug identified by Paul Frantz in his 802.1p/D3 ballot comments.

f) If sufficient join messages have already been sent, and noted by a participant, he can join the group without sending a further join, and, if he never sends a join at all can subsequently leave quietly as well. This reduces the scope for protocol storms brought on by topology fluctuations, an important scaling consideration.

The purpose of this note is to explain the proposed revised protocol, not to describe the design or verification process, nor to provide an intellectual basis for further tinkering. The design of even trivial protocols is subject to many errors and does not proceed by plausible assertion. While the title of this note stresses simplicity, correctness is paramount - protocols should be as simple as possible, but no simpler than necessary. Once we have one sound implementation to cut and paste perceived complexity is unlikely to bother us again.

The actual design process proceeded by way of stepwise refinement from [1], exploring a few blind alleys along the way, and generating many temporary analyses. Acknowledgments and thanks for comment and help along the way are due to Vipin Jain of 3Coms TDC, Tony Jeffree, and Dave Cheriton, though they may not recognize or approve the result. Responsibility for remaining bugs and flaws is mine.

20.2.2 Basic Notions

The basic notions behind GARP, as a simple but efficient protocol, are that:

a) A simple fully distributed many to many protocol is possible. There is no need for an additional election protocol to change the problem to allow a many to one design.

b) The protocol should be resilient against the loss of a single message, in a set of related messages, but does not need to be stronger.

c) A GARP participant that wishes to join a Group (a would be Member, or Applicant) sends Join messages.

d) If an Applicant sees other participants sending two Join messages, it does not need to send a Join itself.

e) A Group Member that wishes to leave the Group need only send a single Leave message, it can then forget all about the Group. There is no need to confirm departure from the group.

f) Missing or spuriously continued memberships that arise from multiple lost messages are cleared up by a periodic mechanism which throws all the members out and forces rejoining.

The proposed revised protocol preserves these original ideas.

To guard against the possibility that a group member misses a Leave message, - thus causing another participants Registrar (the component that records membership of the Group) to think that there are no members -

one additional mechanism is necessary. If a participant receives a Leave message, and no subsequent joins, it sends a further message to prompt rejoining.

20.2.3 GARP Messages

802.1p/D3 [1] is very frugal with the message types it uses, trying to make do with only three types: Join, Leave, and LeaveAll for the periodic garbage collection. With hindsight it appears that the attempt to retain simplicity by choosing to live with so few message types added to the eventual complexity of [1]:

a) Consider the case of two GARP participants at either end of a point to point link. The fact that one of them (Andy, say) sends the other (Bill) two Join messages says nothing about Andys knowledge of Bills wish to be a member of the Group. The recitation of the basic GARP notions above obscures the fact that would be members may also need to know if there are other members - a requirement for bridge ports for example. [1] attempts to work around the problem by setting join timers such that no single participant can send two messages in an interval within which two or more participants can be expected to send messages. This is the cause of the odd (0.5-1.0 of maximum) timer specification, and the timer correctness dependencies. The scenario described is easily envisaged on a point to point link but is equally applicable to a multi-access segment. It is hard to say for certain that such an interval is well defined if there are participants leaving and joining continuously. Depending on timers for correctness is a pact with the devil.
b) Consider the sending of a second Leave by a Registrar to prompt a rejoin. If a Join is just being sent the protocol now depends on the second Join not being lost. The protocol depends on the relative values of the Registrars leave timer and other participants join timers.

The revised protocol is based on the general design principle that protocol participants should communicate their current state, rather than send directions. Unless bandwidth is so precious that even a single bit is expensive this is usually a better strategy. Four group specific message types are used:

a) Empty: I am not trying to join this group. I have not registered the existence of any members, but I care if there are any.
b) JoinEmpty: I wish to be a member of this group. I have not registered the existence of any other members, and I care if there are any.
c) JoinIn: I wish to be a member of this group. I have either registered other members or I don’t care if there are any (I will behave as if there are).
d) Leave: I was a member of this group, but am now leaving.

along with the garbage collection message, as before:

e) LeaveAll: Everybody (all the members of all the groups) have been thrown out, if any of them want to join they need to rejoin.

In theory there could be LeaveIn and LeaveEmpty variants of the Leave message, and I suggest coding them in PDUs to provide maximum visibility into what implementations are doing, and to avoid missing or illegal codes. However it will be seen that the state machines treat these two message flavors identically.

I can find no reason to send a simple In message, i.e. one that means I do not wish to join this Group but have registered the existence of other members (or will behave as if there are other members).

The revised protocol makes good use of the distinction between JoinEmpty and JoinIn messages, and between Leave and Empty.

The JoinIn message meets the requirements for Join message suppression. If an Applicant sees a JoinIn messages it can indeed avoiding sending a Join itself, for now it knows that both the recipients and the transmitter of the JoinIn believe there are group members. The JoinIn is not treated as an acknowledgment, because, on a multi-access segment, there are potentially many Registrars which need register the group. Moreover, participants which don't care whether there are other members or not can always send JoinIns instead of JoinEmptys. However on the assumption that only one JoinIn message is lost, two suffice to ensure that all Registrars have registered the group - to a satisfactory high probability.

The Leave message will cause its recipients to un-registered membership, while the JoinEmpty and Empty messages will just prompt them to rejoin, so can be used at any time to prompt for rejoin without throwing recently joined members out again.

20.2.4 An Informal Protocol Description

In the following description, the term GARP participant refers to that part of a system responsible for operating the GARP protocol on a LAN segment, possibly one of many to which the system is attached. Typically an end station would have a single GARP participant, while a bridge would have one per port. Propagation of information between participants in a system follows Spanning Tree connectivity.

Applicant and Registrar

Each GARP participant maintains two protocol components per Group that it is interested in - an Applicant and a Registrar.

The job of the Registrar is to record group membership of the other participants on the segment. It does not send any protocol messages.

The job of the Applicant is twofold:

a) To ensure that this participant is registered as a member by other participants registrars - if it wants to be a member of the group.
b) To ensure that other participants have a chance to rejoin the group, after anyone leaves - if there are any that want to be members.

The major difference from [1] is that the Applicant is looking after the interests of all would be members. This allows the Registrar to be very simple.

Registrar behavior

The Registrar has a single timer, the leave timer, and three states:

a) IN: I have registered the existence of other members of this Group on this segment.
b) MT: (Empty) There are no other members on this segment.
c) LV: I had registered members, but am now timing them out (using the leave timer). After they are timed out I will become MT.

The Registrar reacts to received messages as follows:

a) A Join message of either flavor JoinIn or JoinEmpty causes the Registrar to become IN (I have registered the existence of other members).
b) If Registrar was IN, then a Leave or LeaveAll causes it to become MT (I am timing out other members) and starts the leave timer. Otherwise (LV or MT) there is no effect.
c) An Empty message (someone else has no registered members) has no effect. If this is not obvious consider Andy and Bill above.

While the Registrar does not send messages, it affects the type of Join message sent by the Applicant. If the Registrar is IN, a JoinIn is sent, otherwise a JoinEmpty is sent.

Against the background of this simple Registrar let us consider the behavior of the Applicant of a would be member, starting from a point where it has neither seen or sent any messages.

Applicant behavior

If no messages were ever lost, the Applicant could either send a Join or receive a JoinIn, and then be content that all Registrars would have recorded its membership. On the single message loss assumption it needs to send two Joins, or receive two JoinIns, or send one Join and receive one JoinIn (in either order). This part of its state could be recorded in a simple counter:

my_membership_msgs = 0,1, or 2

which is incremented for every Join sent or JoinIn received. Above 2 we don't care what the counter value is, so it can be assumed to stick at 2 for the purposes of implementation. If the counter value is 0 or 1 a Join message will be sent and the counter incremented when there is an opportunity to transmit a PDU.

Note: A randomized join timer is set running to ensure such an opportunity is scheduled. There only needs to be one join timer running for the entire participant, not one per group - assuming that messages from the maximum number of groups can be packed in a single PDU.

If a JoinEmpty, Empty, Leave, or LeaveAll message is received the counter is reset to 0.

When the Applicant leaves the Group it sends a single Leave message.

A simple participant

The simplest possible GARP participant is one that only wishes to join groups, has no need even to take note of other members of those groups i.e. is not transmitting to the groups or at least is not attempting to apply pruning at source to its transmissions, makes no attempt to suppress its initial Join and final Leave messages, and offers no additional administrative controls.

Its behavior can be summarized simply:

```
enum my_membership_msgs = 0 /*(0,1,2)*/
...
req_join{...
my_membership_msgs = 0;
...
}
...
rcv_msg{...
if (msg == joinin)
if (my_membership_msgs <2)
my_membership_msgs++;
else
my_membership_msgs = 0;
...
}
...
tx_opportunity{...
if (my_membership_msgs <2)
tx(joinin); my_membership_msgs++;
...
}
```

```
...
req_leave{...
tx(leave);
...
}
```

Anxious Applicants

Expressing protocol behavior in terms of counter and flag variables is not always the best approach if enabling thorough analysis and maximizing implementation flexibility are primary goals. From this point on, the values assigned to the membership message count are given state name prefixes:

a) V or Very anxious equates to my_membership_msgs = 0. No Join messages have been sent and no JoinIns received since the Applicant started or leave or empty messages received. The Applicant has no reason to be comfortable that other Registrars have registered membership for the Group.
b) A or Anxious equates to my_membership_msgs = 1. If no messages have been lost other Registrars will have registered group membership.
c) Q or Quiet equates to my_membership_msgs = 2. The Applicant feels no need to send further messages.

If this seems more complex than 0,1,2 you can turn these state prefixes back into a counter for your implementation.

Members and Observers

The simple participant described above has no existence unless he is trying to join the Group. Bridge ports and end stations which implement source pruning for transmission need to maintain their GARP machines even if they do not want to be members, or have just left. (A GARP machine is just the name for the total state which represent the Group in a participant).

In the context of the Applicant state machine, a Member is a participant that is attempting to join or maintain membership of the group, or who has not yet sent the Leave message to allow him to become simply an Observer. An Observer tracks the Group but does not wish to join.

Active and Passive Members

The concept of Active and Passive Members is introduced to permit the minimum number of messages to be sent when a number of participants are actively joining and leaving the same group.

Since a Member may become Quiet without ever sending a Join it follows that he should be allowed to become an Observer once more without sending a Leave. All Observers are passive of course, so we have three potential (sub)states, distinguished by the following state name suffixes:

a) A, or Active member.
b) P, or Passive member.
c) O, or Observer.

If an Observer is required to become a member it first becomes a Passive member, requesting - if it is not already Quiet, content that all other Registrars have registered Group membership - the earliest possible message transmission opportunity.

If a Passive member sends a Join message he becomes an Active member.

If an Active member receives a Leave or LeaveAll message he becomes Passive.

Receiving a Leave

When an Applicant that is, and wishes to continue being a Member receives a Leave Message, it becomes Very anxious. Unless it receives a Join message from another participant it will send a JoinEmpty itself. This will have two effects on other participants. First, it will cause them to reregister members for the Group. Second, it will cause them to become Very anxious themselves if they wish to be members, and to transmit JoinIns.

This latter effect protects any participant that is a member from accidentally deregistering other members due to a single packet loss following a Leave.

An Applicant that is an Observer has to prompt other members to rejoin in case they have missed the Leave. A further (sub) state is added to the Very Anxious, Anxious, Quiet set with state name prefix:

d) L or Leaving, which records the pending need to send a message at the next transmission opportunity. An Observer will send an Empty message, and then become Very Anxious.

Leaving the Group

An Active Member has to send a Leave message before leaving the Group. The Leaving substate is used to record that fact.

Applicant State Summary

The following matrix summarizes the applicant states and their short names - VA for Very anxious Active member, QO for Quiet Observer etc.

Table 1—Applicant - summary state table

	Very Anxious	Anxious	Quiet	Leaving
Active Member	VA	AA	QA	LA
Passive Member	VP	AP	QP	
Observer	VO	AO	QO	LO

Note that there is no LP (Leaving Passive member) state, since a Passive member can transition directly to an Observer state when he wishes to leave the Group.

20.2.5 The Applicant state machine

While the above description does specify the protocol, and a simple implementation will separate {Leaving, Very Anxious, Anxious, Quiet} from {Observer, Passive member, Active member} here is the full Applicant state machine.

Note: sJ[E,I] means send JoinEmpty if Registrar State Machine is in either MT or LV, and JoinIn if Registrar State Machine is in IN.

Table 2—Full Applicant state table

	VA	AA	QA	LA	VP	AP	QP	VO	AO	QO	LO
transmitPDU!	sJ[E,I] AA	sJ[E,I] QA	-x-	sL VO	sJ[E,I] AA	sJ[E,I] QA	-x-	-x-	-x-	-x-	sE VO
rJoinIn	AA	QA	QA	LA	AP	QP	QP	AO	QO	QO	AO
rJoinEmpty	VA	VA	VA	LA	VP	VP	VP	VO	VO	VO	VO
rEmpty	VA	VA	VA	LA	VP	VP	VP	VO	VO	VO	VO
rLeave, rLeaveAll	VP	VP	VP	LA	VP	VP	VP	LO	LO	LO	VO
ReqJoin	-x-	-x-	-x-	VA	-x-	-x-	-x-	VP	AP	QP	VP
ReqLeave	LA	LA	LA	-x-	VO	AO	QO	-x-	-x-	-x-	-x-

20.2.6 Combined state machine

While the separate Registrar and Applicant state machines form the best basis for design, it is also instructive to examine the combined state machine, to check for unreachable states, and to search for different ways of factoring the problem into smaller machines. The following matrix shows all the reachable states, with cells containing the joint state names, Applicant.Registrar, and unreachable states marked ---. The MT and LV Registrar states are grouped together since the only event which differentiates the two is the expiry of the leave timer, which does not affect any of the other states.

Table 3—Combined states

	Very Anxious		Anxious		Quiet		Leaving	
Active Member	VA.MT VA.LV	VA.IN	AA.MT AA.LV	AA.IN	QA.MT QA.LV	QA.IN	LA.MT LA.LV	LA.IN
Passive Member	VP.MT VP.LV	VP.IN	--- ---	AP.IN	---	QP.IN		
Observer	VO.MT VO.LV	VO.IN	--- ---	AO.IN	---	QO.IN	LO.MT LO.LV	---

There are 24 reachable states in all.

The combined state machine follows. For compactness the obvious - what message is transmitted, when the implementation should check or start timers, when to indicate joins and leaves to the higher layer user - is omitted.

Table 4—Combined state machine

	Leave Timer!	Transmit PDU!	rJoinIn	rJoin Empty	rEmpty	rLeave, rLeaveAll	ReqJoin	ReqLeave
VA.MT	-x-	AA.MT	AA.IN	VA.IN	VA.MT	VP.MT	-x-	LA.MT
VA.LV	VA.MT	AA.LV	AA.IN	VA.IN	VA.LV	VP.LV	-x-	LA.LV
VA.IN	-x-	AA.IN	AA.IN	VA.IN	VA.IN	VP.LV	-x-	LA.IN
AA.MT	-x-	QA.MT	QA.IN	VA.IN	VA.MT	VP.MT	-x-	LA.MT
AA.LV	AA.MT	QA.LV	QA.IN	VA.IN	VA.LV	VP.LV	-x-	LA.LV
AA.IN	-x-	QA.IN	QA.IN	VA.IN	VA.IN	VP.LV	-x-	LA.IN
QA.MT	-x-	-	QA.IN	VA.IN	VA.MT	VP.MT	-x-	LA.MT
QA.LV	QA.MT	-	QA.IN	VA.IN	VA.LV	VP.LV	-x-	LA.LV
QA.IN	-x-	-	QA.IN	VA.IN	VA.IN	VP.LV	-x-	LA.IN
LA.MT	-x-	VO.MT	LA.IN	LA.IN	LA.MT	LA.MT	VA.MT	-x-
LA.LV	LA.MT	VO.LV	LA.IN	LA.IN	LA.LV	LA.LV	VA.LV	-x-
LA.IN	-x-	VO.IN	LA.IN	LA.IN	LA.IN	LA.LV	VA.IN	-x-
VP.MT	-x-	AA.MT	AP.IN	VP.IN	VP.MT	VP.MT	-x-	VO.MT
VP.LV	VP.MT	AA.LV	AP.IN	VP.IN	VP.LV	VP.LV	-x-	VO.LV
VP.IN	-x-	AA.IN	AP.IN	VP.IN	VP.IN	VP.LV	-x-	VO.IN
AP.IN	-x-	QA.IN	QP.IN	VP.IN	VP.IN	VP.LV	-x-	AO.IN
QP.IN	-x-	-	QP.IN	VP.IN	VP.IN	VP.LV	-x-	QO.IN
VO.MT	-x-	-	AO.IN	VO.IN	VO.MT	LO.MT	VP.MT	-x-
VO.LV	VO.MT	-	AO.IN	VO.IN	VO.LV	LO.LV	VP.MT	-x-
VO.IN	-x-	-	AO.IN	VO.IN	VO.IN	LO.LV	VP.IN	-x-
AO.IN	-x-	-	QO.IN	VO.IN	VO.IN	LO.LV	AP.IN	-x-
QO.IN	-x-	-	QO.IN	VO.IN	VO.IN	LO.LV	QP.IN	-x-
LO.MT	-x-	VO.MT	AO.IN	VO.IN	VO.MT	LO.MT	VP.MT	-x-
LO.LV	LO.MT	VO.LV	AO.IN	VO.IN	VO.IN	LO.LV	VP.LV	-x-

21. Peter Wang

21.1 Technical comments

Comment 98

Section 2.6.9.3: Last paragraph of this section implies that the PORT_MODES parameter is always defined regardless of BRIDGE_MODES value in the MANAGER_DEFINE _DEFAULT_FILTERING_MODE primitive. Yet the description following the primitive implies otherwise.

Disposition of Comment 98

Make it clear that the PORT_MODES parameter is significant only for Mode 2.

Comment 99

Figure 3-7: Add arrow from DESTINATION PORT STATE INFORMATION to the bubble 3.7.4. This is for the traffic class lookup while queuing a frame.

Disposition of Comment 99

Change the diagram as proposed.

Comment 100

Section 3.7.4:

a) Should specify the ordinal of the user priority and traffic class, i.e. 7 is the highest priority, to avoid any possibility of misinterpretation.
b) In the paragraph which reference Table 3-2, instead of "recommend', state the use of Table 3-2 as THE default mapping. Then add reference to managed object in Clause 6 indicating that the default can be changed. This is necessary since the appropriate mapping is application dependent and will be affected by how ISSLL is resolved, both of which will take time. In the mean time, we need a common default in order to start from a common baseline with predictable behavior.
c) The recommendation for 2 traffic classes as the default for expedited service should be in the text somewhere. If not normative text, then as a note. There are many good arguments for 2 traffic classes, some which we've heard from Fouad's presentation and discussions on ISSLL. This approach doesn't prevent vendors from implementing more than 2 traffic classes, but will accommodate a large percentage of the users with lower cost switches.

Disposition of Comment 100

a) Add a specification as proposed.
b) Change the text as proposed.
c) See Issue 7.

Comment 101

Section 3.7.5: State that the pure priority-based dequeueing algorithm is the default. Other dequeueing scheme can be installed via specific management action. Argument for taking this approach is similar to that of the traffic class mapping stated above.

Disposition of Comment 101

See Issue 4.

Comment 102

Section 3.7.6: Bullet c) appears to be redundant, since bullet a) subsumes it.

Disposition of Comment 102

Not true. The regenerated user_priority is not necessarily the same as the incoming (signalled) priority.

Comment 103

Section 3.9.3.1: Current description seem to have a hole in term of the dynamic filtering entry, the static filtering entry and the learning process. Section 2.6.8.2 implies that the PORT_MAP of a static filtering entry specifies the set of ports on which a unicast address is allowed. In this case, the learning process needs to establish the actual port on which the source unicast MAC address reside. However, Sec. 3.8, bullet c) seem to disallow this.

Disposition of Comment 103

That is correct. This text is technically identical to existing 802.1D (all that has changed is the references and entry names). In current D, as specified by bullet 3.8 c), the Learning Process does not learn the location of addresses for which a static entry exists.

Comment 104

Section 3.9.3.2: Is only the static part of the Group Registration Entries stored in the Permanent Database?

Disposition of Comment 104

Yes - make this clear in the text.

Comment 105

Section 6.6.3.4.3: Output parameters are not necessary, since the READ operation can be used to verify the SET result.

Disposition of Comment 105

See Issue 6.

Comment 106

Section 6.6.4.1.3: Suggest adding the value of "UNDEFINED" for the Port Filtering Mode and remove the conditional on Bridge Filtering Mode value.

Disposition of Comment 106

Amend as suggested.

Comment 107

Section 6.6.4.2.2: Need to allow setting of Port Filtering Modes as well.

Disposition of Comment 107

Add Port Filtering Mode parameter.

Comment 108

Section 6.6.4.3.2: Suggest adding the value of "UNDEFINED" for the Port Filtering Mode and remove the conditional on Bridge Filtering Mode value.

Disposition of Comment 108

Amend as suggested.

Comment 109

Section 6.6.4.4.3: Output parameters are not necessary, since the READ operation can be used to verify the SET result.

Disposition of Comment 109

See Issue 6.

Comment 110

Section 6.9: Allow setting of GARP timer values?

Disposition of Comment 110

See Issue 8.

Comment 111

Section 7: We need a SNMP MIB. I'm not sure if IETF does all of the MIBs? Or if 802 can generate a new MIB?

Disposition of Comment 111

See Issue 6.

21.2 Editorial comments

Comment 112

Section 1.3: Since we've removed the registration of unicast MAC address and the priority, should the definition of 'Group' still mention them?

Section 2.6.6, bullet b): Should 'unregistered' be 'unfiltered'?

Section 2.6.6.2, bullet b): 'registered_for_membership_on_port_set' isn't defined anywhere else.

Section 2.6.8.1: Should the primitive include a 2nd argument of 'port'?

Section 2.6.9.1: Should the ES_REGISTER_GROUP_MEMBER primitive include 2nd argument of 'port'?

Section 2.6.9.4: To be consistent with Sec. 3.7.3, the initial value of the default Forward With Inbound Priority mode should be "TRUE", not "FALSE".

If priority handling configuration is considered part of Extended Filtering service, then there should be a description of Traffic Class configuration (i.e. add Sec. 2.6.9.5).

Section 3.7, 3rd paragraph (i.e. the Note:), 4th line at the end: Should "passing selecting" be "passing selected"?

22. Alan Chambers

Comment 113

Major editorial, mostly, but with some technical

Clause: 2.6

Concern: Appropriateness of the "service" definitions for filtering

Rationale:

I *definitely* agree with what seems to be the intent of 2.6, to provide a definition of what filtering is about and how it behaves at a level above the mere mechanics of hop-by-hop filtering-table decisions. However, I don't think that what is here is in the right place, and I don't think that the service-primitive approach works at all well.

For example, most of 2.6.8 and 2.6.9 add nothing to what is (or soon should be) in section 6, apart from a duplication of terminology and consequent scope for FUD about what is going on: most of the primitives exactly match the management operations, which would be a much better description. Also, the service primitives pull back to a focus on behavior at individual Bridges and Ports, and do not convey a more holistic view of the filtering behavior in the Bridged LAN as a whole.

(On the other hand, the new improved description of Filtering Modes in 2.6.6 is an excellent step or three in the right direction.)

Proposal:

Ah! There are several detailed adds, moves and changes that cover the easy 80% of what I think is needed. The important 20% is much harder, and I have only a couple of top-level suggestions / observations to contribute at this stage: I promise to give it more serious thought, and I hope to have sensible initial proposals for the Vancouver meeting.

Easy bits:-

a) Everything from the start of 2.6 up to the end of 2.6.3 is informative, not normative: it should be reduced to appropriately placed notes in whatever normative text is developed, or moved to one or more informative annexes (current, or new).
b) When we've worked out the hard bits, 2.6.4 and 2.6.5 will fit into place (modified a bit) as useful normative introductory stuff; 2.6.7 will be replaced by equivalent throat-clearing.
c) As it stands, 2.6.8.1 adds nothing useful to the real specifications in 3.7, 3.8 and 3.9, and should be deleted.
d) 2.6.8.2 describes purely management functionality: it should be deleted, with any necessary detailed semantics moved to clause 6 (6.7.5.1 and 6.7.5.2).
e) Similarly, 2.6.9.2, 2.6.9.3 and 2.6.9.4 describe purely management functionality and should also be deleted; detailed semantics that are not already in clause 6 should be moved there (to 6.7.5.5/6.7.5.6, 6.6.4, 6.6.3 respectively).
f) The service primitives in 2.6.9.1 do not add any real explanation of what happens when group registration / de-registration are invoked: the second sentences of 9.5.1.1 and 9.5.1.3 can be deleted, along with 2.6.9.1. (The useful list of unGARPable addresses is already present in 9.6.1, so does not need to be moved to safety.)

Hard bits:-

g) It seems that the more holistic view of filtering behavior across the whole Bridged LAN, suggested above, probably belongs in clause 3, under Principles of Operation. There seems to be an analogy here with what we found when doing 802.1G, where it was necessary to extend clause 3 from its original (802.1D) single-bridge focus in order to discuss and specify how certain collections of bridges operated together in order to make the right things happen. A similar extension to deal with the flow of filtering and registration information throughout a Bridged LAN may well be what is needed for 802.1p. I think I can see 2.6.4 - 2.6.6, and a few fragments from elsewhere in 2.6, fitting into such a framework.

h) Also analogously with 802.1G, there is likely to be a need for a fair amount of informative / tutorial material to support what should be a reasonably hard normative piece of clause 3 text on filtering: as in 802.1G, the natural home for such material is an informative annex.

Disposition of Comment 113

See Issue 9.

Comment 114

Major technical (Issue 13, from Ottawa)

Clause: 2.3.9 (b), etc.

Concern: "VLAN frame header"

Rationale and Proposal:

I strongly support the single-ethertype encapsulation, to include priority or VLAN ID, or both. There is no editorial or procedural difficulty in specifying the encapsulation independently but compatibly in both 802.1p and 802.1Q, with in each case the other standard's bits specified as reserved, and perhaps a note about their possible uses.

The concerns about "what will we invent next week that will not fit inside the VLAN encapsulation" are, I believe, unfounded. In support of this belief, I observe that the proposed dual-use encapsulation does not actually add any sudden new functionality to the traditional MAC service. Rather, it

a) provides the possibility of supporting priority -- which has long been a feature of the MAC service -- over bridged / switched LANs containing CSMA/CD segments, and

b) provides scalability by extending the ability to provide the traditional MAC service, as provided in small(ish) LANs, to apply in very large bridged / switched LANs by dividing them up into multiple small(ish) VLANs. On that basis, I would suggest that we ought perhaps to consider renaming the VLAN header as, say, the HLAN header (for Harmonized LAN). (We should certainly rename it in some way, to avoid giving the impression that priority-enabled CSMA/CD, as per 802.1p, requires that VLANs be implemented.)

802.1p therefore needs to include a specification of the "HLAN" encapsulation for use over CSMA/CD LANs.

Disposition of Comment 114

See Issue 1.

Comment 115

Major technical (Issue 12)

Clause: 9

Concern: Simpler, more correct, GARP

Rationale and Proposal:

In principle, I support the revised, simplified and improved version of GARP that Mick presented at the Ottawa meeting. Simpler specs with fewer bugs are what the world needs! Obviously, I shall want to take the time to study and compare before actually voting Yes on this. Presumably a revised annex H, or alternative indication of correctness, etc., is to be expected in due course.

Disposition of Comment 115

See Issue 2.

Comment 116

Minor technical (Issue 8)

Clause: 9.13.2

Concern: Timer resolution

Rationale and Proposal:

The standard should mandate the timer resolution, to avoid confusion among implementors. I assume, from the way the protocol has been developed, that the required precision (or do I mean accuracy?) will be quite low, which is likely to be important in enabling low-cost implementations.

Disposition of Comment 116

See Issue 8.

Comment 117

Minor technical (Issue 7)

Clause: 9.13.1

Concern: Timer values

Rationale and Proposal:

I support the idea of fixed timer values, for the plug-and-play reasons mentioned in the text; provided of course that nobody comes up with a convincing scenario where they do not work.

Disposition of Comment 117

See Issue 8.

Comment 118

Major technical (Issue 5)

Clause: 7

Concern: GDMO / MIB

Rationale and Proposal:

I support possibility (d), and I hereby volunteer to produce the necessary GDMO. (There appears to be little more volume, and no more complexity, than was involved in doing the GDMO additions for 802.1G, and I don't remember them causing me any grief.) Probably not until during/after the Vancouver meeting, though.

Disposition of Comment 118

See Issue 6.

Comment 119

Major technical (Issue 4)

Clause: 3.7.5

Concern: De-queueing algorithm

Rationale and Proposal:

Keep It Simple, Stupid. The simple exhaust-the-higher-traffic-classes approach, as currently specified, is the only one that makes sense in a LAN context. If it doesn't work in practice, it's something else that's broken. (Overloaded LAN, faulty bandwidth allocation / reservation, inappropriate traffic class assignments,...) However, I suppose it would be acceptable to have this as only the (strongly recommended) default, with permission for alternatives to satisfy people who really want to fiddle.

The Notes on page 24 shouldn't be: they both contain normative stuff. Note 2 needs a bit of word-smithing to reinforce the idea that the simple default algorithm is the one to go for, if we do permit alternatives.

Disposition of Comment 119

See Issue 4.

Comment 120

Major technical (Issue 3)

Clause: 3.7.4

Concern: Number of traffic classes

Rationale and Proposal:

Some recommendation is needed, to avoid the risk that everyone will be forced to implement eight classes just in order to get the boxes checked by feature vultures. In the LAN context, queues are typically short because bandwidth utilization is on average comfortably below the level that makes them grow long. This SERIOUSLY reduces the point of having lots of queues. Two classes will allow time-sensitive traffic to cut through peaks of bulk data, as has always been the basic intent; figures / simulations would be nice, but I should be surprised if this didn't cope comfortably with quite high loadings of the upper-class traffic (say 30%+?).

A couple of considerations keep my mind open to the possibility that slightly larger numbers of classes could be useful in some contexts. One reason for going beyond two classes could be to support different loss/discard characteristics for different classes of time-sensitive traffic. This raises interesting issues of coordination / integration with higher-layer functions, since consistent end-to-end mappings of traffic classes would be needed. (For example, I have heard of a real-time video encoding that relies upon a core of transmission that needs to get through with quite high reliability, to give basic picture integrity, with "shells" of additional data, making up most of the maximum bandwidth requirement, that improve picture quality if they get through in time but can be discarded without serious harm. The core service would seem to map obviously to the highest available traffic class, but an intermediate class perhaps with impatient discard timers on the bridge queues might be better for the better-never-than-late characteristics of the rest of the data.)

Another possible reason is to deal with existing LAN protocols that are delay sensitive, but not necessarily to the same extent as, say, real-time voice. It could make sense to put, e.g., LAT or IPX in a traffic class higher than the bulk file transfers, but lower than that for the real real-time traffic.

Disposition of Comment 120

See Issue 7.

Comment 121

Major editorial

Clause: 3.7 - 3.7.3

Concern: Specification of Forwarding / Filtering

Rationale:

These first two and a half pages of 3.7 are not quite there yet. These elaborated forwarding conditions are very difficult to write down accurately in English (802.1G experience again!). Two main problems, it seems. First, the terminology used could be made simpler and more consistent, both within 3.7 and between 3.7 and other subclauses (particularly 3.9). Second, lines 25 and 26 on page 20 are easily overlooked or forgotten by the reader wrestling with pages 21 and 22: the text in 3.7.1 - 3.7.3 needs to reiterate the focus on single (potential) transmission Ports.

Lesser points: the hint about flooding has been lost from 3.7.2 (see 10038 3.7.1 (3)(a)); and page 22 lines 7-16 are out of place in these densely normative subclauses.

Proposal:

The following first re-write is offered as a possible next stage.

3.7 The Forwarding Process

Frames submitted to the Forwarding Process after being received at any given Bridge Port shall be forwarded to the other Bridge Ports subject to the constituent functions of the Forwarding Process. These functions enforce topology restrictions (3.7.1),...<<as in P802.1p/D4>>... and recalculate the FCS if required (3.7.7).

The Forwarding Process functions are specified in 3.7.1-3.7.7 in terms of actions performed for a given frame received on a given Port (termed "the reception Port"). The frame can be forwarded for transmission on some Ports (termed "transmission Ports"), and is discarded without being transmitted at the other Ports.

Note: <<as in P802.1p/D4>>

Figure 3-7 <<as in P802.1p/D4>>

3.7.1 Enforcing topology restriction

Each Port other than the reception Port is selected as a potential transmission Port if, and only if,

a) the reception Port was in a forwarding state (4.4), and
b) the Port considered for transmission is in a forwarding state, and
c) the size of the mac_service_data_unit conveyed by the frame does not exceed the maximum size of mac_service_data_unit supported by the LAN to which the Port considered for transmission is attached.

For each Port not selected as a potential transmission Port the frame shall be discarded.

3.7.2 Filtering frames

For each potential transmission Port selected as at 3.7.1, the frame shall be forwarded or discarded (i.e., filtered), according to the information contained in the Filtering Database taken together with the Bridge Filtering Mode and the Port Filtering Mode for the potential transmission Port, as follows.

For all Bridge and Port Filtering Modes: the frame shall be discarded for a given potential transmission Port if either

a) the frame's destination MAC address is an individual MAC address and the Filtering Database contains a Dynamic Filtering Entry (3.9.3.1) for that MAC address, with a Port number different from the Port Number of the potential transmission Port; or
b) the Filtering Database contains a Static Filtering Entry (3.9.3.1) for the frame's destination MAC address, with a Port map that specifies filtering of frames for the potential transmission Port.

For Bridge Filtering Mode 2 and Port Filtering Mode B at a given potential transmission Port, the frame shall also be discarded for that Port if:

c) the frame's destination MAC address is a group MAC address and the Filtering Database contains a Group Registration Entry (3.9.3.2) for that MAC address, with a member_port_set that does not include the potential transmission Port.

For Bridge Filtering Mode 2 an Port Filtering Mode C at a given potential transmission Port, the frame shall also be discarded for that Port if:

d) the frame's destination MAC address is a group MAC address, and the Filtering Database does not contain a Group Registration Entry (3.9.3.2) containing both that MAC address and a member_port_set that includes the potential transmission Port.

Each potential transmission Port for which the frame is not discarded as at (a)-(d) above is selected as a transmission Port, to which the frame shall be forwarded.

Note -- In particular, if the received destination MAC address is not in the Filtering Database, the frame is forwarded (flooded) on all Ports that are in a forwarding state, apart from the reception Port and any Ports in Port Filtering Mode C.

3.7.3 Regenerating user priority

For each transmission Port to which a frame is forwarded (3.7.2), the user_priority is regenerated using information contained in the incoming M-UNITDATA.indication primitive, the Forward With Inbound Priority parameter of the reception Port, and the Outbound User Priority of the transmission Port, as follows:

a) If the Forward With Inbound Priority parameter for the reception Port has the value TRUE, and the user_priority of the corresponding M-UNITDATA.indication was specified,...

<< rest as per P802.1p/D4 except delete "transmission", line 31, and "inbound", line 36 >>

Disposition of Comment 121

See Issue 10.

Comment 122

Major editorial

Clause: 3.9 - 3.9.4

Concern: Specification of the Filtering Database

Rationale:

Again, this is not quite there yet, editorially. I think the technical content is sound, but I found the text very confusing. It is VERY hard to see what is static, what is dynamic, what the significant differences are, and when things can change from being one thing to being another.

Proposal:

I have not yet got a complete proposal for improvement, but I will do one before the Vancouver meeting (I hope well before). A few thoughts:

Why not just "Group Entry" instead of "Group Registration Entry", not least since the entry does much more than just the registration?

Then, why not "Dynamic Group Entry" for the GARP-created kind and "Static Group Entry" for the rest (I can happily live with the idea of dynamic elements inside, without complicating the naming scheme)?

Page 29, line 29: cut-and-paste error, "Dynamic Filtering Entries" should be "Group [Registration] Entries".

Page 29, lines 37-38: no point in trying to allow permitted_port_set to be absent, so delete last sentence of the paragraph.

Figure 3-8 would give a more intuitive feel for the possible updating sequences if it were rotated 90 degrees clockwise, with the "NO ENTRY" circle pulled up to the top of the diagram.

Disposition of Comment 122

See Issue 11.

23. Tony Jeffree

23.1 Summary comments

The full text of Tony Jeffree’s comments is reflected in section 23.2.

Comment 123

The operation of the GARP state machines should be revised in line with Mick Seaman's paper "Simplifying GARP" distributed and presented at the Ottawa interim meeting.

Disposition of Comment 123

See Issue 2.

Comment 124

The description of the GARP architecture should be revised in order to allow the transport mechanism (the state machines & PDUs) to be disentangled from the applications that use it (currently multicast group registration and Port filtering mode propagation). This will simplify future use of GARP for use as a means of propagating information on VLAN membership.

Disposition of Comment 124

See Issue 13.

Comment 125

The current mechanisms for static control of Group membership should be replaced with the management controls described in the papers referenced above, as these controls provide a more straightforward means of applying management control over any GARP-based application.

Disposition of Comment 125

See Issue 2.

23.2 Proposed revision of GARP architecture and state machines

23.2.1 Introduction

This paper constitutes the major technical content of my ballot comments on P802.1p/D4.

Mick Seaman’s paper, “Simplifying GARP”, describes the rationale for a re-think of the GARP state machines, for the following reasons:

a) To fix the known bugs in the 802.1p/D4 state machines (the Leave problem identified by Steve Horowitz, the slow Join case mentioned in Annex H, the dependence on timer races...etc.);
b) To simplify the state machines & at the same time, make them more robust.

“Simplifying GARP” describes a revised state machine that addresses the problems identified above. In the process, it also addresses the issue of how GARP operates under re-configuration conditions (topology change).

This document additionally describes a revised architectural and service description that will allow GARP to be re-used for purposes other than Group membership, and in particular, for distributing:

a) Port state information (Port Filtering Modes); and
b) VLAN membership information.

It also describes the set of override controls that are needed in order to allow GARP to be managed in a sensible manner.

The major changes that are visible in the revised state machine described in “Simplifying GARP” are:

a) GARP now operates on all Ports, whether the Port is in Forwarding or not. This ensures that membership information is up to date on all Ports, in preparation for a Blocking Port to switch to Forwarding;
b) As a consequence of a), the Leave All State Machine is much simplified, becoming just a regular “tick” on which Leave All messages get generated;
c) The Registrar state machine is now completely passive, responding to observed messages by changing state, and does not generate any messages itself;
d) The Applicant state machine now looks after all aspects both of Joining a Group and “shaking a group up” to persuade other participants to respond;
e) Further simplifications are possible, where it is known that a given Participant will never source frames destined for a Group, and when it is known that there are exactly two participants, i.e., the segment is point-to-point (not shared).

The revision to the architecture/service description is based on a paper written by Mick Seaman (not so far been distributed within 802.1), entitled GARP Information Transport (GIT). This paper is summarized in 23.2.2.1 below. This structure makes it possible to disentangle the underlying mechanics that allow GARP to propagate information across the Bridged LAN (the GARP state machines) from the applications that generate and make use of the information so propagated. In the current 802.1p, there are two such applications; one (the GARP application itself) which manipulates/propagates Group memberships and applies that membership information to the Filtering Database as a set of filtering rules; and one which manipulates/propagates Port Filtering Modes, and applies that information as Filtering Mode settings for its own Ports. Use of the revised architectural concepts described here will allow the protocol description to be simplified, and the applications to be described separately. Additionally, this will pave the way for further applications of GARP; in particular, its potential as a vehicle for distributing VLAN membership information in the port-based VLAN approach now adopted for 802.1Q.

23.2.2 The proposed revisions

This section describes the revisions to 802.1P necessary to achieve the aims introduced in section 1.

a) GARP Information Transport (section 23.2.2.1) describes the revised architectural framework for GARP;
b) Management controls (section 23.2.2.2) describes the control mechanisms needed in order to provide appropriate administrative control over GARP;
c) The GARP applications (section 23.2.2.3) gives an abbreviated flavor of how different uses of GARP can be described using the revised architecture.

23.2.2.1 GARP Information Transport

The components of the GARP mechanism are as shown in Figure 2—1, and as described below.

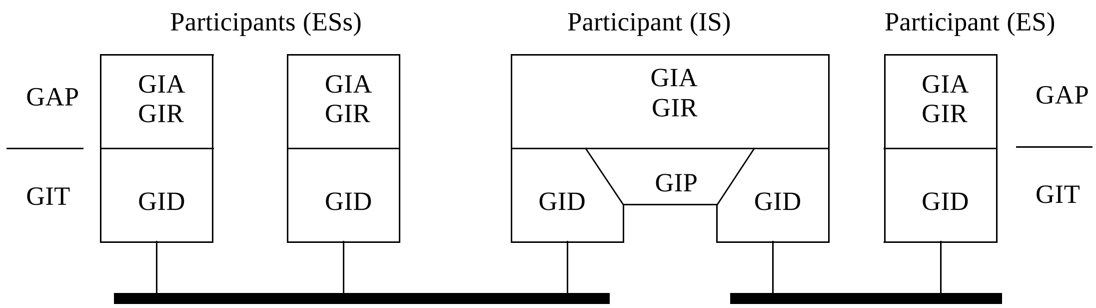

Figure 2—1 GARP components

At the Twente meeting of 802.1 there was discussion of the use of GARP to do ‘other things’, e.g to convey VLAN information. At the same time a question was raised - ‘what does this mean, for GARP is a complete protocol with a single purpose?’. What is ‘GARP’ about a protocol which attempts to achieve a different object?

This section attempts to clarify how, or at least state a view of how GARP can be used to transport different sorts of information. It partitions GARP into its constituent elements, distinguishing the workings of the GARP Information Transport (GIT, to coin one of several acronyms used in this note which may not survive long) from the application (e.g. group membership on a LAN segment) to which it is put. This distinction, and an examination of the internals of GIT, is then used to describe the characteristics of GIT and of the applications that night use it. This description can be used to evaluate whether a new application should really be using GIT, benefiting from its characteristics, or whether the association is purely accidental and of little merit.

Throughout this description there is a need for the concept of a ‘GARP like’ application or protocol, something which is both ‘GARP’ and ‘non GARP’ in the sense of the opening paragraph of this section. The term ‘G’ will be used to denote such a thing (actually it will be used to denote an instance, the set, and members of the set - which should be determined from the context, if it matters). GARP will be used to describe GARP specifically.

23.2.2.1.1 The Structure of G and GARP

G are partitioned into two major parts, the Application (GAP), and the Transport (GIT). Each of these can be further structured as follows:

a) GIT, the G Information Transport is the same for all G, independent of GAP and comprises:
   1) GID, or G Information Declaration, a protocol operated by a number of G Participants on a LAN segment (or a set of LAN Segments bridged together by G unaware switches). GID allows the participants to make declarations (such as ‘there is a member of group X on this LAN’), maintain them for a period of time, and withdraw them.
   2) GIP, or G Information Propagation, a set of rules by which G Information is propagated from one port of a G aware bridge to others.
b) GAP, the particular G Application comprises:
   1) GIA, the G Information Application proper, which makes the logical connection to the particular use of GIT in this instance, e.g. translation of notifications from GID into changes in the filtering database.

2) GIR, G Information Resolution, which resolves any discrepancy between the various declarations made by GID.

G are implemented in both end stations (hosts) and intermediate systems (bridges or switches). The implementation of GID is identical in both cases. GIP is only implemented in intermediate systems. Components of both GIA and GIR may reside in both switches and end stations, as required by the specific application.

Following sections discuss each of these components in more detail.

23.2.2.1.2 GID and Information Declaration

The GID protocol allows G Participants to make (declare) statements or declarations, register declarations made by others, and eventually to withdraw declarations.

In the context GARP, declarations are made about the membership (desire to receive multicast traffic) of Groups. Such a declaration is called a Join (formally it is a Join Request since a 'Join PDU' can be issued subsequently purely to maintain the declaration, as a part of GID protocol). In GARP, declarations are withdrawn by Leave.

These declarations have some important characteristics.

a) They are issued as statements of fact. There is nothing inside GID to resolve discrepancies or conflicts between statements. For example x=5 and x=10 would be two quite separate statements in GID. As a less abstract example, when GARP unicast registration could have associated priorities, it proved troublesome to get the GARP protocol to resolve two apparently conflicting priority settings. The view taken here is that such a resolution is outside the GID context, within which the two registrations would be equally valid and separate - and possibly propagated as such. It is GAP's job to sort this out, not GIT's.
b) They are relevant to the majority of participants and without GID many of them might send PDUs to make identical declarations. There is little point in using GID if there is to be no reduction in protocol overhead due to only one participant having to communicate the statement rather than several. In particular, in GARP, the statement is 'there is (now) a member of Group X on this segment', not 'I wish to join Group X'.

23.2.2.1.3 Interfaces to GID

GID provides interfaces to both GAP and GIP to allow the latter to:

a) Request a Declaration (GARP Req.Join).
b) Receive an Indication of a Declaration made by another G Participant (GARP Ind.Join).
c) Request Withdrawal of a declaration (GARP Req.Leave).
d) Receive an Indication that all participants formerly participating in a declaration have now requested its withdrawal (GARP Ind.Leave).

In addition, the controls described in 23.2.2.2 will allow the management functions of any participant to determine the Declarations currently in force on the LAN (segment).

23.2.2.2 Management controls

The interface to GID (between GID and either GAP or GIP) provides the means of applying management control over the declarations currently in force. For each of the primitives identified in 23.2.2.1.3, these controls allow the following three states to be applied, on a per-declaration (e.g., per-Group) basis:

a) YES. Setting the control to the YES state means that "X.Join" is assumed for the declaration concerned;
b) NO. Setting the control to the NO state means that "X.Leave" is assumed for the declaration concerned;
c) Dynamic. Setting the control to Dynamic means that the primitive is not modified by the control.

For example, consider Group M. There are two controls that can be applied to M. One controls the Req.X primitives that may pass into GID from GAP or GIP; the other controls the Ind.X primitives that GID will deliver to GAP and GIP.

d) Setting the Req.X control to YES for Group M means that, until such a time that the control is changed, GID will declare membership of M on the LAN segment to which it is attached. (In effect, a Req.Join is assumed to have been issued to GID, and any Req.Leave generated by GAP or GIP will be ignored).
e) Setting the Req.X control to NO for Group M means that, until such a time that the control is changed, GID will cease declaring membership of M on the LAN segment to which it is attached. (In effect, a Req.Leave is assumed to have been issued to GID, and any Req.Join generated by GAP or GIP will be ignored).
f) Setting the Req.X control to Dynamic for Group M allows Req primitives for Group M to cross the interface to GID without modification.
g) Setting the Ind.X control to YES for Group M means that, until such a time that the control is changed, GID will declare membership of M to GAP and GIP. (In effect, an Ind.Join is assumed to have been issued to GAP and GIP, regardless of the state of the GID's Registrar, and no Ind.Leave is issued, regardless of subsequent Registrar state changes).
h) Setting the Ind.X control to NO for Group M means that, until such a time that the control is changed, GID will declare non-membership of M to GAP and GIP. (In effect, an Ind.Leave is assumed to have been issued to GAP and GIP, regardless of the state of the GID's Registrar, and no Ind.Join is issued, regardless of subsequent Registrar state changes).
i) Setting the Ind.X control to Dynamic for Group M allows Ind primitives for Group M to cross the interface from GID to GAP and GIP without modification.

A further management control is available, that allows GID to be enabled or disabled on a per-declaration basis. In the disabled state, GID does not participate in any protocol activity for the declaration concerned.

Note that all of these controls can be applied on a per-GID instance and per-declaration; e.g., per-Port (in the case of a Bridge) and per-Group.

23.2.2.3 The GARP applications

23.2.2.3.1 Multicast Group Membership

This application of GARP was the original motivation behind developing the protocol; namely, managing the membership of Multicast Groups.

The operation of GIA/GIR for this application is straightforward, as follows:

For participants that are only receivers of frames destined for a Group or Groups (e.g., end stations):

a) GIA issues ReqJoin and ReqLeave service requests as appropriate to its Group membership needs;
b) GIA receives IndJoin and IndLeave indications from GID indicating the state of external memberships, which it can ignore.

For participants that are both transmitters and receivers of frames destined for a Group or Groups (e.g., multimedia servers, Bridges,...):

c) IndJoin and IndLeave indications from GID are used by GIA to update its local knowledge of Group memberships. In the case of a Bridge Port, this information drives the updating of Dynamic Filtering Entries in the Filtering Database, in a manner consistent with any static declarations that the FDB contains. In both cases, this information is used to suppress unwanted traffic on the attached LAN segment, by means of source pruning in Servers or filtering in Bridges;
d) In Bridges, the IndJoin/IndLeave indications are also used to drive the GIP function, below;
e) In other Participants that source frames (e.g., servers), GIA issues ReqJoin and ReqLeave requests as appropriate to its membership needs.

The operation of GIP is as follows:

f) All IndJoin/IndLeave primitives issued by an instance of GID are submitted to GIP;
g) An IndJoin is propagated as a ReqJoin to all other instances of GID;
h) An IndLeave for Group G is propagated as a ReqLeave to a given Port if there are no longer any members of G on the other Ports of the Bridge.

23.2.2.3.2 Port Filtering Mode propagation

This application of GARP allows Port Filtering Mode propagation to take place in order to cater for the needs of legacy segments in the Bridged LAN. The operation is very similar to that of Group membership; the primary differences are that there are only two possible declarations of membership:

a) Port Filtering Mode A membership;
b) Port Filtering Mode B membership.

The operation of GIA/GIR for this application is as follows:

For participants that are only receivers of frames destined for a Group or Groups (e.g., end stations):

a) GIA issues ReqJoin and ReqLeave service requests as appropriate to its Port Filtering Mode needs. For most end stations, this facility will not be used; however, some devices that wish to be able to perform promiscuous receive (e.g., monitoring devices) can use this mechanism to force local Bridge Ports to operate in Mode A or B as necessary;
b) GIA receives IndJoin and IndLeave indications from GID indicating the state of external memberships, which it can ignore.

For participants that are both transmitters and receivers of frames destined for a Group or Groups (e.g., multimedia servers, Bridges,...):

c) IndJoin and IndLeave indications from GID are used by GIA to update its local knowledge of Port Filtering Modes that have members. In the case of a Bridge Port, this information drives the current Port Filtering Mode for the Port concerned; the current Port Filtering Mode is set to the lowest Mode for which members exist, or to Mode C if no members exist. In the case of servers, source pruning is disabled if members of Modes A or B exist;
d) In Bridges, the IndJoin/IndLeave indications are also used to drive the GIP function, below;
e) In other Participants that source frames (e.g., servers), GIA issues ReqJoin and ReqLeave requests as appropriate to its Filtering Mode needs; for example, Routers may require local Ports to operate in Filtering Mode A or B.

The operation of GIP is as follows:

f) All IndJoin/IndLeave primitives issued by an instance of GID are submitted to GIP;
g) An IndJoin is propagated as a ReqJoin to all other instances of GID;

h) An IndLeave for Mode M is propagated as a ReqLeave to a given Port if there are no longer any members of M on the other Ports of the Bridge.

23.2.2.3.3 VLAN membership propagation

The operation of this application is very similar indeed to that of Multicast Group Membership; instead of propagating multicast addresses between Filtering Databases, GARP propagates VIDs. The essential points, without repeating the GIA/GIR/GIP descriptions in 23.2.2.3.1, are as follows (assuming our current picture of Port-based VLANs and one Spanning Tree):

a) Each Port of a Bridge has a defined VLAN ID associated with it that is used to tag incoming untagged frames. For the purposes of GARP propagation of membership, this looks like setting the Ind.Join management control to YES, in the terminology of 23.2.2.2 above;
b) Each Port of a Bridge will see incoming join/leave behavior relative to the active VLAN IDs in the Bridged LAN. These Joins and Leaves may originate from Bridges (propagating the information they see on their Ports) or from end stations, that can use GARP to declare their membership of VLAN RED, or revoke their membership of VLAN GREEN, and so on.
c) The result of this is that a directed graph is constructed across the Bridged LAN for each active VLAN, in the same manner as for multicast groups. This, in conjunction with the MAC address information (both unicast and multicast) in the Filtering Database is used to determine the destination Port(s) for a given VLAN frame.

24. John Grinham

Comment 126

I agree with Norm Finn that the priority mapping table 3.2 must be changed.

0	0 0 0 0 0 0 0	bits of priority field. 2 queues
1	0 0 0 0 0 0 1	use high bit, 4 queues high 2 bits.
2	0 0 1 1 1 1 2	On intermediate boundaries, differ-
3*	0 0 1 1 2 2 3	entiate priorities closest to 3/4
4	1 1 2 2 3 3 4	boundary, the "center" of the table,
5	1 1 2 3 4 4 5	before differentiating the ends.
6	1 2 3 4 5 5 6	Differentiate high priority levels
7	1 2 3 4 5 6 7	before low priority levels.)

In fact I would even consider making the user priority at which the traffic class is incremented even higher than in his table if there are more than two queues (depending on what applications we think people might actually place on user priority levels 1-6).

If people do decide to start splitting different applications over different user priority levels (e.g. bulk data at 0, IPX at 1 etc.etc.), unless we choose the map carefully we run the risk that the one class of applications that everyone agrees needs to be separated (real-time collaborative), won't be.

Disposition of Comment 126

See Issue 1.

Comment 127

Anything over 2 queues/traffic classes should be optional.

Disposition of Comment 127

See Issue 7.

Comment 128

Pure priority based de-queueing should be the default. Others can be optional but every 802.1p bridge should initially come up with straight priority.

I wouldn't object to incorporating some simple protection mechanism to prevent low priority frames being starved. (I don't think it's actually necessary but....).

Disposition of Comment 128

See Issue 4.

25. Wayne Zakowski

Comment 129

Page 8, lines 41, 42: "Overview". We need a better introduction into the relationship with the work of IEEE 802.1Q. There needs to be more of a description here (yes, I realize that this section will be removed in the final approved version).

Secondly, there needs to be a diagram (perhaps in an Architectural section such as clause 3 (Principles of Operation)) to help illustrate the working "partnership" on the tagged header of which functional bits that each 802.1p and 802.1Q are operating upon.

One suggestion would be to provide a figure similar to 3-7, in which there is a division of labor specified between the 802.1p forwarding process and the VLAN administration process.

Disposition of Comment 129

See Issue 12.

Comment 130

Page 23, lines 31, 32: "Queuing Frames", Clause 3.7.4. I support the notion of more than two traffic classes. While I understand the implementation motivation for just supporting two classes, I think that a case can be made for differentiation into at least three or four classes. I have supplied some comments about this in Annex F comments. But for interoperational simplicity, I would support the use of "2" as the "default".

Disposition of Comment 130

See Issue 7.

Comment 131

Page 25, line 1: "Selecting Frame for Transmission", Clause 3.7.5 For the de-queuing operation, this seems to be getting into implementation a lot. I do not think that this algorithm needs to be defined in greater detail. I think that it is sufficient to state that processing resources in the transmitter are devoted to prioritize the treatment of some traffic classes over others. Until there is strong agreement about the needs for greater than 2 classes, it will be difficult to precisely define how each queue is handled.........and you will always have one vendor arguing that their implementation can "more efficiently" process multiple queues!

Disposition of Comment 131

See Issue 4.

Comment 132

Page 41: "Management Protocol", Clause 7 I think that we are the "experts" in defining our functionality choices and are obligated to finish this work. I think that we should define the functional components. It will probably be fastest if we bite the bullet and describe these with the GDMO extensions to the 802.1j material.....BUT, we should work with the IETF to get someone to first establish a work item for an SNMP MIB, and secondly get some volunteers to see that this work item is progressed quickly.

Disposition of Comment 132

See Issue 6.

Comment 133

Page 52: "GARP State Machine Descriptions", Clause 9.6. I feel that Mick Seaman's proposal for a "Simpler Garp" is a step in the right direction and should be incorporated before we close the ballot on this component.

Disposition of Comment 133

See Issue 2.

Comment 134

Page 75: "Timer Values". It is acceptable to not specify a way in which management updates the timers, but the values need to be defined. Why can't the GARP timer values be specified in a unit such as "seconds". The use of "centiseconds" is not familiar to me.

Disposition of Comment 134

See Issue 8.

Comment 135

Page 89, Lines 1-15: "Performance Considerations", Annex F. I would like to propose a slightly different division of traffic examples.

a) For the item (a), I think that this should be called "Real-Time Control" information.
b) For item (b), I think that this should be called "Real-time User Application" information. This is referring to the category of user information associated with a "near" isochronous delivery capability.
c) For item (c), I think there should be a category called "Non-Time Critical Control" information. Examples of this would be to allow network management, and client-server control information to have the ability to be afforded a higher priority than file transfer or email.
d) For item (c), I think that there should be a second category called "Non-Time Critical User Application" information. Examples of this would be data transactions such as file transfer, interactive terminal support, and email.

I feel that for this proposed division to have real application impact in vendor products in a consistent fashion, that it is necessary to provide some provision for delay guarantees. In other words, there must be some tangible differences between "real-time" and "non real-time". My reasons for separating "control" from "user" is that there are always reasons for prioritizing the ability to sample status and being capable of reconfiguration if the original engineered configuration turns out to be suboptimal.

Disposition of Comment 135

See Issue 12.

Comment 136

Page 89, Line 29:"Performance Considerations", Annex F. Until we can make some definite association between the term "QOS" and some measurable performance elements, we should remove this association! Otherwise, we are adding to the religious confusion of claiming conformance to the QOS principles without offering any single means of adherence to measurable elements.

Disposition of Comment 136

See Issue 12.

Comment 137

Page 92, Line 28: "Non-goals", Annex G. Please remove the term QOS, see my reasons from the last issue.

Disposition of Comment 137

See Issue 12.

26. Vipin Jain

Comment 138

3.9.3.2 (page 29, line 36) still makes a reference to "existence". I read the sentence as "The permitted_port_set defines the set of ports on which dynamic registration requests for existence or membership of this group are permitted". Don't know if it is intentional.

Disposition of Comment 138

Editor to remove spurious reference to “existence”.

27. Summary of Issues

The following list summarizes the major issues raised as ballot comments. Each issue is cross-referenced to the ballot comment(s) that raised it.

Issue 1. Priority signalling

Summary of points made:

Q-related points:

a) Priority signalling should be separated from VLAN tagging (Comment 1, Comment 3, Comment 24, Comment 60);
b) Cannot publish this standard with a forward ref to priority in Q (Comment 42);
c) Resolve the reference to Q problem by referring to signalled priority, but not specifying how the signalling gets done; referring to Q only in a Note (Comment 81);
d) Specify the priority-related bits of the VLAN frame format in 802.1p (Comment 114);

Relationship between tagged priority & MAC priority:

e) Priority in VLAN-tagged frames should override MAC signalled priority (Comment 46);
f) MAC priority signalling should be used to convey priority in tagged frames (Comment 57, Comment 73);
g) Need a per-medium (802.5, 802.6,...) set of tables mapping 802.1p priority to M_UNITDATA priority (Comment 49, Comment 58);
h) Global significance of the VLAN priority drives how we map priorities in the Bridge (Comment 50);
i) Optimize traffic class mappings for the 2-queue case (Comment 48, Comment 126);

Other points:

j) Need to allow ISSLL to use all 8 levels of priority in the 802.1p/Q tag (Comment 51);
k) Need only 2 priorities (Comment 2, Comment 25);
l) Unclear as to the rationale for “Forward With Inbound Priority” control (Comment 8);
m) Should be possible to change the priority within the mechanism (Comment 29);
n) Need to indicate what the “highest” priority is (Comment 40).

Resolution of Issue 1

Issue 2. GARP complexity/resilience

Summary of points made:

a) The GARP mechanism as currently described in D4 is by far too complex. I propose to use the IGMP mechanism for IP multicast as a guide (Comment 4, Comment 5, Comment 28, Comment 44);
b) Need to fix the Leave problem (Comment 15);
c) Would it help to differentiate between end station and Bridge Joins? (Comment 20);
d) Use Mick Seaman's "Simplified GARP" (Comment 96, Comment 115, Comment 123, Comment 133);
e) Revise the management controls in GARP (Comment 125).

Resolution of Issue 2

<<Editor's note: when discussing relative complexity of IGMP versus GARP, it must be borne in mind that IGMP assumes that it has an RPC mechanism available to it. In the context of a Bridge, which does not necessarily have RPC available to it, IGMP is therefore not as simple to implement as it might at first appear.>>

Issue 3. Default Port Filtering Mode

Summary of points made:

a) What is the motivation for Mode C as the default Port Filtering Mode, and for providing mode switching? (Comment 7).

Resolution of Issue 3

<<Editor's Note: With C as the default, all Ports will gravitate towards the Port Filtering Mode that does the most filtering; i.e., the Bridged LAN will operate in the configuration that minimizes traffic. The plug N play works fine for pure GARP-aware environments; devices with particular needs (routers, sniffers...) advertise that need via GARP, and the appropriate Port Mode switching occurs automatically. That is essentially the motivation; providing a means whereby the LAN can self-configure to the state that involves the maximum filtering of multicasts, while also providing for the needs of devices that want to see more of the multicast traffic.

For GARP/legacy environments, it is not plug N play. It is necessary to configure GARP Ports that serve the legacy segments to Mode A in order for them to integrate. Not clear that there is a way round this; the alternative (default to "leakiest" state) results in having to configure all Ports serving GARP-aware segments. As (I assume) the future world is GARP-aware, it seems right to place the configuration burden mostly on the legacy world.>>

Issue 4. Priority queueing/dequeueing

Summary of points made:

a) Specify the strict priority algorithm as a default scheduling algorithm to be supported by all 802.1p bridges, but acknowledge that bridges may be administratively configured to support other (unspecified) algorithms (Comment 27, Comment 31, Comment 41, Comment 43, Comment 101, Comment 119, Comment 128);
b) Agree with current description of de-queueing algorithm (Comment 9);
c) Little alternative to highest priority first (Comment 76);
d) Remove Note 2 on Page 24. Vital to recommend one default algorithm (Comment 66);
e) Make no recommendation on de-queueing algorithm (3.7.5 is acceptable) (Comment 33);
f) De-queueing algorithm should be left to the vendors (Comment 85);
g) Current description is too detailed/implementation-like (Comment 131).

Resolution of Issue 4

Issue 5. Interoperability with IGMP etc.

Summary of points made:

a) Need to clarify the sequence of IGMP & GARP interactions (Comment 17).

Resolution of Issue 5

Issue 6. Managed objects

Summary of points made:

Overall MIB definition:

a) Define the functionality of the M.O.s & let IETF define the SNMP stuff (Comment 19);
b) Write a pseudo-MIB, but rely on IETF to do the real one (Comment 37);
c) Let the IETF do it all. If they won’t, then 802.1 will have to. (Comment 54, Comment 93, Comment 111);
d) Define extensions to 802.1j first, then raise a PAR for a MIB definition later (Comment 69, Comment 118, Comment 132).

Detailed comments on management operations:

e) The "Set Port Inbound Priority Handling To Default" and the "Set Bridge Filtering Mode to Default" commands should be removed (Comment 36);
f) May need to be able to “clamp” priority in/out, perhaps per-VLAN. Does this belong in P or Q? (Comment 39);
g) Need to be able to specify address ranges for management operations (Comment 83);
h) Management of the port Traffic Class Table should be optional (Comment 92);
i) No need for output parameters on 6.6.3.4 and 6.6.4.4 (Comment 105, Comment 109);

Resolution of Issue 6

Issue 7. Traffic classes

Summary of points made:

a) The recommendation on the number of traffic classes should be to support “at least 2” (Comment 26, Comment 30);
b) Make no recommendation in the standard; rely on rationale in Annex F (Comment 32, Comment 47);
c) Restrain number of classes to 8; 2 is too few (Comment 54);
d) Should not recommend 2 classes. The mapping of 802.1p priority -> queues should be up to the implementation (Comment 65);
e) Happy with 2 classes, as long as it is not expressed as a requirement (Comment 75);
f) The draft text is fine; no need to specify number of classes (Comment 84);
g) The recommendation of two classes should be there somewhere; maybe as a Note (Comment 100);
h) Need to specify the number of classes, in order to avoid having to implement 8. 2 is probably the right number, but maybe slightly more than 2 (Comment 120);
i) Anything over 2 queues should be optional; use 2 as the default (Comment 127, Comment 130).

Resolution of Issue 7

Issue 8. Timer values

Summary of points made:

a) Fixed timer values are OK (Comment 38, Comment 117);
b) Centiseconds is fine, but don't mandate that all devices support that fine a granularity (Comment 56);
c) Should allow setting GARP timer values (Comment 110);
d) Standard should mandate a (low) value of timer resolution (Comment 116);
e) Specify timers in units of seconds (Comment 134).

Resolution of Issue 8

Issue 9. Service description

Summary of points made:

a) Priority tagging service should be described in section 2 (Comment 45);
b) Is it necessary to describe unicast filtering service? (Comment 82);
c) Need to re-vamp the service descriptions (Comment 113).

Resolution of Issue 9

Issue 10. Description of the Forwarding Process

Summary of points made:

a) 3.7.7 should not preclude making adjustments to (as opposed to complete recalculation of) the CRC (Comment 86);
b) Improve the description of the Forwarding Process (Comment 121).

Resolution of Issue 10

Issue 11. FDB entry descriptions

Summary of points made:

a) Should be possible to convert a Static Filtering Entry to a Group Registration Entry (Comment 88);
b) Improve the Filtering Database description to make it less confusing (Comment 122).

Resolution of Issue 11

Issue 12. Miscellaneous issues

Summary of points made:

a) Annex F should draw the distinction between delay-sensitive and loss-sensitive traffic (Comment 94);
b) Need a better description of P<-->Q relationships (Comment 129);
c) Modifications to terminology in Annex F (Comment 135);
d) Should not use "QoS" in Annex F until we know what it means (Comment 136, Comment 137).

Resolution of Issue 12

Issue 13. Architectural issues

Summary of points made:

a) Re-describe the operation of Port Filtering Mode propagation as a distinct GARP application, and make adjustments to the forwarding rules accordingly (Comment 1, Comment 97);

b) Revise the architectural description of GARP to distinguish between the transport mechanism and the applications of GARP (Group registration, Port filtering mode propagation and possible other apps in the future) (Comment 124).

Resolution of Issue 13