P802.1CQ Discussion
2019-11-13

Roger Marks (EthAirNet Associates)
Antonio de la Oliva (Interdigital)

2019-11-13

From minutes of TSN Call of 2019-11-04:

P802.1CQ Multicast and Local Address Assignment
(https://1.ieee802.org/tsn/802-1cq/)
Roger Marks, the Editor presented:
http://www.ieee802.org/1/files/public/docs2019/cq-Marks-telecon-discussion-2019-11-04-1119.pdf

Disposition: further discussion needed
The discussion will be continued at the November 802 Plenary next week.

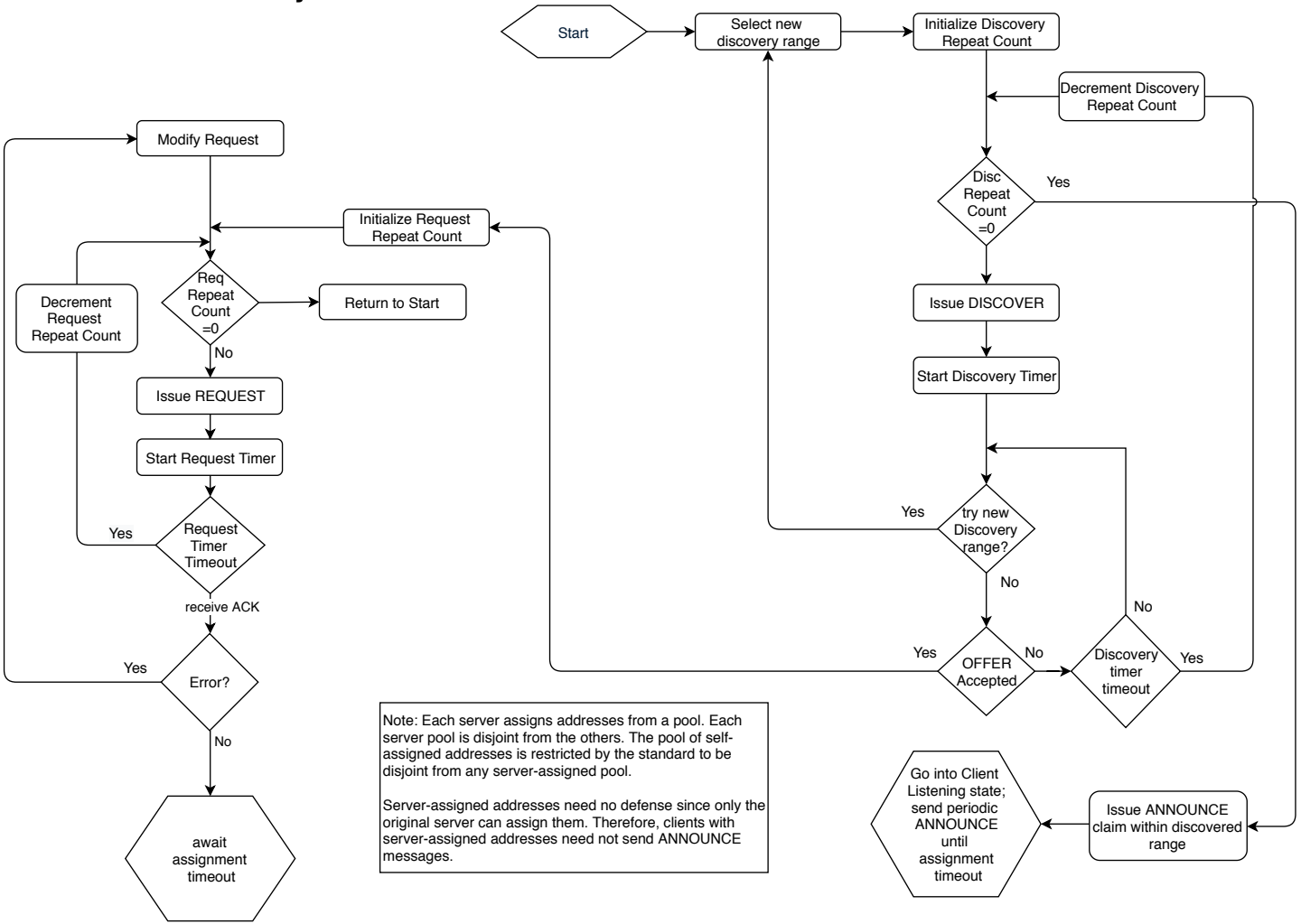
The Editor explained that first TG ballot is expected in January.

Updated PALMA flowcharts

- Updated PALMA flowcharts presented in
  cq-Marks-telecon-discussion-2019-11-04-1119
  - Client Discovery
  - Client Listening State, Self-assigned Client
  - Server

- Here we repeat those three flowcharts, with an update to the Client Discovery

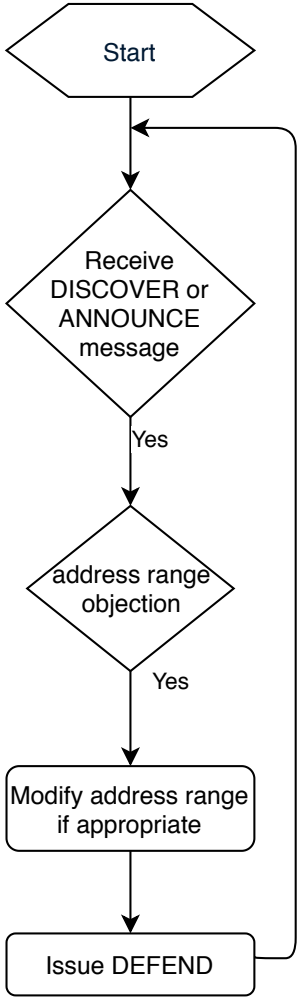
Client Discovery



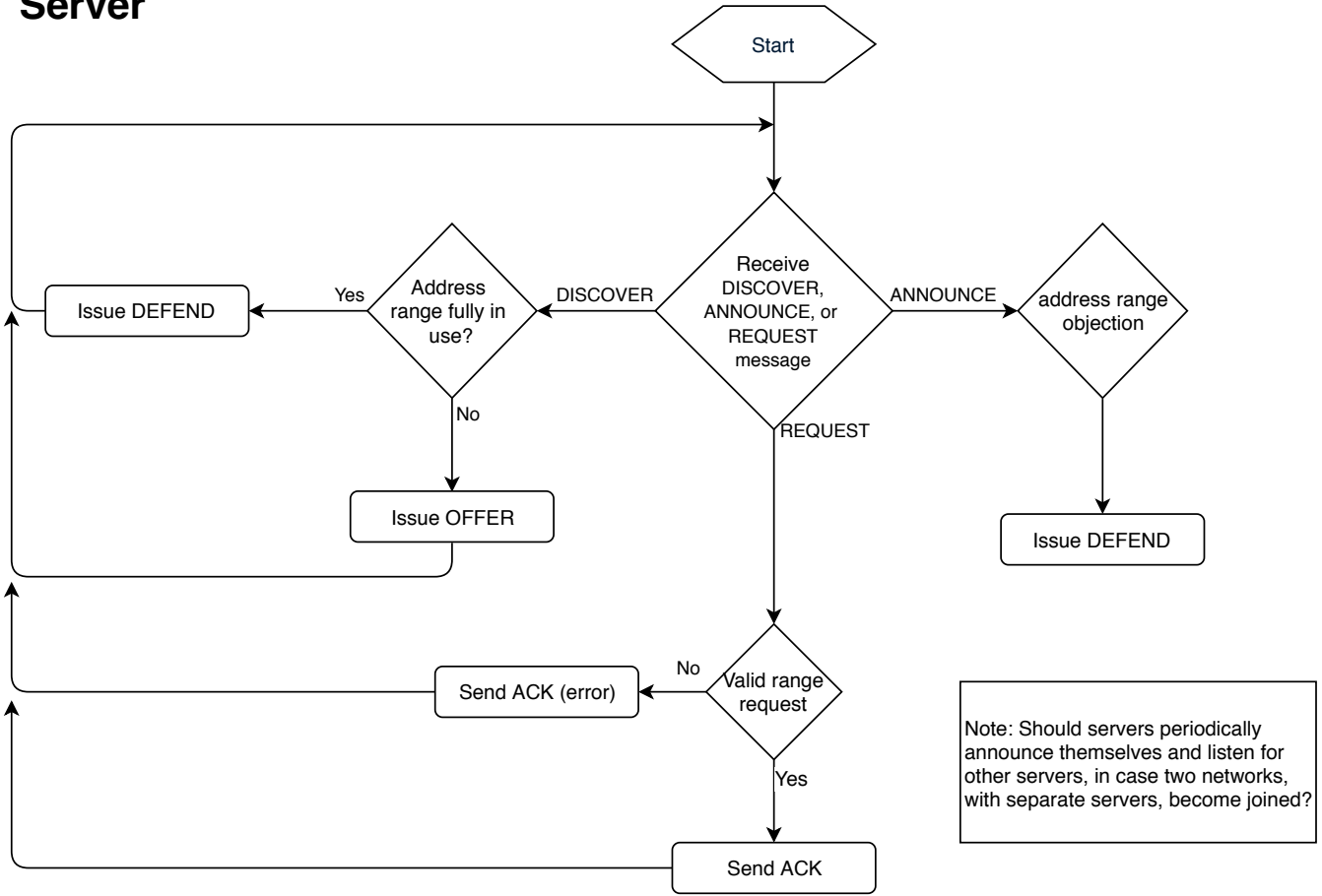
Client Listening State,

Self-assigned Client

Note: Clients with server-assigned addresses need not receive or respond to the multicast DISCOVER or ANNOUNCE messages, since the server can respond to them.



Server


Start
Receive DISCOVER, ANNOUNCE, or REQUEST message
DISCOVER
Address range fully in use?
Yes
Issue DEFEND
No
Issue OFFER
ANNOUNCE
address range objection
Issue DEFEND
REQUEST
Valid range request
No
Send ACK (error)
Yes
Send ACK
Note: Should servers periodically announce themselves and listen for other servers, in case two networks, with separate servers, become joined?

Reflector Followup to TSN Call of 2019-11-04 (1)

Craig,

On the call, you had a suggestion. I've thought about it, and I have an alternative proposal.

My client flow diagram included a text box to cover the case of a server outage. It said: "Clients with server-assigned address need not send periodic multicast ANNOUNCE messages, since the server can handle the defense. Should client with server-assigned address send period unicast message to server to confirm that server is still active?"

You described an alternative, which I understood this way (in the language of the slides):
(a) The server-assigned client would listen for DISCOVER messages.
(b) When the server issues a DEFEND, it uses multicast.
(c) If the server-assigned client hears a DISCOVER that needs to be defended, then it listens for the multicast DEFEND from the server.
(d) If it does not hear that multicast DEFEND, then it issues its own unicast DEFEND.

This saves unicast traffic, but it adds multicast traffic; maybe a lot of it. On the prior call, Norm expressed a concern with the extra multicast when all the clients need to listen for DISCOVER (even though many, or even all, of them may have server-assigned addresses).

So, I came up with yet another alternative, which is my favorite so far:

(1) Self-assigned addresses and server assigned addresses will be restricted by the standard to disjoint address blocks.
(2) Server-assigned addresses therefore need no defense since only a server can assign them. Clients with server-assigned addresses need not listen for DISCOVER messages (or send ANNOUNCE messages). They are not part of the claiming process at all. If a server goes down, all of the existing server assignments operate normally.

I'm planning to include this idea in my presentation for Waikoloa. I'm open to feedback before or during that meeting.

Cheers,

Roger

Reflector Followup to TSN Call of 2019-11-04 (2)

Just a reminder of the side effects of changing a clients assigned multicast address: the associated stream(s) must go down and come back up on the new assigned multicast address. That means a glitch in the audio, video, or other stream contents.
I mention that as I think about your proposal. Let's say you have the scenario I suggested on the call. The network comes up with a server in place and most of the talkers get assigned multicast addresses; however there are still a handful of nodes that haven't booted for whatever reason.
Now something breaks the network into two segments by powering off a switch, pulling a cable, whatever. As the remaining nodes come up they will get server-assigned addresses if they are on the segment with the server, otherwise they will get client assigned addresses if they are on the segment without a server.
So now all nodes have addresses assigned. Next, the network gets healed and the server is now present for the entire network. One of two things will happen, the client-assigned addresses will get changed to server-assigned, which means a glitch in the audio. Otherwise, the clients will continue to operate as-is, some with server-assigned and some with client-assigned; and maybe that is not a problem at all.
Now the entire network is shutdown overnight and brought back up for the next show on the following day. All the clients would send out requests using their "remembered" multicast addresses from yesterday, which guarantees an immediate startup because there can be no conflicts since there weren't any yesterday. The talkers that used client-assigned addresses will get NAKed and forced to change over to server-assigned addresses since the server won't like the suggested address range. Again, maybe that is okay.
Or maybe they never ask the server again and just stick with their client-assigned addresses. That feels wrong to me since we now have a server on the network and the clients with client-assigned addresses never ask the server to assign an address. So now the network has a mix of server-assigned protocols running with client-assigned protocols. I don't think we want to allow this case. If you agree with that, then does it both you that the network would be running exactly that way as long as it wasn't power cycled since the network was recombined as mentioned above? I don't like that inconsistency. Therefore, I think that client-assigned addressing should switch back over to server-assigned as soon as a server is detected. The server could realize there were client-assigned talkers running on the network and NAK (not the correct word, but the correct concept here) the addresses they announce and force them back to server-assigned; another cause of glitches in the audio.
If we don't segment the range between server and client assigned then there is a chance these glitches would not occur.
We can talk about this next week during your presentation time. Be sure to ask for enough time for discussions like this. I also don't want your presentation to be in the 60802-only session since I will be chairing the TSN sessions at those times.
- Craig

Reflector Followup to TSN Call of 2019-11-04 (3)

Thanks, Craig. This is a good topic for discussion next week. I'll try to give a brief view on this scenario:

I don't foresee any harmful operational effect of having some devices with self-assigned addresses and some with server-assigned addresses. The network won't care. The only problem is that the self-assigned addresses need to be tended after, which leads to extraneous communications of some sort.

If, with the split address range, a client holds a self-assignment, it should probably look for an opportunity to swap to a server-assignment when it can, since that will free it from the ANNOUNCE-DEFEND rat-race. Boot time is a perfect opportunity; in your scenario, as I understand, it would send a DISCOVER and get an assignment from the server.

I like the divided address range because it allows at least some of the devices to escape that rat-race when there is a server available. I want to avoid condemning every device to the rat-race just for the sake of consistency.

I'll try to get some slides together for next week to frame the discussion.

Cheers,

Roger

Address Pools and Reboots

- Each server needs a disjoint address pool
- Serverless operation needs a disjoint address pool.
- Those address ranges vary; that's no problem.
- It's a good idea to support retained storage, as in MAAP:

  > *If the application has previously obtained an address range and has access to persistent storage, the application may record the previous address range and attempt to reuse the saved address range.**

- At bootup, a device may issue a DISCOVER with the prior self-assigned address.
- A server may offer an address.
- Devices should prefer server-assigned addresses, because self-assigned addresses result in a lot of multicast noise.

Updated PALMA Description

- Updated PALMA description presented in
  cq-Marks-telecon-discussion-2019-11-04-1119

- Here we repeat those edits (in yellow), with a trivial update (in green).

5.2 PALMA

The Protocol for Assignment of Local and Multicast Addresses (PALMA) is specified herein. PALMA is specified as a single protocol supporting both server-specified allocation and serverless peer-to-peer claiming-based assignment. Both claiming-based and server-based PALMA handle the case in which the station lacks a valid MAC unicast address assignment prior to execution of the protocol.

The PALMA protocol is initiated by the PALMA client's transmission, from a source address selected ~~from~~within a specified range to a known multicast address, of a DISCOVER message, optionally specifying a request for a preferred local unicast or multicast MAC address range out of a pre-established set specified herein (specified in subclause <<TBD>>). The client need not know in advance whether PALMA servers or PALMA client peers ~~or~~are available on the LAN, but the DISCOVER message can be limited to stimulating responses from only servers or only client peers.

A PALMA server, listening for ~~probe~~ DISCOVER messages at the known multicast address, responds to the DISCOVER message with a unicast OFFER message, specifying the particular addresses available for assignment in accordance with the DISCOVER request. If this assignment is acceptable to the client, it responds with a ~~<<multicast?>>~~ REQUEST message to the server, which terminates the exchange with a ~~<<multicast?>>~~ ACKNOWLEDGE message.

A PALMA client ~~peer~~ that has been assigned an address from a PALMA server does not respond to a DISCOVER message, since the PALMA server can presumably respond on its behalf. However, a PALMA client that holds active self-assigned addresses ~~assignments previously allocated by a PALMA client peer~~ is required to listen to the known multicast address and respond to a DISCOVER message with a unicast DEFEND message if the request specifies addresses that the responding client has ~~been~~ previously self-allocated and wishes to retain, or in case it has been programmed by an administrator to defend against certain requests for administrative reasons, such as if the request is larger than the administrator allows.

After issuing the DISCOVER message, the client awaits OFFER and DEFEND messages. The DISCOVER is repeated a specified number of times after a specified waiting period. The client then adopts addresses from its claimed range, with the exception of those defended, and issues a multicast ANNOUNCE message specifying those addresses. The client then enters the DEFEND state, during which it listens for a DISCOVER requesting any of its adopted range of addresses while also issuing ANNOUNCE messages periodically. If the client, upon receipt of one or more DEFEND messages during the discovery period, discovers that its original request is too blocked for acceptable use, it may re-initiate a new DISCOVER message with a different address range. If, during the discovery period, the client receives, from a PALMA server, an OFFER that it considers acceptable, it responds with a REQUEST to inform the offering server, terminating the repetition of the DISCOVER message.

In the presence of a server, the PALMA ~~a~~ four-message (DISCOVER–OFFER–REQUEST–ACKNOWLEDGE) exchange is similar to that of DHCP.

In the absence of a server, PALMA takes a form similar that of MAAP. Using MAAP, a station can claim a multicast address range to be unique on the LAN. IEEE 1722 intends for MAAP-acquired addresses to be used to identify frames in time-sensitive streams, not for use as IEEE 802 MAC source or destination addresses. Nevertheless, the MAAP framework is well suited as the basis for assigning unicast IEEE 802 MAC addresses. Although IEEE 1722 identifies a specific block of multicast addresses for MAAP assignment, it does not limit MAAP operation to multicast addresses in general nor to the identified MAAP multicast address block in particular. MAAP presumes that the requesting station holds a valid MAC address, while PALMA covers the case in which the station lacks a valid MAC unicast address assignment prior to execution of the protocol. The PALMA message format is a significantly modified version of the MAAP message format.

<<Editor's note: Should the DISCOVER be restricted to claiming only a single unicast MAC address, or perhaps only up to a few of them, to avoid excessive claiming? For multicast addresses, perhaps larger blocks could be expected and enabled, as in MAAP.>>