PIM chairs need to clean this raw text:

Go to 00:00:37

Okay, let's get started. Yeah All right Welcome to Welcome to PIM 122. Not too many people here, but I think we should get started How many do you have a lot of people? remote? A few. Yeah. Okay So I hope you all have seen a note well or maybe you haven't seen it yet yet this week. It should be the same as last meeting we well yeah you should all all read this and make sure you follow it Here's the agenda. So it's a pretty packed agenda this time So we should I guess, hurry up any comments on the agenda So I think we have like 10 minutes spare at the end here but i expect that some of these might take a little longer than the schedule scheduled time Okay, I'll quickly go through the working group status so

00:02:00 - 00:04:00

Go to 00:02:00

let's see since last time we have we have the best drafts, the IGMP version 3 and MLD version 2 and the the 3228 biz they are in the RFC editors cube but they are just about to get public shortly and a couple of these are like an off 48 state right now now then we have the last bisdraft 1112. Torres will present that shortly shortly. So we didn't really have enough responses during the working group last call, but Brian gave some good comments and and Troilus has done a lot more work so thanks for that and probably want to do another last call shortly um DR improvements are not discussed this time and BD BDR neither Let's see okay, I've used this one Okay, point to multiple, point to multipoint drafts we are requested publication. So waiting for for a d and IESG. PIM light was just published. that's great. thanks to Sandy for being the shepherd for that that we got the joint prune extensions for LISP there on the RFC editors queue We also requested publication for the MOFRA draft So that's in IESG processing processing um then we have a few a couple more draft that I'm presented this time A couple of young models Lessons learned will be discussed this time time and finally let's see we got some some autocomph drafts that they will be discussing discussing and let's see yeah, they

00:04:00 - 00:06:00

Go to 00:04:00

got the multipath IGMPLIA proxy. And finally, got this pfm forwarding enhancements that will not be discussed this time So that's the working group status any comments before we start on the first? presentation? presentation? presentation no tourless you're up them So you should be able to use this thing thing. Yeah, it just, I'll pay you on yeah Yeah. Okay. Right. So um last version before this one IETF 120 went into working group last call last call window closed shortly before 121 as far as the timeline everybody was happy uh-huh except for for brian who provided some very short questions and I went to the microphone and said let's quickly be done, famous last words and then it turned out to actually you know rambling into one thing thing that took a lot longer So the easy things right so um the meaning of IP is now RFC-80-200 compliance before, obviously, it meant meant IP4 because when it was written there was nothing else else and now it means like 8200 any version of IP that of course is also explained so there was a lot of changing of IP into IPV4 and then of course also several fix-ups where you know the wonderful people of the IETF chose to have new naming for IPv6 like Hop Limits instead of TTL and other stuff i could rant about but won't um So the introduction explains now also also the

00:06:00 - 00:08:00

Go to 00:06:00

content of the document better and one of the things I walk up to is that I really need to expand the title to say host extension for IP multicast and because in before it was just called host extension extensions but now it is of course not the complete host extension it's only the host extension and the ASM service interface on top of it but it doesn't describe the SSM service interface on top of it And I needed to take these two things apart and describe both of them in the introduction so that people understand there are two service interfaces, ASM and SSM they share the same host stack, right? Because there is no different Ethernet binding for SSM, example, and so on So that was also simple, it's just the introduction text is now new and then of course also the wonderful wonderful new versions of IGMPV3 and MLDV2 are now referenced as a opposed to the old and buggy ones that we're still having until, you know, these two new ones come out out well and that's when the trouble started so So there's this wonderful wonderful small text block that was rambling of Mr. D. It's in parenthesis, why not simply remove it? And then of course Brian stumbled across it And I was looking through it and figuring out what the heck to do it with it And well, you know, it has a couple of outcomes. So this is the rest of the presentation talking about this, right? So the special treatment for 2240 -001, that actually is in that text, but it actually belonged into the IGMP version 1 text, which was already removed So that part that we wanted to remove, and the same text with respect to the special treatment is also an IGMPV2 and V3 and an MLDV1 and V2 for V6 of course we decided to change that behavior but so removing that part was fine There is a new

00:08:00 - 00:10:00

Go to 00:08:00

subsection of the whole section 10. Section 10 is why the heck, you know, are there changes? and what do they give us? But as far as the normative specification, that part is just removed now interestingly enough IGMPV3 light rFC probably an oversight does not have this, you know, must always join to the group, which is also fine, right? So it's really a decision of the IGMP or MLD protocol. So that was simple now the other part the the beginning part, now all host groups are record recognized as a special case, never forwarded beyond a single network That is actually a router requirement, right? So that whatever TTL you send, you're not forwarding it. It's a link scope address. Now I didn't introduce the word scope, but I did look um can I reference an RFC that normatively state what link local is for IPV4 and there isn't, right? right? The only stuff where that later happens is in, you know, INA registry RFCs, which of course later became obsolete and are now just Ianna registries so because there was no normative text for it in another normative RFC, I add text to say, well, link local and there's another name for it it local network control block. This is link local addresses There is now a new kind of of paragraph section router requirements, which just talks about about the link local I didn't introduce the word scope because the scope stuff came later and And I pointed to the RFCs that introduced the overall scoping for IP So that was basically the leftovers of that small red part and of course there is all also be a section in subsection section 10 explaining that that so there is also the even more interesting thing that that the IGMP protocol specify not to send IGMP membership for 22401, which wouldn't be compliant with RFC 1112, so I wanted to see what can I do for RFC 1112 BIS so that that behavior

00:10:00 - 00:12:00

Go to 00:10:00

is compliant or should it be compliant? So actually turns out that I remembered the history of this holy crap which was that the use of 2240X addresses was started to be used in router protocols like OSPF and others even before multicast and IGMP implementations were being done you don't need to have IGMP to do link local scope multicast, right? You can perfectly send and receive these packets with just the Ethernet filter without any IGMP protocol because the IGMP is only good for a router to get traffic in from another network and because it's linked local you don't need it. So the only reason why we chose to mandate MLD to do that also for Link Local IPV6 was because we also wanted to support MLD snooping switches But we had already defined in the RFC for snooping, informational, that that we shouldn't expect any IG and link local with IPV4 yada yada yada so this ended up being then the text that says, you may not send IGMP for link local IPv4 addresses so there is a may now that permits it making all the other RFCs you know in compliance with that but still if you do like to say send it for whatever reason, nobody prohibits you to do that And then this whole history of the implementations, that is now a whole appendix in number three appendix A3, that's the longest thing because it's really convoluted for people to understand because, you know, when I joined Cisco, I didn't understand why the heck are you not sending IGMP, right? in your OSPF implementation see this this OSPF RFC even the first first one talks about that that you should send IGMP and why? and I well because we say so yeah so technically why well no good reason right so that's uh history that I put in there so people understand all the technical background but the total changes in the main normative text are really quite minor Can you ask questions now or? Sure. So the first

00:12:00 - 00:14:00

Go to 00:12:00

question is about two 2D4 001. So you say that it can be left to the IGP protocol, right? But are there other RFCs, possibly, that are using or relying on hosts responding to the 4-001? Yeah, what Steve Deering said that other protocols may use it, think I'm trying trying to remember if I found another RFC to use it and I'm drawing a blank right now okay because Because, I mean, at this in theory if hosts in the future you end up you know implementing something other than IGMP or this that they don't really need IGMP They should, they may still have to listen to that address you know it's used for other purposes Yes, I mean, we still have enough, you know uh 224 zero 224 00x addresses left over, right? So I think we would make a good look whether we give out a new address for a link local new use for IPV4 or whether we use the old all host all multi-cost hosts actually it is right this my point is if it might be used for other purposes, than just IGP protocol then it might be best to leave it as possible of the multicast host requirements and not have it only in the IGMP protocol right now i think this is orthogonal I think whether or not you need to send IGMP memberships for any of the 2240 X addresses that is really the decision the only reason why you need it is because you might want to have an IGMP snooping switch that other otherwise doesn't forward the packet to you and we already have the IGMP snooping RFC saying, no, must not expect IGMP membership report in 22400X. Well, least one example is people sometimes ping that multiclass address just to check which hosts -cast hosts they have on the land So there could be reasons like that to. And that works fine, So because we we already excluded for

00:14:00 - 00:16:00

Go to 00:14:00

IPV4 IGMP snooping to expect IGMP memberships So at this point in time, is really no good reason why anybody would ever want to send IGMP memberships other than, you know they didn't understand the history and when they read rfc 1112 other groups in 224 000 x you know would you would expect that you should send it it yeah um okay just one more brief comment you said it's fine fine that the IUP light doesn't mention that it should join this group but it's important right Because it needs to get crazy. No, no no it's um that you you always join it So I think Steve Deering, did try to figure out how the heck he implemented that, but the only text that was no, you need to join it if you build IGMP at user level otherwise you don't get the packets through the Ethernet filter, right? Yeah. So there is no risk that you don't join it through the Ethernet filter, right? otherwise you wouldn't receive the packets you don't need to send out the members the IGMP memberships sure that's Also related question question, David Lumpurter, I do know that switches do behave like this for the slash 24 but you're listing 224 slash 8 here. Do we know that? did i make a typo in there is there a slash 8 it should be 624 yeah yeah yeah, course Hopefully it's not in the the draft, right? so but uh you have a wonderful tablet to check the draft. So thank you you you five minutes okay right um right so the explanation section so um is i'm service model, right, the initial summary now is explaining better a SMS SSM as the service model with a common host stack stack and then these new explanation subblocks right, 10.7, that's the link local in the INA registry, it's called the local

00:16:00 - 00:18:00

Go to 00:16:00

network control block, 10.8 that's a special treatment of 2 to 4 001 and then 10 .9 is the, you know IGMP MLD messages for link local IPV4 host group addresses addresses So, and the sales pitch, of course, that that we need to be aware of of is that one of the big challenges when we hopefully at some point in time get rid of this in our work group and get it over to the the IETF i esg is that we of course want to have the standard full standard back that 1112 has and these things actually you know yes they do add more explanatory text but the actual requirements and explanation now make this align better with the reality of, you know, hundreds of millions of deployed devices than 11 1112 was right because it didn't have that may to say, well, you don't have to send ID IGMP memberships in that space for example and a couple of other details, right? So, but I haven't tried to, you know, know, summarize kind of a sales pitch, but I've tried that all the changes that were done do make it better in that respect or not worse and we're closing standards gap right we were closing the gap that there was no standard specification for defining the link local that was kind of hidden in parenthesis in text that you know, now it says must right you must not for to 4 packets, whatever TTL they have And that obviously is also something that everybody does right so i think that was it it. Yes, about the RFC 791. Oh, yeah, the update. did I? forget the slide about it? maybe forgot the slide to add a slide about that, yeah. you good, thank you Right so i was uh and i have a section in the text now that justifies this. The whole point is we had an open issue, it was a to do, whether or not to call this RFC an update

00:18:00 - 00:20:00

Go to 00:18:00

to IPv4 791 and uh i finally decided if I just have it as a to-do when it goes to to IESG, nobody cares, right? So we need to make the claim. Of course, it's an update to 7-9-1 and then if the majority of IESG doesn't like it then, well, we drop it right so my argument why it is an update is is because by implication you look through 791 and there is nothing that tells you that some address range, let's call it these, know, um, 2 to 4 slash 8 range would not be subject to the forwarding and processing rules of IPv4, right? so if Unicast if if anybody would apply, IPV4 unicast processing to these, you know, possible you know, madness could evolve with someone that actually is doing multicars. So from that perspective, because because we're taking the address range away and changing completely changing the forwarding rules on it, it an update to 791 so that that's the argument and i'm i'm not adamant of getting it through if it is fine there is a another complex IETF bureaucracy to it in so far as that there is the standard five and that's actually a package of all the things which are not updates they didn't have the term update when 7-9-1 was written, but they have a standard five that says this is all of the good IP stuff. RFC 11, 1112 is part of that as well So obviously there is also the ask to make the BIS part of standard five again and replace 1112 in it And there's, of course, standard 86 which is for IPV 6 where of course, they did everything different standard 86 does not include any of the extensions of IPv6 You know, go out, figure it out yourself So I also then said, well, we're not bothering about standard 86 so so standard things in preparation of what comes ahead with i sg are also now in this version. So this is a lot of process this is a what is a 35 year old documents so

00:20:00 - 00:22:00

Go to 00:20:00

definitely, know, there is nothing wrong in taking our time and doing all these improvements and was so happy that Brian triggered me to revisit that part and then, you know fall into the trap of identifying all these gaps that were missed around ron yeah thank thanks for all that work and uh we'll check with our AD and OSG in general maybe how to do some of this Yeah, I was just thinking that I'm not sure the ramifications of changing a draft name is when it's already working group draft but it seems like this would probably be better to be named pym multicast Protocols or yeah, less maybe yeah it doesn't really matter Once it gets published, well, if it gets published published no one will care what the original draft name is is. So when it gets published, can just change the name to whatever you want you mean the draft name right yeah the draft name. Yeah, no one will see that, right? will just be rc some number yeah Yeah, that's true Because this is focused on protocols and so many of you are familiar with this draft we have a history of protocols that were just giving an overview of it came up in a couple years ago that would be helpful to have this kind of a draft and so we've been slowly developing it it's going to be a draft that we'll be working on on public for some time the one that we just added from the last IETF was on Reliable Multicast There are a couple additions that we're going to add to this this based upon what we have in the draft today today um but a lot of us don't have a whole lot of the history of reliable multicast. We weren't really involved in the RMT i know some of you were but i yeah and Bill, but I wasn't

00:22:00 - 00:24:00

Go to 00:22:00

wasn't. And so their purpose was to standardize reliable multi-gas transport and so i just included a little bit about that in this last update talking a little bit about NORN and ALC protocol and just talked about, you know, what their purpose was and how the pinworking group later produced our own experimental extension to PIM which is port for reliable transmission of PIM joined PURM messages and um what we don't have in here is maybe a good history of why we've, why the RMT working group closed maybe and why we're at where we are and what we may need to develop in the future but this is what we did add in the last last I since the last i atf and we're going to continue to make changes to it hopefully it's a helpful document to the industry and thanks to you to provide these suggestions on what we can include Does anybody have anything that comes in mind right now? that we can add? for the next update? update I'll try to reach through that reliable multi- section and see if I want to add something to that because that obviously was a very interesting process thing right So the problem was really when the the working group was created Every research on the planet came with three proposals proposals and so it became pretty clear that that no the working group could not create good, comprehensive, reliable multicast protocol Instead, they said, we're just going to define building blocks And in parallel, there was the thing that actually, you know, the you know five big players typical Cisco, Microsoft, and I forgot one in the industry had implemented a very good product on the

00:24:00 - 00:26:00

Go to 00:24:00

IETF said go away you're just one out of of, you know, 500 research papers that we don't want to adopt and so that had this convoluted process of outside of the IETF defining something called called RFC 3250, I think, is the number of PGM, pretty good multicast which was implemented you know, Windows XP, Tipco, everywhere, different routers, of course and then ultimately the IETF gave in and gave it actually, what is it experimental or so right and they ran outside of the IETF and IETF process with meetings everything because you know know, PGM, the, the, RMT working group didn't want to have them so it was pretty crazy the protocols that you're mentioning no more so they were done at the tail end as kind of, the most simple reference protocol you can think of from the building blocks we've defined right so yeah i'll look if some of that might be useful for posterity because it was pretty crazy Sure. Thanks I've got two clarification questions first um it currently says the the path and the source street is the determined by the unicast routing table and also known as the RPF path I do know that we've been asked to support different underlying topologies for this specifically, and we've gone to a lot of work to make that more correct correctly. I'm not sure if this is intended to be an observation that this feature isn't used much or it's just an oversight and it needs to be mentioned I think it came straight from the documents documents. Okay. Then I'll, yeah. Which, which one are you Thank you talking about? This is 3.3 where it says it's determined by the Unicast routing table so it's it's by default might be the unicast routing table but at most routing stacks, as far as I know, support using it different topology And I do know that we go to a lot of work to support different mixing modes as well, which might be a of extreme but we have been asked for this by customers so

00:26:00 - 00:28:00

Go to 00:26:00

So it's out there at least somewhere. Okay yeah i think that's just maybe just being a little too brief kind of of or you know it's not what many people do, but it's not like what you have to do do another thing is that in 3.5 the r pf is a security feature I'm not sure I would consider that it's a necessary functional feature, is a high higher requirement than a security feature Like if you remove a security, feature it still works in this case if you remove it it just breaks. So that's a little bit confusing to me and last thing there is this, there's already this section on data-driven state creation There's also this behavior-based on bandwidth or packet rate where you switch between the RPF and the SPF and I don't see that mentioned mentioned. I'll double check. I'm not sure. It should be there, but thank you well i'll pour this into mail as well. Okay, you. We'll include all that you. And Turles comments as well and we need to move on so thank you Okay Is there a name? Hey, can you hear me okay? Very loud. Yeah, just to speak a little quieter, which is usually not a good. I'm going to give you slide control. It's kind of a meeting, see if it'll work I'll talk software. Hopefully that helps Okay, we have three documents here, all kind of related and this is uh kind of our final review before we
00:28:00 - 00:30:00

Go to 00:28:00

do work with last call so i'm going to try to roll through as quick as I can All right, so the documents are divided three. The first one is a problem statement and requirements It's got the criteria for evaluating the proposed solution. The second is updates to IPD6 multicast group ID allocation And the third one is an algorithm for dynamically assigning IPv6 multicast addresses And this problem was first presented at IETF 114. So coming into the first one, the problem statement So this is an overview of the networks that we work with on marine networks networks, you have basically we're using multicast snooping to prevent high band with multi-gas data from overwhelming the links to low bandwidth sensors So when we start looking at how the multicast addresses play out on the network just a summary of what multicast addresses have IPV6 multicast addresses, you have 120-bit group ID ID and then 2426 talks about the first two octets are 33, 33 the last four four actets of the ethernet address are the last four octets with the IPV6 multi-gas address address. When we look at RC 3307, it says group IDs, less 100 to 30 bits long we'll generate a unique link local layer addresses within a given multicast scope. So the problem with this is that different nodes, different IPV6, link local addresses they can generate the same group, group ID, and accordingly the same Mac address address. The second problem that you can run into is that the switch adders table can have collisions this is talk about more in the uh the uh
00:30:00 - 00:32:00

Go to 00:30:00

but basically if you have an address table, oftentimes that is implemented as a a hash table with buckets if you've got four buckets in there and all of those buckets are full, you can no longer add addresses that use the same buckets. And so you need to have a way of basically saying yeah the multicast address you're trying to use here is not going to fit into your hash table and you need to pick a different one We looked at different solutions out there there the switch paying attention to that the IPV6 address Those switch vendors don't support that Source specific multicasts, not supported by the switch hardware Madcap, which is a use, it's a way to dynamically allocate addresses that relies on the central server and within our networks we don't any central points of failure and then ZMAP was an attempt to do a distributed mechanism for allocating addresses and that was never published So we decided to come up with something different So the solution is a decentralized zero configuration protocol for dynamically assigning multi multicass addersons. Going into the requirements here, if you listen to the document, we want any solution not to have a single point of failure it does not depend on user configuration coexist with other assignment protocols on the network network, such as MATCAP supporting operation on a single subnet does not require internet multiple applications on the same host, and then detects results the address collisions Desire features, which are not necessarily requirements but the things that we want They support operation across multiple stem nets does not require significant changes to existing standards common functionality functionality uses capabilities commonly provided to unprivileged applications I'm always depending

00:32:00 - 00:34:00

Go to 00:32:00

on configuration data, minimizes network traffic All right, so that ends the first one of the problem statement that we jumped into the updates to the dynamic I can be six multicath group IDs so just a refresher on the multi-gas address guidelines and 33 3307, the group ID is kind of divided up into different ranges if zero one to 3 FFF, that is coming at addresses those are are allocated by ion. Then you've got four zero through seven fff those are permanent also allocated by iona and then you have eight zero through FFFFF Those are dynamic addresses And then those are kind of describing describing two different mechanisms where you have a server allocation and a host allocation One problem we can see immediately with this is that both of those allocation methods the group IDs, overlap with each other other which can potentially be a problem if you're running multiple allocation methods on the same network The other problem is that it's the group IDs are conflicting with a solicited node multicast addresses that are currently being used. So basically if you're looking at the range FF00 through FFFFF you're seeing that there's a potential collision there with three different allocation mechanisms. The first one one being solicited a node, then the others are madcap and a host -based allocation Again, solicited note is required by any device that running multicast or IPV6 rather So our proposal to the that we want to do here is to set up a range of allocations in the table that is administered by

00:34:00 - 00:36:00

Go to 00:34:00

by Ianna. The first change would be an up, it would update matcap essentially to say that uh the range eight zero through eight 8 FFF, that's designated for Madcap assignment and that doesn't actually require an update to the Madcap just to the assigned range and then leave 9-0-0 through FEF-F-F that would be unassigned and then FF-0 through F-F-F-F-F covers the already assigned solicit node multicast addresses The idea being also that in short order we would assign two of the ranges here, the first one nine zero through nine FFF that would be for the zero configuration algorithm that we are going to discuss next. And then A-0 through A-F-F-F, that would be for the GAP protocol that is, I think, I don't think it's in last call yet, it would be very, very shortly And then, of course, this solicitor one one is there. And then there's a large block of unassigned addresses that could be used for future protocols And then the third one, this goes in the IPV6 based, the zero configuration assignment assignment. So this basically uses NBNS to publish records records and to some degree resolve conflicts in there So you have a record name, which you are generating a random group ID in that range that we discussed previously You use that to generate a link scope type PB66 multicast address and convert that to the MAC address And then you publish a new PTR record with a corresponding Mac address So if you can look at the example, down there, I've got your group ID is the first line the IPV6 link local link scope address is the second line

00:36:00 - 00:38:00

Go to 00:36:00

Third line would be your corresponding MAC address and the fourth line would be the name of the record that's published. And this is of course is kind of in line with what you would see with a reverse address address, IPV6 and IPV4 record record. The contents of the record would be in the name of the application combined with the host name So in the document we talked about application.example .local if the host name would be called example .local, this allows multiple applications to run on the same host. And then you also have a record called veto and that basically indicates that there's a problem with assigning the address such as an address table collision We engaged the DNSD working group because they have expertise in MDNS and to get comments from them. So we just were a couple changes that came out of our discussion with ESCO on that that. Basically pointed out the the comparison with DNSD. You're saying DNSD uses different DNS record types to facilitate service discovery, whereas what we're doing here is we're using a single record type, the PTR record um to facilitate address assignment and noting that the it may be used in combination with DNSD or by itself and that we're doing this with the expectation that existing in DNS implementations can be used We also clarified that address records may be published for the host name included in the record or they may not be if you're not doing DNSD. And then we also received approval to only require IPv6 in DNS over IPV6 The idea being that if

we're doing IPV6 IPV6 address assignment, IPV4, requiring IPv4, didn't seem necessary So the intent again is to to move forward with last call on this
00:38:00 - 00:40:00

Go to 00:38:00

and just want to open it up for comments and questions Yeah, let's dig here so if i got this right, you have one problem statement draft, and then there was one draft that is basically changing how Ianna, you know, that that range that is used for for group assignment, right, to split that into multiple ranges and then you go to two solution drafts right uh right The one I'm discussing one here for IPv6, and then Dino and Mike have one one called Gap it's still job on there so So one thing I think we need to I assume that you believe there's a need for both of those solutions, right? the V6 one and the gap one Yeah, I think they can both satisfy different um you know, different customers or both of them are other ones going to be better at something than the other one right so i think may may need to have some text in those solution draft drafts kind of explaining a bit, you know know, what their purposes or why this is is one solution is better for a certain you know use case or something like that it's going to hard for customers to or not customers but implementers or vendors maybe to know which one they should implement Correct. Sure Um Yeah, we can Well, I feel like, I don't know in a way that the very first draft is really the uh the problem statement draft that's right and see
00:40:00 - 00:42:00

Go to 00:40:00

if everyone is happy with a problem statement and if it covers what you want to solve and in that case case we can do working group plus call on a problem statement But for the solutions, think we kind of need need to first narrow down a problem statement and see that that you know the solutions actually fit within a problem statement One of the things included in the zero conf. I don't remember this included in gap either is is but an evaluation of how well the solution fits with the criteria and the problem statement All right So, yeah, one thing, also yes it's great if you can can, you and Dino can also have another look at the other problem statement and see if you if it covers well what you actually have solved Yeah, so so I would say the problem statement draft can maybe go soon and the group ID draft, that doesn't really depend on the solutions right so that can also kind of go before the solution drafts right I agree. I think the biggest um advantage to that is it points out the conflict with this solicited node addresses So it's an existing conflict with server-based allocation and solicited node node. Oh, okay let's start with a problem statement, maybe and check should we do a poll or? Yeah. Okay. Okay okay, yeah, I think we have to take it to the the list But we'll check on the list what people think about their problems statement and see if that's something we can start with at least
00:42:00 - 00:44:00

Go to 00:42:00

least I got it eight minutes, Okay. in the last last IETF meeting, we presented on some uh potential issues with IP, or sorry, with multicast snooping Since then, we have started working with some folks from Dunapur who actually read a document to address the issue issue. So the document just kind of having a status update on that, we are going to list it as an update to RFC 4541, which is the multi-gas snooping informational RFC. We're including both clear clarifications to RFC 4541, and then also optimization is not considered to that original And then also we're going to do an update to R.C. 30810, NLDB2, there'll be more updates on that. The document currently contains a lot of to-dos jump into what the progress that we have made on there so this is a refresher from RFC 121 There was a kind of a gap in the 40 rules that are listed there where it talks about what you you should do with multicass routers and what you should do with hosts but it does not talk about other multicast meeting switches on the network there. So we do discuss that in the new draft document and then i'll also that there's this issue with a a non-optimal distribution of the multicast data So if you're looking at the red stream from switch one to switch two, that stream doesn't actually need to go up to the query, but it has to do the requirement that the packets must be forwarded on all rod reports there's also a an issue in that the membership reports are only forwarded up to the career and not down to other switches on the and that would prevent the

00:44:00 - 00:46:00

Go to 00:44:00

the membership reports from the host getting back to the other host So ultimately, in order to solve these issues, you need to have a better use of reports or the better use of queries And we looked at it and the reports would require additional protocols to address compatibility issues between IGMP V2 V3 and MLD V1 and V2 So what we're doing is electing to go with query messages The problem with that, the immediate question is, are IGMP? VA and V2 queries compatible with each other? And you'd have to look at the checksum for that and how the checksum is defined The checksum is basically, or they RC defines the check sum is needing to take place of the whole IG &P message. So if you have an IG&P V2 host, getting a V3 query, and it goes to calculate the check on it would look and calculate over the whole length not just the length that it understands. Similar to thing with the MLDB1 and V2 again it would calculate that checks on over the length of the entire ICMP message. So our conclusion there is that the new newer version queries are going to be compatible with older version hosts, most likely if the hosts are implemented correctly, if they're not implemented correctly, the solution would be to update the hosts to do that correctly we also there's an issue with a proxy queries that are described in 4541 It basically describes a query with the source address of all zeros That is not used for query election, but it doesn't address the same problem with IPV6 There's three documents listed there that all

prohibit using the all zero's IPV6 source address So while we could potentially update those documents to allow for that, it would also
00:46:00 - 00:48:02

Go to 00:46:00

go against existing implementations And so rather than kind of fight the established precedent we come up with a new mechanism here The idea is that we would take one of the reserved bits in the query message and reallocate it as a a p vint if p is zero which is what it currently would be used, are sent out with with, that is in the that it is used in query election and if it is one that it to not use it in query or election. We also noted that the IGPV3, the corresponding bit in IGMP V3, is left on modified modified. We basically divide the changes down into control plane and data plane basically multicast taping switch is a proxy query that is sending out proxy queries to all of the other ports there periodically then there's new timers that are analogous to the query and enroll and the IGNP and MLD documents for that And then also talks about how are you receive queries You forward them to adjacent nodes only if there are not non-proxy queries. If they are, you don't don't And then you also reply with a report that's containing all the groups in the membership table except the group where the port, the prairie was received on it's the only member report so that's a complicated but basically saying you don't want to tell somebody sending you a report that you also have that group unless somebody else is interested in it it and then we do have a lot of to-does listed to what happens when you receive reports, transmitting reports reports, then also removing groups from the membership table. How do you update the rest of the network? to reflect that? that? On the data plane, we talk about we divided into a non-roditable multicast traffic so for that you would only forward a rotable multi-cast traffic to the multi-gas router port. So if you know, know FFO2, for example, it's non-routable multi-cast
00:48:02 - 00:50:00

Go to 00:48:02

We lost you, Nate Nate. can hear you now now. Okay Okay. We also noted the potential for add-bous collisions and then for routable multicast traffic specifically for ASM, the router will handle improved messages those groups back towards a multicastery source and for SSM the PIN sends a request to receive the traffic there, so that's already different different. So at this point, what we're looking at is comments on what we're just discussing here, any sort of comments on what we've laid out so far Specifically, would say, that the use or allocation of the reserve bid there If we could get confirmation, we're OK heading in that direction And then we plan to continue refining the document All right, we got one minute Any questions for Nate? I think it's just about 2 a.m him, so if not, he's going to go to bed Right. Thank you, Nate. Appreciate it. Thank you Right, David Thank you very much for your time
00:50:00 - 00:52:00

Go to 00:50:00

Today I'm bringing a use case. A problem statement to check if Multicase is a solution that we can work together So distributed consensus is the problem and it's all about shared state agreement currently applications like file storage, online gaming, and cryptocurrencies try to maintain this share state Any change of this state needs consensus among distributed participants and one key operation that assures these distributed consensus is the majority rule that had been outlined by phone Newman in the 50s and in this context of distributed consensus systems, there is this new sort of type of permission systems remove the central coordination through an operation that we call the iterative decision. So what is this iterative diffusion is about maintaining a pool of unicast relations that we call the contraplane and these these unicast relations are maintained at the global scale And on top of that, these pools are randomized and constantly being replenished So people started to build these to disseminate state changes that are the data plane and with this operation the peers cannot really control what they receive because it's only a local validation and they keep disseminating this state We call this operation an uncoordinated operation and these are in fact actual common deployments deployments um that affect um the control plane overhead so these constant replenishment and but randomization that has been designed to go with churn and enrichability cause contra pain

00:52:00 - 00:54:00

Go to 00:52:00

overhead in terms of the number of bytes that one needs to establish and maintain these unicast relations This is done to to maintain the this data plane and this data plane keeps being floated in between the pools and the peers cannot really control when they already received. So we call this data plane overhead and we measure this in that in terms of the number of bytes that are being duplicated in the system system. These replenishment and this duplication delay um also causes that the agreement gets delayed the consensus latency in terms of the time that these systems need to to reach this state agreement gets longer and on top of that that, all of these issues, peers can cannot all can neither judge the dissemination, hence they cannot judge the agreement of the consensus we call this the consensus finality and it's just an indicator for determination of the consensus. But it's interesting here is that this deployments actually are running in data centers, so hyperscalers, cloud service providers and ISPs. And the scale of this is not that huge as the real internet So we think could Maltica has come here for better to move? move from this picture of these uncoordinated operations? where peers are establishing relations among the at random and constantly replenish to a more structured way like this figure where we call it a state replication points try to orchestrate the peer to SRP communication so peers do not need to establish this end-to-end relation at global scale, but they only need to maintain a contraplane to the SRP. This

00:54:00 - 00:56:00

Go to 00:54:00

is another late dedicated SRP that maintains these two types of relations peer to SRP and the SRP to SRP We have been comparing these iterative diffusion towards an Stroman architecture that is an Nobel-Ale and we align the benefits that this can have in terms of the number of bytes that can be saved for establishing the relations. So we arrived to this comparison in terms of the country plane can save around 700 times the message is that one required to establish two relations and three times fewer bytes in terms of maintaining this relation once the country plane is established On top of that, this contraplane does not duplicate the state that is being fluided. So it means around 20 megabytes per less traffic per transaction I forgot to mention that we compared these iterative diffusion from the Ethereum system that is currently deployed at global scale with around 72k peers so that are deployed in around 10 ISPs and from there comparison is coming from from. And this this country plane establishment and the data plane packet forwarding results in three main stages of dissemination so it results in a consensus latency being four times times faster than the one that people realize in iterative fusion And furthermore, these SRPs warranted the majority surplus So the peers know when the consensus is achieved I'm happy to take questions in regards of this work I've been working on the measurement, modeling and there are are some papers around and I know I tried to

00:56:00 - 00:58:00

Go to 00:56:00

summarize everything but this is the the idea of the happy to take questions questions um yeah so one question so all the peers are more or less equal like all of them may be sending data to all the other peers. Yes. That's the main idea of uh yeah decentralized participation participation right and do you envision there being any need to make any changes to any flavor of PIM or do you think? you would likely use what we have today? and just maybe introduce it to some sort of of a distributed consensus system to to improve their latency. Actually, this Drummond already outlines a small feature that probably this multicas-enabled pair ISP could do we call it it the counting of peers that could improve on the forwarding of the SRPs SRPs. That's a small feature that we envision but we actually came to to here to the group to question that if that's possible or not or do you see for their solutions or for problems? Thomas Schmidt from Hamburg One question, mean, at least to my knowledge, of textbook distributed systems, there has been I believe, 2017 or so an improvement of Paxos that is called Raft and ISDI. That works with multicast, right? So I wonder how this compares um yep um so there are kind of a two families of these distributed consensus. The Paxos fills into the family of the deterministic ones and this ones are the the permission permissioners ones. Paxos is built in an entire closed environment

00:58:00 - 01:00:00

Go to 00:58:00

these guys build among environments and anyone can join the consensus impacts you you need to select a leader from a a known identified domain and here I that the leader is set selected randomly from many from not many from 10 ISPs so to say Man, Kamala, Ms.ras, systems. So if everyone is talking to every other, so PIMBider, may be a better choice than just going PIM BIM bi -directional where since everyone is talking to everyone else else, so PIM bi-directional will be better choice. Or the variation of pym i think mike was asking if different variation must be used so I think bi-directional will make more sense for these type of use cases yeah No tip. Thank you you yeah another thing maybe is whether you can expect all of these to have native multicasts or if you need to handle some case where some have native and some need to use an overlay and how to kind of bridge the gap between overlay and native native. Mm-hmm. Hi, I'm Dr. Rossen. I'm wondering to call us maybe a few clarification to the question that we're asked The overlay, going to the question of overlay, the SRPs are overlays. We do envision that the local breakouts would just use native multicast capabilities that's that's that's that's that's that's the advantage of it. And one of the extensions that, um that David mentioned that could be um done and we are still investing in that if it it's actually worth in terms of overhead is if you do disseminate you could just simply choose in your local domain to flood, hence you send it to the entire group, or you would like to send it to a selected group, which then, course, has less data playing overhead because you only need to reach, let's say

01:00:00 - 01:02:02

Go to 01:00:00

another 1,000 users in order to reach consensus Whether or not that's even worth in terms of data plan overhead versus changes to be made in the system, that's something to look in evaluation but that would require change because you're not sending it to all, you're sending it to a subgroup right? But it may just not be worth doing that And you just flood it to whatever it's locally, locally One of the parts that's really important which motivated us to look at the multicast part, and that's very possibly a little bit undervalued in the community is the last point on the slide here that you can see see that multi-cast really gives you consensus finality And in order to highlight, what that means, and it goes to systems, you're uncoordinated sending to everybody the only way to judge whether or not you are done disseminating is downloading the result of your consensus. So in the blockchain, that means if you want to see if your transaction made it into the blockchain, have to download the entire blockchain to check if you're envied it that's a huge overhead given that the blockchain is usually quite large. The ledger right in multi-cast the operation gives you finality If you send it, it's being sent and you're done, You can do a little bit and be evaluating this at the moment as well a little bit of an overhead. So instead of sending it to the majority you send it to the majority plus 10% because it's not unreliable service but But the finality judgment is a huge improvement which is one of the reasons why we're turning to the multicast community to say maybe you want to do it by a multi because it allows us to build systems that allow you to realize that your operation is finished without downloading the result, which in crypto is a real issue thank you Thank you. Thank you Welcome to do some contribution for beer use case dropped because we'd like to move it forward and if before us you can add the use case in it it will be very great

01:02:02 - 01:04:00

Go to 01:02:02

Yeah, thanks thanks. Thanks. Thanks, Mark. Thank you Yes, ma'am Can you hear me? Yes Okay thanks I will be presenting experiences with list multicast deployments, you know know this essentially we have the 68 RFC that is now right being moved to standards tract so there is the 68 BIS document being done done at LISP. Am I audible? Hello Make a stigma go ahead? Yeah, can hear you, and you have control the slide slides okay okay thanks so, sorry, sorry so basically the 68 31 this draft is being done at the list working group and and you know in order to support uh advancing that to the WGLC and and essentially 6831 becoming a full you know, I mean, standards track of ETF, RFC, we, we some requests from the list working group to have the LISP use cases, multicast use cases documented, and so this draft is an issue support of that This is primarily an informational draft but you know being done particularly a separate draft, essentially to highlight the different flavors of multicaster deployments we have at least so that's why we have a separate document and not a section in the the in the 6831 so with respect to the scope i think you know we have you

01:04:00 - 01:06:00

Go to 01:04:00

know from the technical perspective i think we we cover the layer to BUM, that is the broadcast unicast and multicast you know, service that can be provided with the multicast underlay or actually let me take a step back i think we should just say that, you know, we support deployments of various you know types of uh underlays that is both the Unicast and the multicast underlay in 6831 with both IP4 and IP IPV6 you know kind of combination you know quite leads to some permutation combinations in the new overlay and the underlay and then you know in the respect to the services, we have the L2 layer to BUM you know essentially for flooding things like ARP and, you know, layer to unknown unicast you would use a multicast underlayed to essentially take care of the replication in the multicast underline. Now, to the layer three, we have, you know, of course, EASM, SSM and BIDER, where, you know, the overlay clients can send ASM SSM or BIDER and then these lists, you know, sorry multicass packets get mapped into the LISP underlates that get built you know, with all the four combinations that I mentioned, you know, underlie being unit IPV4, IPV4, V6, Unicast, Multicast, know, and and how you can you know essentially have them just because in a in in many real world deployments we have some quite existing scenarios also as well well right now just you know in terms of you know, the types of networks that we have seen of course you know the reason why that is being even highlighted is essentially some of these requirements for example, you know, the networks for build for, you know, trade shows are, you know, expose are probably slightly different from the networks that are built for hospitals right and and you know each of them bring their own you know a little bit of of uh you know customization that is needed and we have have a few lessons learned i'm not going to cover

01:06:00 - 01:08:00

Go to 01:06:00

them in the slides but i think some of them are described the documents you know for example short-lived flows uh you know a announcement like you know PA systems in the the airports you know are probably slightly different from compared to long-lived flows you know, there are some optimizations that we have done and we try to cover them again, you know, nothing you know, quite different but just trying to you know kind of highlight you cases so that, you know, if can kind of come to the same page okay so what is not in scope i think what is not covered is definitely you know for example things like BER, we're not going to cover or or even for that material three unicast, you know, between the sites right? You know, that is not covered because this is purely restricted to 68 31 which is essentially the multicast document now um of course non-lisp networks are not being covered here you know any kind of of, you know mvp based networking or you know any other IP multicast is not being covered here last but not the least we are not trying to cover external it, is the out -leaking part in this document, you know, there is an external implementation that we have you know done of course course you know probably not much deployed as the regular one, but still we have some some you know experience there but I think we'll try to keep it into a separate document just for the process of of, you know, the scope defined okay so now uh you know in terms of using pimo or pym i think you know the reason why, you know, we even did the 6831, I believe, you know essentially was to reuse the know PIM uh cold base and the PIM you know deployments that are very well through both from the operators and, you know other people alike you know it provides a very convenient method for us to kind of get the multicast service running you know, because just the reason why I bring this point specifically is that there is a signal free RFC just I mean also being

01:08:00 - 01:10:00

Go to 01:08:00

improved from or rather upgraded from experimental to standards track which is the native list method of doing things you know essentially having the MS participate in the in the flow right just for the you know, general awareness the MSMR is essentially the central entity in LISP which controls the distribution of of LISP overly reachability information in the Unicassward and there is a way to do that for multicast as well But that, you know, probably did not, you know, kind of, again, typical engineering approach where, you know, you try to do things iteratively where you try to reuse whatever you have and then try to build on it and then come you know try to take up the more complicated stuff later think that was one of the primary motivations of you know using the more time i think think you know stick is here and he can probably correct me from you know my understanding is not correct but then the PIM comes up with the cost of, you know, essentially a PIM hello and neighborships being cost, you know, considered chatty, Especially if you have a large number of lists sites in the in the network you know I mean we we've seen this sites in the order of few hundreds for sure, right? So then PIM becomes very quite chatty and before we had to do some simple you know, optimizations where the pin neighbors were the neighborhood was was kept in a passive mode where you know the hellos would really not be set. But the PIM neighbors, neighbors would still come up, you know in a kind of, you know in a pseudo connected state But there, the reachability can always be gotten through Lisp and we kind of got away with you know, really having the Pimhallows exchange. There is a bit of disadvantage there in terms of, you know, heterogeneity being lost, you know, in a reasonably controlled entranum take an enterprise, where the home homogeneity is kind of ensured still kind of you know helps us you know keep the implementation simple right? And last but not release film is also very well understood so so you know that's one of the helpful factors. I mean, point I forgot to mention is that, you know, a simple optimization was done in the underlay where we only chose to use s s m and i specifically mentioned this because

01:10:00 - 01:12:00

Go to 01:10:00

this takes away a lot of complexity in managing you know one layer of r p and one layer of you know, all the complexity that comes along with synchronizing skates between the RPs so you know the reason why you could just simply get away with the SS7 in the underlay is that any strategy or HG is just being being forwarded to a particular exterior node depending upon where the RP are the source is hosted and that information is anyway being on claim from the Lisbon rotate so we could essentially make that optimization very conveniently and and simplify the solution right Now, you know, a typical campus deployment again you know reason why I bring up this is you know there is a terminology that probably needs some work, you know, how to kind of get it probably that is a question for the list working group, I know I'm, you know, definitely comments from here are welcome as well well uh you know the traditionally the list terminology that is defined by the base Lisp RFCs are essentially every Lisp XTR is considered a LISP site Now, you know, in a typical campus, you could have have, for example, know, a multi-site corporate building or you know multi-site factory floor right multi -building factory floor, I'm sorry, right? essentially you could have one campus network where there are many lists sites for example take two floors of a building right and each floor is just you know considered as one campus so i call the word campus here if it could could be your floor like building floor a lot large building floor right now what we typically do is for the purposes of administrative convenience we kind of of classify, you know, and group them and manage them into, you know, in this in this diagram I refer to as a campus or you know some group of sites, right? interconnect them with with under you know the global internet depending upon if they are very far away connected or even you know even even uh another list net where if they are reasonably very close closely connected then you have some L2 connections between them, you L2 connectivity between them them, you have, you know, at least what do you call, a site of sites right and and and

01:12:00 - 01:14:00

Go to 01:12:00

and kind of interconnect them so now the question from terminology perspective is what we really call these campuses you know we cannot call the them science because they're you know uh it's already defined as as, as, as, as, very close to an XTR right and and so there is a little bit of a a, you know, know, terminology required that is is needed here, not just any for any specification because the specification part is all done, but just to document it right when when you get to document it you know there is a little bit of inconvenience on how do you describe it so that that's the, you know, one open item that at least I have in the drive I have shared this with Dino and I think he has given some comments and I've been working with him offline and i try to close the gap with you know i have one word that came up was to have the word domain, you know, being used used, right? Then, you know, asked me the question, how do you really define a domain? So, you know, we have to figure out a way on how to kind of of define this but that is just for the purposes of essentially making ourselves clear but in terms of the actual specification of of what needs to be the procedures that needs to be attached or adapted i'm sorry in signaling these flows setting up or tearing down these flows or maintaining this close there are all defined and quite honestly, you know, this is one RFC where you know we have a good amount of implementation with the last large, you know, you know, deployed code base, you know, in terms of variations, know, different kinds of networks and you know happy to you kind of share the experience so that, you people who are evaluating this for example for their networks right i think the reason why you know the sales pitch i think, as, as, it was referred to an earlier draft why do we need this document essentially for a I would say, two kinds of audiences? at least two kinds of audiences. One, primarily for you know from the perspective of somebody who is evaluating list for their network right you know what capabilities is this list multicast offer me compared to for example something like bgp vp or some other, some other technology. I think somebody can really read through and kind of make sure sure that you know what they're looking for can be achieved And second thing is essentially for other you

01:14:00 - 01:16:00

Go to 01:14:00

know other working groups essentially best are people who are kind of in doing a cross, what do you call, analysis a comparative analysis of you know what are the pros and cons of course you know everything is got its own advantage or disadvantage and kind of of know kind of comparing them it would really help as well and and last but not the least, you know, even from the objective of making the success, 68 31 best to a full, standard strike RFC, I think this would be a good collateral to have. So that's all I have from my side side. I'll stop here for questions and yeah, other comments Yeah, yeah we need to move on but one quick question or comment um We published just recently what they call pim light as an an RFC-9739, and that discusses how to do PIM without PIM neighbors and and how you still can send joins and so on. I think it it closely matches what this done with this multicasts so maybe good if you can look at that rfc and see how it fits and maybe incorporate that in your slides somehow yeah yeah definitely definitely thank you thank you for for that comment thank you Thank you you. Thank you for us out-statement's reputation with S.R. B6. Is this on right? Something like that Something like that Yeah. Let me give you. I remember the buswords. don't the exact yet. All right Hold on right. Second round Okay, what we would like to propose is to extend the SRV6 architecture with a

01:16:00 - 01:18:00

Go to 01:16:00

stateless multicast replication and delivery rights. same goal as SRV6 Unicast, right? So, but of course for us we don't want to have tree state we uh in the network, state this replication in the network and like in the SRV6 architecture as well also with or without steering steering is for traffic engineering non-steering is just you know best possible delivery and of course we want to follow the principles of RFC 80200, which was very painful for SRV-6 unicas. So why should we have it any easier? and only change them if it's really necessary? right? So which basically means that in an SRB6 network, network, the forwarding that customers want to have is through an IPv6 routing extension header, which is also what done in Unicast. So roughly speaking, this document is a was a quick know, point into the work. I think we would also follow the way that the work was done for SFB6 Unica. So architecture solution document, for UNICAST was done in spring and for multicast the i think springs had something about multicast. What is Multicast? Go to PIM and that would be defining and using all this wonderful sit terminology, which I haven't put into this draft yet, right? So I'll probably, you know, rewrite this into more formal language for the next IETF So that's defining the architecture, the solution how that would look like Right now, spring has really only done very simple things for multicasts, which are the tree set, for example, RFC 9524. So that's really only good for very few let's call it IPTV trees or so I'm always calling that multicast PVC's x25 just with SRV6 now so it's a little bit bit so it's certainly not what we need for for scaling and dynamic services

01:18:00 - 01:20:00

Go to 01:18:00

And in PIM we also have a few of these stateful solutions solutions right so but the whole goal is to really have the state less solution in the same way as also also unicast habit. The actual encapsulation, of course, would have to be outsourced to six men men, tentatively, you know, multicast routing header That's exactly where the 8200 rules apply aligned with the SRV6 architecture also trying to reuse and apply as much of the SRV6 extension header functionality and definition as possible And so the whole relationship of course is is you first have from the solution working group this case, PIM, the solution argument document, and with that you know to show the interest of the community, you go to six men and let them as the expert expert review and drive the extension head of work So here is kind of the simple technical proposal outline. So you see down there there, typical IPV6 extension header that is pretty much you know taken from the SRV6 SRV-6 SRH RFC. There are two main building blocks that are replacing the sequence of IPV6 addresses The first one is actually the last segment, which is what the receiver will interpret and that's actually carrying the ipv6 multicast address or if it's unicast and any new group semantic sit that we want. The second block is then one out of hopefully as few as possible different encoding of the stateless delivery and then followed by the standard SRH optional type length vector TLV objects which are not interpreted hop by hop hop by every router. The first two ones, the MSCR and the the MRH steering stuff that is interpreted by every hop so that is important for the hardware. The TLVs are only important They're also sometimes done by the hardware they're on the exit router

01:20:00 - 01:22:00

Go to 01:20:00

right? And so I've put the details in there. So for example, routing type indicates, which routing extension header it is So they can be up to 256. For Unicast, actually are by now four allocated If you've been following SRV6, you're say, well, wait a second, it's only three Yeah, that's true. SRV6 has only, only, quote, defined three the original SRH and two compressed variants, but of course, before SRV6 was done, all the the same work was already done by the IoT world, and they actually have the first Unicast extension header so we'll also try to see if or how we can make the IoT world for the multicast happy If you remember several years, ago we had these discussions in pym or beer in terms of that they were also interested in stateless multicast and the lower networks so maybe it ends up in the same way as for Unicast that that would be yet another routing type because they have some other metadata so we'll see for that. So far I haven't checked checked so then what would be in that subtype a blog So this is basically the most simple subtype block right so we should shouldn't reinvent the world. should keep everything that is proven to work well and the beer architecture 8279, defines a brilliant simple destination forwarding stateless with a bit string. only problem that

IPv6 networks have is with the hop-by-hop forwarding with the 80296 beer header, which is exactly what this IPv6 extension header replaces And that, of course, also makes it a lot more simple if you see the header below we only need the architecture elements, BSL, SD, SDSI, the entropy, and the bit string. And then all these other field version, BFR-I-D,-C-S, ttl d s cp proto that are in the beer header, that was a little bit bit of a consortium header, right? so everything was put in there trying to make everybody happy and we don't need them these things are in the
01:22:00 - 01:24:00

Go to 01:22:00
IPV6 base header that we're adding afterwards. So this is the optimized version so to speak for IPV6 networks of course, forwarding hop by hop is by rewriting the IPV6 destination address as a unicast right so this this forward part is all the same as in as a Unicast except of course your extracting to whom to send the next copy from the bit string or and that is the other proposal where I don't have the structure where instead of a single bit string we're having a compressed structure representing a deliberatory and that's a very long story So there was a lot of work also already done done in in um in research So we had the P4 Tofino implementation for that. We presented most of that work in in the beer working group just to draw the comparison there and because we also did beer there, which doesn't really work well And so the steering option has two main points, right? one is traffic engineering on top top of it right it's the steering the explicit steering, so you can assign resources to that the second one is that you can actually also make that work much better for larger networks so we're having different drafts right now that we need to merge and come up with a common proposed option and I think one of the important things is trying to figure out if we can come up with the solution solution that doesn't only work in vendor ASIC, but also in something like Tofino for researchers, because I think one of the big problems we always had is that we need to have more research researchers being able to adopt our technology verify them independently and allow extensions from that space Yeah, the MSCR segment, the why do we need an IPV6 address? in there, multicast destination address because we want to have a solution like Unicast where i don't need multiple layers to represent that
01:24:00 - 01:26:00

Go to 01:24:00
but it's just end-to-end IPv6 multicast just without state so I need to be able to have in the packet in the same IP IPV6 header in the extension part that address it's the first one so that is at a fixed offset and that also means that yeah you know need to look a little bit further down but it will be possible with a single layer IPV6 header to again reintroduce or keep all the ancillary forwarding plane functions that are operating on the source and destination addresses of an IP header So, and I was just trying to to list the ones that quickly came to mind there's IP fix for

accounting, and other monitoring purposes. There, of course, a lot of ACLs against source and destinations really various of architectures based on these address ranges bent with filtering policing setting DSCP based on the group address, VRF mapping based on group address range, So there were a lot of things done and by having that address in the header and being that real IPV6 multicast allows to reuse all of these things that is also one of the key differences over beer, which puts these things into different layers right so beer is it layer 2.5, and then you need to put inside another IPV6 header which makes all these functionalities now a two-layer and a layer violation problem problem. Yeah, so, and if that destination, address isn't a multicast address, then we can come up with new sit semantics, right? We already have two semantics for group addresses SSM and ASM so now we even have the full IPV6 address space open and, you know, can use in the same way as SRV6 Unicast the last 64-bit for interesting parameters programming and and so on. So that's going to be a lot of fun So summary, right? this is operationally, the most preferable option for SRV6 network operators because it really follows exactly all

01:26:00 - 01:28:00

Go to 01:26:00

the SRV6 Unicast architecture designs in the forwarding. It's a single layer network solution for IPV6 with the MRH header, so there is no additional overlay encapsulation to forward IPV6 multicast and there is no additional per hop tunnel encapsulation to basically make the packets on the wire look like IPv6 as another solution Support for all the ancillary IP multicast forward forwarding plane functions or IP header functions because they're all, you know, unicast and multicast multicast and and uh define yeah so basically the simple form flat bit string. Let's use the B architecture that works well is and we don't want to lose the investment in hardware that can do the replication based on the bit string there's just a slightly different header field, but all the bit string, all the important parts would be in the same way as also in other other other beer encapsulations. And then the steering solution based on TBD, that's a longer discussion Yeah, and that's it Any questions? Overwhelmed You've got three minutes Matt Women, Nokia Yeah, I overwhelmed uh but just uh just follow this up, basically what you're saying is that we're going to do PIM with some kind of a beerhead header to introduce SRV6 into multicast. Did they get this right? No, mean, there is no pin in this. This is just the forwarding plane, right? obviously for the overall solution, we're also going to have to say what the control plane is we're having which is going to be an interesting discussion because most of the SRB6 unicast control plane that we've defined so far is purely for everything where

01:28:00 - 01:30:02

Go to 01:28:00

the senders and receivers are routers that's all through the IGP. So we'll have to see what we want to do as the control plane for that, can simply follow the IGP stuff, we can can also simply take the thing that we think is the best SRV6 network solution for, you know signaling the receiver to the sender so that the bit string can be built, right and we have BGP for that we have PIM for that right so we'll need to ask the SRV6 people what in SRV6 networks is your preferred option? that I would think, I mean, I for a decade tried to persuade network operators that PIM is great then they, in the meantime, read it everything in BGP, so I would expect that they prefer the VGP option to signal the information but But those control plane aspects, of course, we would do aligned with, you know, what's bringing once okay so so just to be clear um like we tried to do IGP and Treesit. One of the issues that we start facing is that if you just want to do like a three-seat S -6 with IGP multicast is too chatty, So that didn't go too well Yeah, is no, there is nothing here in intermediate routers, there is no, trace it. There is no state in any of the intermediate routers because it's just you know ultimately beer forward forwarding through the SRV6 extension header but then why not just do whatever the heck we are doing in B right now? now? And that's the long discussion about, you know, what do IPVC so I have appendix slides that is trying to show that that you know we went through the whole experience experience once already, right? so we custom who wanted to have multicars and the layer three VPN services came to us and said, hey, we've got MPLS VPN networks with layer 3 VPN service we need multicust. Customers are asking for it We gave them, you know, Draft Rosen with IPVL multicast forwarding right and uh

01:30:02 - 01:32:00

Go to 01:30:02

Okay, you can deploy it now right? That was very quick Is everything working for you? Yeah, is working, we hate it, right? It's not using MPLS forwarding, right? So what difference does it make to you? No, well it's not our operational model right? So basically, if you need to use something now, yes, you're going to pick up some, you know, a strange beer header that is optimized for MPLS in your IPV6 network you may have another tunnel encapsulation so you're ending up with three headers right, and you're grumbling going to accept it. don't think it's the best way to sell the beer architecture right so as soon as the customers are going to have more time, and that's what we're seeing with customers in China, right? There's saying we love the beer idea but we hate that it's not aligned with our SRV6 network architecture, So, you know, up to the point where, you know, you're not going to deliver anything better and we make give up on the idea of using multicars at all in our network, we're going to wait it out until you come up with a better solution This is what this extension header is meant to do It's the operational thing, So we had a very strange process which didn't take operational requirements or the you know experience we've made in the past with mdpn into account right there were so many customers that came back and then said please do proper and MPLS network solution with an MPLS forwarding plane, right, instead of going IP4, same story just backward. Okay, I realize that. just have one last comment I guess we need to understand that the problem that we are trying to solve because right now we have a lot of flavor of stuff and it's becoming very confusing when you start putting too many flavors, pistachio, chocolate, chocolate, strawberry, nobody knows what to deploy and then everything dies anyway. So I'm all like i think this is great but let's come up with exactly what we are trying to see solve and then we obviously will so i forward. I think it's very simple, right it's it's basically taking the great beer architecture idea and the basic forwarding, giving it the right

01:32:00 - 01:34:00

Go to 01:32:00

flavor of encapsulation to get rid of all the state in the network and then whenever we're running into the networks where either any traffic engineering, right? Let's not do BRTE because that's I did the RFC, right? So I know all the pain with it, right? But let's do the more, you know, structured encoded compressed trees, right? So, and then if MPLS, people come and say, now you did all this cool things for SIPv6 networks, right? What are we getting? You know, then we can of course say, well, well, you can put that stuff, of course, also into a beer header for MPLS networks, right? So it's got to be really quick, We've got to move on. Sure from Juniper. I just want to confirm the my understanding here. So in the past, we have talked about three options. One is that in the PIM working group where I believe it was why more more proposed to you to encode a tree or subtree in the MRH. That's one. And then in the beer working group we'll talk about two things. One is your RBS staff and then there's a beer b6 so it seems that's all these three things right now are all put into this document The BRV6 stuff is basically repackaged into this one as one of the options is that correct are you fine taking it offline I mean I'm happy to to also answer on the mailing list, but let's it offline because it's so long. Thank you so much Luis Oh, maybe slide Thank you. Hello, everyone This is Luis from Telefonica. will present this document on behalf of my co-authors, Hongi and Shufeng. And the trouble is about the

01:34:00 - 01:36:00

Go to 01:34:00

proposal of a new Jandata model, well, an extension of the jam data model for ITMP, MLD proxies in order to support the multipath approaching in the in the in in proxies so basically this document just as a simple recap up or recap sorry this is part of a work that there has been known initially with with Hitoshi, with the idea of providing the support of multipath in the proxies basically essentially to have the possibility of activating more than one upstream interface on the proxies So the framework document, let's call it like that, is the adopted document, IETFP multipath, IGMP, MLD proxy document but basically in that document we propose different ways of configuring the upstream interfaces interfaces a statically in a static way or either in an automatic way so for the automatic configuration, the could be two options, one based on signaling that would require some stations to the IGMP, MLD messages, and the idea will be to leverage on RBC 9279 for that and this is documented in the the Draft-Contreras PIM multi-if config. And the other option would be the controller based configuration So following a kind of sdn like approach and for that trying to i mean on data models for having that configuration, essentially. And this is the topic of this doc so yeah I will go through the through it. So what we are proposing is, as I said, to augment the data model for the proxies and basically incorporating in that data model the necessary attributes that could help us to configure from the SDN controller So we have proposed the extension

01:36:00 - 01:38:00

Go to 01:36:00

of augmentation now for IGMP, the idea would be to continue with MLD for sure And essentially what we do here is to propose the augmentation for those attributes for instance, default upstream interface the interface take over in case of some failure what would be the group to be subscribed in which which interface, and so on so far as next steps we will like to first to collect feedback and interest from the working group we clearly will work on the MLD part in order to make the that complete we need to analyze further security considerations so we depart from the security considerations in the young data model, we need to understand what could be the implications of implementing that and then we would like to move forward together with the framework document the signaling document and also adding this possibility of configuring through the SDN control controller. Yeah, so So whatever comment that we could receive today as well will be incorporated in the document and the idea is to prepare a new version for next ATF in Madrid and yeah I would like to have comments from the work group, any Probably we could go for adoption adoption for the framework document probably there's the one for signaling and the one for this was the controller base let's say would require more work at least for this for sure because this version zero but they think but for the draft IETF in multipath probably we could go for the for okay you mean sorry, I'm a little bit tired. You mean adoption of the working group last call? mean, no? Adoption of this yang draft. Oh, the general well I would wait for having at least the MLD part covered in Madrid yeah sorry a bit tired yet oh yeah oh it looks useful to have a young solution for this, at least to me

01:38:00 - 01:40:00

Go to 01:38:00

Okay, thank you thank you I This second document is about the proposition of extensions for signal in echo mode so that either the subscriber or a device on behalf of the subscriber can see signal in the echo mode mode at the time of receiving the content So the rationale for this is the following, so the video is basically the major contributor of traffic to to the Peritos network And there are even multicass is benefits so much much in at the time of reducing the traffic on the network, there are some aspects that could be improved or could be optimized let's say. For instance, the fact that in many cases there are the same content is distributed with different resolutions so basically we are replicating the content that we are replicating with different bandwidth associated to the resolution that is applied to the video and also the fact that in many cases as well there is distribution trees are built let's say up to the very axis with the different different flavors of the content so yeah even if there are no customers or there are no subscribers the it so the idea for this would be to enable a mechanism that could allow the customer to signal that the subscribers are willing to apply this ecomode consumption and this could be done either by default so always assuming that the subscriber will request this echo mode or even allowing the subscriber to explicitly request this echo mode because of the to do so so So the different benefits that could have this echo mode could be maybe reducing

01:40:00 - 01:42:00

Go to 01:40:00

the number of parts of the multiple multicast parts in the the network, enabling sleep mode maybe also the string case will be to switching up some ports if they are only devoted to multicasts and so on on so far so what we are doing here is to propose this extension and maintain the i give mp and meld messages And for doing that, we we are proposing to leverage on rfc 9279 the idea would be be to incorporate this extension basically indicating the commode value by now very simplistic in this first version essentially as a Boolean variable. So one would be echo mode set to to true, zero echo mode set to false and the echo mode will be intended to be supported in both query and report messages so in such a way that the subscriber can indicate that or the network can ask for it and basically having induced two messages the logic behind this echo mode is not yet covered in, I mean, the logic behind in the sense of what can be done with this echo mode is not covered now in the draft but we can assume a set that could be could imply optimizations in the in the in the network and could be seen either by the subscriber or even by the the on behalf of the subscriber, for instance, the TV to enter in a sleep mode or basically this kind of things or the subscriber just simply will be willing to reduce the resolution of the video because he can live or she can live with less resolution content So next steps would be, okay, this clear that this is a very basic draft by now so the day would be to collect feedback and check if there is interest in the working group in going in this direction We think that it is interesting it's somehow the IGMP and MLD messages are somehow a vehicle for the customers to express willingness somehow we have data, we can have the, let's

01:42:00 - 01:44:00

Go to 01:42:00

collect the data interest from the from customer on this we would like also to analyze the possibility of more complex coding and logic for the common value so by now we have the true or false. So I mean Boolean variable, but probably we can, the idea could be to select among different modes maybe different resolutions for instance could be an option so we will analyze that we would like like also to address the point that could be raised along today and prepare a new version from Addit as well Thank you, questions Oh, mine is up Mancomna from Cisco So I don't think that even if we want to do some eco known multicasts so multicast protocol may not be the right place it has to be done in the routing so if the routing path is being picked using any of these characteristic multicast will just work on top top so iGMP or p m extension for these eco mode may not be really needed needed well in my view this said this could be just simply a vehicle for taking the willingness of the customer for doing that. I mean, by now we don't have any other vehicle any other way of collecting the interest from the customer So that would be this implication of this maybe can the activity later on to the routing and so but we need need to understand the customer or the subscriber is willing to be applied this echo mode for for him or her Yeah, I'm far less confident so I'm more on the asking side i remember 10 15 years ago that we wanted to have credentials to allow you to join something. We never got that through it, so it's nice to see I haven't followed all the RFCs that, you know, such as uh that r fc could could be used to do something where we know it needs to to get into IGMP but so yeah, I'm on the

01:44:00 - 01:46:00

Go to 01:44:00

edge. So if basically there is what is it, lower bit rate, a kind of less energy consuming version of the street stream, the application client would tell the server, hey, want to be, you know, nicely conserving and it gets a different group address and or SSM channel, course and joins that, right? So I you have any hard example where that shows it must use the igmp as opposed to just, know, joining a different channel? or group group well i need i need to think on that yeah I'm probably can come back with more clear examples. And as I said, right? so I think I couldn't find a really good replacement for the the access control things kind of where you have a password or so that allows you to join a channel so that that was pretty pretty good we kind of could do it now with these extensions but for the echo i have no good idea why it must be in the joint so that that think that would be okay thank you you you happy to prenaxie scott so i just a quick comment that you are described some kind of a signaling but you don't know what you want to do with it it. Well, we have not entered yet into details i mean because there could be things that can be done in the network also outside the network Well, that's exactly what we would like to understand before we decide what signaling this makes sense or not because I think we are going way ahead ahead of ourselves. OK, got it so I will try to thank you more details Sanizan ZTE, in fact, haven't get the logic between the problem statement and the solutions but it seems like this draft will be updated from before next meeting right? So it's better to add the more description about advantages to use the echo mode mode, what it will bring for us yeah true thank you All your comments are totally fair

01:46:00 - 01:48:00

Go to 01:46:00

so I will try to provide more details and justify better the idea thank you you. Thank you you Thanks, much. Thank you right, We have 15 minutes I'll try to find it here Yeah, exactly Hey, PIM working group It's been 20 years since RC 4601 has been published so I figured I'd come back and say hi So I'm here to talk about about a deterministic ECMP selection in PIM RC 4601 had this really naive statement of you pick the RPF neighbor or the RPF interface and it was naive because it just assumed there was unanswered to that that. But if you look at, you know, a modern leaf spine network there's a lot of ECMP available to you And so, in this, in this, example we have the source over here on the right and two receivers on the two left routers And the problem comes when uh leaf zero picks find zero and leaf one picks spine one and sends their joins in those directions and then the spines all send their joint to leave two this isn't really a problem traffic is flowing, but traffic is flowing through both spines and it doesn't really have to to because each of these leaf routers make an independent decision about

01:48:00 - 01:50:00

Go to 01:48:00

which RPF neighbor they're going to send their join to so this is this is just a little bit in inefficient um as the scale gets bigger the inefficiency gets bigger also. This is a tiny little example, so it fits on the slide slide so the solution here is for the two leaf routers to do a hash computation on the source group and router ID and since it includes the router ID from each neighbor, they're going to come to the same decision about which neighbor they're going to use and so they both both come to the decision, we're both going to to spine one um and so now we have less traffic we have no traffic traffic flowing through spine zero Leaf 2 doesn't have to do any replication. It's just sending it towards one spine And so we have more efficient traffic flow flow wait a minute say, RFC 67 6754 talks about PIM, ECMP. Why don't we use that well there's a couple of reasons reasons, but the main reason is because it's designed for for an ECMP path between two routers Router A and router B in the left diagram, they have four links between them. so router B can say, oh, router A, you know, I'm forwarding you know, 90% of traffic over this link and this other link only has is 10% utilized. why don't you switch your join over to this other link? that's how it works as B kind of informs A, which, what kind of join to send it. If you go and look at this diagram on the right which is a subset of the diagrams on the previous pages, there's no real way for this leaf two to tell spine zero

01:50:00 - 01:52:00

Go to 01:50:00

to switch to the spine one link like that's that's not how it works. You can't tell one router to start send having traffic flow through another router router so um so that's uh that first level of this document is it describes this hash algorithm SG router ID and and solves the problem in this kind of network topology. But this kind of network topology, Toilis, want to ask now? or save it for a little bit? bit? This kind of network topology is people are growing past this have long grown past this so you have leaf size fine super spine, have multiple replications of this and they are interconnected by a another layer so this multi-tier fabric, you end up having the of these pods at the at the bottom level well they can use the algorithm that I've already described here, but you could have a scenario where in Pod 1, well okay the joins hash to the yellow router And in pod two, the joins hash to the green router. You know, they'll be deterministic within the pod but as soon as you go outside of the pod, you have the same problem at the next layer up up. And so in this scenario, you could have that, you know, it gets harder to draw so I didn't draw it, but you could have the same exact scenario as i had on my second slide where you have traffic flowing through in this case two different spines where two different of the top spines where you could with quarter with

01:52:00 - 01:54:00

Go to 01:52:00

right coordination just have it flowing through one And so the, oh um so the answer here is to have these spines within a pod advertise their color downwards. And so the red spine says, I'm a red spine, the yellow spine says I'm a yellow spine, and then you do the same kind of hashing except with SG color color. And so in Pod 1, you hash to SG yellow. And in pod two, you hash to SG yellow So every pod is going to hash to the yellow spine for the same same SG. And so you get the same mechanism again at the next level level. That's most of my technical content. Troilus, you want to ask? your question? Why don't you go on to? The, the hash is just just each router takes for each of the ECMP next hops possible it takes either the router ID, in the first algorithm or the color in the second algorithm. Yeah advertised by that that. Next stop, it hashes them all and choose the highest hash or lowest hash or whatever I'm trying to remember all this stuff. is no more than 10 15 years back that I last looked at all these things. So i think if you just take you a hash in the same way as the IPV6 or so PIMRFC does it just you know adding the color color in there. You're still not having a stable hash right so

01:54:00 - 01:56:00

Go to 01:54:00

what happens is, um, if a link fails then you will rehash you'll I mean, it's the modular the N links right the no no you that's what was asking what this is okay so this is stable because because instead of hashing and then using that hash to pick a neighbor, we hash every neighbor and we pick the highest value. Right. So that's, yes is stable unless the current select neighbor is the one that goes away Right. right so so and i'm trying to remember if we just implemented that or if there's also an rfc for that that because I don't remember but yeah so that that has been implemented the the color stuff i think um wasn't implemented but instead you I think the implementations that were using the IP addresses of the next hop so that obviously wouldn't give you exactly the same results over multiple zero hops right? the color. The color is introduced in order to handle this scenario where you you the decision has to be made all the way at the leaf but you don't have the information you're at least one layer away in the in the network from the thing that you need to have on. And so that kind of, the yellowness gets communicated downwards by a hullab option Yeah, I think that's, that's, I mean, the, I do agree that the the coloring sounds correct i need to figure out again what impact this may have if you use it in different topologies kind of the polarization issue which we also looked into which is in other topologies, right? like the diamond topology, that's, that's when you're getting polarization based on what hash you're doing so i'll need to figure out what that is. But obviously, you know, if it's just one conf

01:56:00 - 01:58:00

Go to 01:56:00

you know policy then you may not want to use another policy in other topologies if it doesn't work um and yeah whether to use signaling or if i mean these people have shitloads configuration right? mean, configuring, you know, I'm just trying to, to, figure out if additional signaling stuff is really that worth it right but yeah the colors sounds nice nice just to give a little bit more about why I published the document, this is is something that ERISA implemented in his shipping using using proprietary hello options and our customer asked you know can we publish it so that we can we can use other people's routers in these locations in the network so with that does mean is that the algorithm is different from the PIM, like the hash is different from the PIM hash. I can go into more details, it's better to hash that out on the list David? That's know. Actually clarifying question I don't believe I understand why the color is necessary to handle the deep case because if the last level like the bottom one in your diagram is consistent in this in deciding which middle level it chooses yeah then that one being consistent in what what top level chooses doesn't that that solve the problem already the problem exists in these dots i i said i didn't draw all of the nodes. That was a mistake should have drawn all the nodes and then I could point to the solution, reason that it's a problem. So it's not full mesh? no it what i'm the dots mean like all the way on the left it says pod one all the way on the right it says pod 120. That implies there's a hundred eighteen more instances of this in the middle middle but okay scale is big and that's why it's hard to

01:58:00 - 02:00:00

Go to 01:58:00

draw the actual problem but the actual problem comes when you have have at least three of these pods not two This was my mistake in drawing the, the, the, the but even with three or four however many as long as the inside the box, the bottom layer will be consistent in choosing the first color layer right but even without the color it's based on in my in my previous solution it's based on router ID right and so um in Pod 1, it might choose yellow and in pod two it might choose choose green because you're not because the router idea is kind of arbitrary not associated with the color at all thank you. think I go ahead you. Yeah. It's tricky for sure sure. Sandy Zong Ziti, I think this problem is actually true in the large data center So I think this solution is very important And I have some questions. The first is that that in R RC 660395 the hollow option includes the router ID and the interface ID so seems like we only use the router ID right? Right. Right And it's interesting to use a Hush algorithm. The Hush algorithm introduced in the draft is very simple and we would like to know the if the the implementation or deployment will choose that it's efficiency or practicable. So practicable ability. Yeah, we would like to know if it's really used very good in the actual deployment

02:00:00 - 02:02:00

Go to 02:00:00

So this is a second question. And then a third question about the color. We think that the color indicates the plane right? We always use a plane in a large data center. So I think it's really good But I'd like to know what value will the color be because we must use it for harsh, right? We must make sure the harsh is good for every selection So we'd like to know how to set the color value yeah so regarding the hash, we have a a lot of experience with this hash. You're talking about the hash that's referred to in the document which is different very very different from the one one in the PIM spec We have a lot of experience with this hash we use it all everywhere in our code base and we use it in places unrelated to this that are very very much need of well -distributed hash. And so we have a lot of confidence that this hash algorithm has a good distribution distribution as to picking the color values, in the current spec there are 32-bit integers And because the hash algorithm is quite good it's very arbitrary You can pick any non-overlapping values The hash algorithm will spread them appropriately so So you don't have to worry about how you pick the the integers representing the colors of the of the of spines okay by the way, please ignore my previous email send in the mailing list Yeah, as far as the hash is concerned, the implementation I know with the stable thingy, right, we're in

02:02:00 - 02:04:00

Go to 02:02:00

of doing the modulo, you're picking the highest one. That's the implementation I know it's been using the PIM hash and that's exactly where I told you already on the floor that a couple of years later some customer came and said we did large simulations and we're looking at our large service provider core and it's, not really very well splitting on that so then I was talking with with the scott and so so the recommendation I got for for the best hash was murmur which has been known to be very good but yeah if if we're going to pick a hash algorithm, for this, I guess I would see if we can ask people like him or so to give opinions. So one of the details about why it uses why the spec says use this hash is because half of the reason I published this document is to be interoperable and so Arista's existing implementation uses this hash and so it would be easiest to be interoperable if the standard version also use that hash of course Okay, with that, we're going to add. We're out of time.