DUANE WESSELS: Okay, everybody. Welcome back to the afternoon session to continue discussing root server system metrics. We have two sessions this afternoon, and in both of these, our primary goal is to talk thresholds and to hopefully get some agreement on actual threshold values.

It may be the case that as we go through this, we need to clarify some particular aspects of the metrics and that’s fine, but really, the goal is to start talking thresholds.

Before we jump directly into the thresholds though, I want to spend a little bit of time talking about something Paul raised, which is sort of the way the document currently lacks any text about rationale for how we come to select the threshold.

So as Paul mentioned, he wrote in a message to the caucus list a while ago with his rationale of how he came up with his proposed thresholds, and I've asked Paul to append that at the end of the document, in the Google doc, it’s sitting at the end. I'll let Paul say a few words about that if you would like to now, but as we go through these metrics throughout the day, please keep in mind your rationales and be very explicit about them so that we can get them captured in the transcript. And then for those that we do come to an agreement on, we can document the rationale that the group came to when we publish the document.

Paul, would you like to say something?

PAUL VIXIE: Actually, Ozan, do you want to show the text or not? So the last page of this doc, literally all the way at the end. Keep going. There we go. So I just added this at lunch, which is why it's green.

What I had sent to the caucus mailing list a few weeks ago as we were starting to discuss thresholds was I think that we should put rationale into the document with the thresholds because the people using these thresholds a couple years from now are going to see numbers and they're going to assume that they know why we thought that, and they're going to be wrong. And those people might even be us and we might have forgotten. Most of us in the IETF have done this over time.

Not only who wrote that, but oh, I wrote that, and what I wanted at the time was X, and somebody's like, "No, you didn't want X." So the summary is the major part there which is that the importance of the metrics in my mind is that correctness is the most important metric and soon after correctness is publication latency, and that those are much more important than availability and response latency.

Again, this is just for RSO, not RSS. And then the following paragraphs go into why I have that, but basically, the idea that I had is not that these thresholds are supposed to be about today, but they're supposed to be about what we expect of each RSO. And we expect goodness, as Duane kept saying this morning, but at a minimum, we expect each RSO to do something that will be of benefit to the resolvers that are making root queries.

The way that I categorized when I came up with my thresholds from this rationale was basically we know that each of these are important, but to get a number out of correctness would be too hard for anything less than 100%, like let's say the one root server operator was consistently being incorrect on one answer for one record. Well, oh, but it's not an important record, oh, it is an important record. That would drive us crazy. And therefore, I came up with a correctness of 100.

Publication latency, again, is slightly less important, so it doesn't have to be 100%, which is good because actually, the initial measurements I did, only a few RSOs would even get there. But looking at the current root zone, publication latency, you want to have all the root server operators correct within, say, 95%. You don't want someone to be more than 5% late on something, so that's about one hour out of 24, was how I came up with that – like we talked about 95% threshold. That seemed like a reasonable one.

Going to availability, the reason why availability's so much less important is we know that every rational resolver – and that's what we're assuming, we say that we're not going for every possible resolver but for resolvers that follow RFC 1035 and such, if they're going through their priming or repriming and the person they're talking to just doesn't answer, they know that they should go somewhere else. That's a complete known thing.

So it's not super important that everyone is up all the time, because if it's you and you're not up, they will move on. So I was much more lenient on that than I was on correctness or publication latency, and response latency is even less important because we know that all the resolvers are using software that also looks at response latency and has its own secret sauce for determining that. There's no way we can say this is the right way to measure response latency because Bind does it different than Knot Resolver. And you can also tweak that.

The way to come up, with me, for a number on response latency, which I had judged the least important of these, was how long would it take for a message – and again, we're doing this based on the way we are having the vantage point How long would it take for a vantage point to go halfway around the world to the upstream ISP for an instance. Let's say that it's a root server operator that has exactly one instance. And the vantage point is literally on the opposite side of the world. How long would it take to get to the ISP that's there, and what if they actually put that instance on a fairly slow link on the other side of the world? So you go around, latitude and over longitude.

That's about half a second in each direction, so I made it a second. That was the way I came up with a number for this, knowing that a stronger number would be good but not necessary because the resolvers are going to fix that. If that's what you're doing at your letter, boy are you going to get pretty much no traffic other than priming queries, and that's okay. If everyone did that, that would be a problem. That's not what we're discussing.

So that was how I came up with the rationale and the actual thresholds. Other people might have different rationales, as Duane said, which is just fine. Or you might have different thresholds even if you like this rationale.

DUANE WESSELS: Alright. Thanks a lot, Paul. So I think what we're going to do next is, if you don't know, we asked the caucus – which you're all part of the caucus, of course – to fill in a little spreadsheet with suggested threshold values, so I'll ask Ozan to put that up on the screen and sort of start going through this line by line.

I think the first one we want to talk about is we'll just go in order of the document. Root server availability, you can see that there's a number of responses here. The first one under column B, which says John Q Caucus, this must be an example of how to fill out the spreadsheet with silly examples, silly numbers, so ignore those numbers. But we asked people to put in their suggested threshold values, and ideally, they would also put in some comments into the cell explaining their decision. So that's what the little yellow orange triangles in the corners of the cells means.

If you look at the row for root server availability, you can see that the responses generally vary from 90% to I guess 99.995% is the range I see here. I know Paul gave his rationale for his, some of the other people did as well. Anyone who hasn't had a chance to fill in the spreadsheet, if you have an opinion, now would be a very good time to express your opinion, and ideally, we'd like to get some kind of agreement on what this threshold should be. I'll open it up.

UNIDENTIFIED MALE: For single root server, I think this is based on some math we tried to do an ICANN or two ago. Just if the combined RSS has this number of nines working back, you need at least this many, and it turned out to be a surprise – I think it may have been the numbers you're doing as well about [K of the] 13 servers that are available, bla bla.

UNIDENTIFIED MALE: Do you remember what that yielded for a whole root server system based on – I mean, this is just math. We could probably figure it out.

DUANE WESSELS: So we're going to have a whole discussion about that tomorrow, actually. I've got a slide deck, but I think it's the same formula. But let's save that for later, if you don't mind.

So the goal is to get through this afternoon to get through the root server availability, the response latency and the publication latency thresholds by the end of the day.

DANIEL MIGAULT: So we're not discussing availability now.

DUANE WESSELS: We're discussing RSO availability but not RSS availability.

DANIEL MIGAULT: Well, the question I had, because I [put the] number, the question I had really in mind was, what is an easy number for an operator? Something, yeah, no problem. Because if it's easy to have 99.99999 ... I don't know what is hard and what is costly. I think 90% is very easy, I think it might be sufficient. But that's a really operating question that I don't know. So the number I tried to reflect is something that is easily achievable, because I did not want to give the impression that we're asking every operator to be online all the time at any cost. So that's ...

DUANE WESSELS: Thanks, Daniel. One thing you reminded me that I probably should point out, and I actually have this on the slide that we'll see tomorrow- sorry, I don't have it right now – the way that these measurements and metrics are defined, there is a network in-between the vantage point and the server, and that network might not deliver all the packets.

So even though we might like to say that it should be 100%, I think the realities of the way networks work, this has to be a little bit lower than that to account for packet loss and things like that.

JEFF OSBORN: Hey Duane? [inaudible]. My personal opinion is the number should be high per root server operator. What's high? I guess we can talk about that. But my point is that you should not dumb it down or bring it to a lower level because you're baking in maths saying that by having it at a lower number but there's 13 root servers, the root server system as a whole is a big number.

I still think that each individual RSO should strive to maintain the best availability possible. So that's my rationale. Like 99% per month is a little over seven hours of downtime. I would argue that no root server should be down more than seven hours a month. That's my argument.

DUANE WESSELS: Thanks, Jeff. So the spreadsheet here does have good performance thresholds. I think we'll not talk about those at all this week because based on the discussion we had this morning, we've decided that the work party will not make recommendations on good.

JEFF OSBORN: [It probably affects what people put for the minimum.]

UNIDENTIFIED MALE: Yeah. [inaudible].

UNIDENTIFIED MALE: [Yes, it does.]

DUANE WESSELS: Explain how it affects –

JEFF OSBORN: I was told by somebody offline that they wanted the numbers to look different, and so that affected their view. That's why I was a little bit concerned about us having the good ones there before we had to decide whether to do good or not.

So I think a very natural thing for engineers such as the people sitting in this room to do is to pull numbers out of our ass and then try to justify them. One of the ways we do that is look at who said anything first, and either we want to be better than them or less good than them. Which is why I started with the rationale because I did that myself and I was like, “Oh, this is bad.” Which I guess is the opposite of good.

But we do that. We just do that, and we may end up doing it. I'm not saying we can't do that, because we all do that naturally. I tried not to, but we all do that naturally. And especially when you are faced with the minimum and good, it’s like you would be embarrassed to have the minimum, the good be the same. So if we aren't asking the good, other people, I believe, would change their minimum.

I tried, for my own, to do this based on an understanding that at least five root server operators were up at the moment. That is, at some point in the future, the number 13 might go down to five, but I based mine on – if this was the only root server up, I absolutely would have this be 100%. If it was one of the two root servers up, then it gets a little bit like, what if one of them goes down while you're measuring stuff?

So I just picked the number five and tried to do real math based on that.

FRED BAKER: I can give you something that would be similar to a vendor viewpoint, and that’s coming from – what was it, 22 years? At some hardware vendor. We really strove for a network to be up [five nines.] For something to be up five nines, we tended to take the perspective that any particular piece of equipment should be capable of five nines. If there's something wrong with it, there's something wrong with it.

So then we looked very hard at what five nines meant, and it meant basically five minutes a year it was allowed to be down for some unexplained reason. Somebody might take it out of service. That’s another question.

So this 9995 at Cisco, we kind of said there should be another nine in there, and we looked at our hardware designs and our software designs with that perspective.

I think the things you're going to show tomorrow kind of say it’s fine for a service, it’s not fine for an individual element because the individual elements are more unreliable. And fine, I agree. But I can tell you from a vendor perspective, I tried very hard to make it as close to 100% as we could get.

DUANE WESSELS: But are you comfortable with being held accountable to that level as an operator?

FRED BAKER: AN operator’s responsible for the service, which is, [inaudible] just talking about a piece of equipment. And as an engineer at Cisco, we were held accountable for the equipment.

DUANE WESSELS: Okay, but the point we need to get to today is we need to settle on a number, or the work party needs to recommend a specific minimum threshold for RSO availability.

FRED BAKER: Well, yeah, and so my understanding – and I think I understood this from your comments a few minutes ago – the root server operator is for one piece of equipment.

DUANE WESSELS: It’s for one operator.

FRED BAKER: For one RSO.

HOWARD KASH: I don't know if we've looked at the existing sites out there like DNSPerf.com. Right now, all the operators range from – the top is 99.96 and the bottom is 98.2, and that’s based on IPv4 measurements once a second with a one-second timeout. That may be why it’s a little lower, but just to give you an idea.

So that represents what we sometimes call peacetime. That’s normal operations. But I think we may want to also think about other times when there's more load on the system. What should the availability be in that case? How many queries should get responses?

FRED BAKER: The other question I had for folks that filled the numbers in up there for availability, we’re talking v4, v6, UDP, TCP, were you thinking in terms of the numbers you put in the spreadsheet of the aggregation of all of those, or for each?

DUANE WESSELS: Yeah, that's maybe the fault of the person who designed this spreadsheet, which is probably me, that they're not separated there. But it was not the intention that they be aggregated. I guess ideally there should be – if we feel that TCP has a different threshold than UDP, then there should be separate lines in this spreadsheet.

UNIDENTIFIED MALE: For those of you who are also still looking at the document, when I sent in some samples that I had done – and this was in an earlier version of my just proof of concept one, going to your question of UDP versus TCP, the one date where I have measurements, there are almost no timeouts for any of the RSOs for TCP and there are timeouts for almost all of the RSOs for UDP.

So we may end up having to make those distinctions, and also, v4 and v6 turned out to be sort of similar, but it's hard to tell because of the TCP UDP difference.

DUANE WESSELS: So, what if – since we have a range here from 90 to 100%, what if we said 95%?

UNIDENTIFIED MALE: 95 you said?

DUANE WESSELS: Yes. There's two things hard here. A, we have to come up with a number, and B, we have to rationalize it. We have to come up with our rationale for how we got there. So we've got this spread here. How are we going to get past this?

UNIDENTIFIED MALE: I personally think 95 is just too low. That’s 36 hours of downtime a month. I just think we all hold ourselves to a higher standard than that. I would start as 99 as the base level.

LARS JOHAN LIMAN: I agree that 95 is too low, but I don't do it for rational reasons. So following on to Paul’s idea, do you have a rationale for that? I'm honestly asking and I'm curious, I'm not trying to be nefarious. So I think we should try to follow Paul’s idea, what are the technical limitations that drive us towards a certain number. I fully agree that 95 is too low.

UNIDENTIFIED MALE: If 95 is too low, then maybe the question should be, what's the allowable downtime per month? If we could answer that, then you come up with what the percentage is.

LARS JOHAN LIMAN: And why, and based on what.

UNIDENTIFIED MALE: Okay.

DUANE WESSELS: I think that's a good approach, but I wouldn't think about it in terms of months, because the reporting here, again, as proposed, is per day. So you could hit your monthly limit in one day and be way out of compliance or expectations.

BRAD VERD: You could work into it. you could start with like, okay, what is it on a monthly basis, and then break it down.

DUANE WESSELS: Yeah.

UNIDENTIFIED MALE: I would like to suggest two different questions than the one we're stuck on. One is if we were not operating the root name server, if we were just depending on it like the rest of the world is, what would we accept as the standard that they thought they could uphold? And what is the number below which we would giggle and say no, that's not okay?

And the second thing I'd like to point out is that this is the system's availability level, correct? Not individual root op.

DUANE WESSELS: This is individual.

UNIDENTIFIED MALE: Okay. So if we are having, let’s say, downtime somewhere because there's maintenance going on on the route for a given Anycast node, that’s not an outage. That would not count against your SLA. It’s only if nobody can reach you and the problem isn't on their end that you would be considered down, because someone else – so a ship drags anchor, breaks a transatlantic fiber and so you can't be reached until the routing system stabilizes, that also should not really count against you because the packets aren't reaching you.

So there's no way for somebody to enforce an SLA because of that subjectivity, because your position is relative in the topology at all times, you're always lagging. So we've gotten this far on best efforts, and we've gotten this far by not all being down at the same time. And I don't know how we can write this as an RSO-specific SLA default that doesn’t take account of the fact that if you're down but the rest of the system is up, it’s not a problem for anybody, so nobody cares. Thank you.

FRED BAKER: If I could just go first here, Jeff. One of the things I wanted to point out relative to what Matt said earlier is that as we currently have things defined in the document, the queries are generated from vantage points. 20 is what we’re working with right now. And they are generated going to randomly selected RSO addresses with the results then put together to give us these numbers. And it is something I think we need to consider, whether we're looking at 99 or 95 or something else in-between, that what we're going to see occur on a daily basis is things will change, they will go up and down, and we don't have, I believe, enough data yet to set a very high minimum objective for ourselves here.

So I think we need to be careful in what we pick, whether it's 95 or 98 or something else. But that's one thing that I think people are finding it a little hard to adjust, is from thinking about terms and how many hours of downtime does this machine have in a given month versus what is the likelihood of these numbers that are going to work out from packets going from vantage points to an RSO and back and getting measured from that. We don't know the real answer yet.

[JEFF OSBORN:] Well, again, for one, I'd argue that the number of hours you're down per month is very different from applying the same thing per day, because seven hours in a month is a lot easier than beating 14 minutes every single day. You could have a 15-minute outage and blow your day where you never hit an eight-hour in a month.

I guess what I'm trying to say is I know F-root as a system hasn't gone down as a system in a hell of a long time, so if we're measuring our organization and system, then 15 minutes a day or seven hours a month, I can't imagine why we'd want to say 95%. 99% almost feels like cheating.

UNIDENTIFIED MALE: But if you look at like DNSPerf, you're at 98.2, so there's a lot of things in the middle that you can't control that would affect those numbers.

LARS JOHAN LIMAN: I think that looking into numbers that measure different things that we're talking about here is going to be a red herring, reversing – where was I? Yes, the thing I want to be allowed to happen in my life is that someone who operates the I-root cloud makes an error, gives the wrong flag to the route down command and blows out the entire set, realizes that, and puts it back up again. And that's probably a 15-minute effort. If that happens one month, I would like that to be a thing that can go under the bar without being a disaster. But not twice the same month.

UNIDENTIFIED MALE: So people keep talking about months, which is fine. As written, the document focuses on days. If we need to change it to focus on months, then we can do that, but what we've been working on so far is daily thresholds, daily metrics.

LARS JOHAN LIMAN: So following from my statement, okay, we need to either rephrase that timeframe or make that allowed per day.

UNIDENTIFIED MALE: Or have different thresholds, one daily threshold, one monthly threshold, if you like.

UNIDENTIFIED MALE: This is [inaudible]. I'm having trouble coming up with an exact number for this, but the approach that I want to see if it makes sense to consider is if I were operating a resolver, if I want to somehow quantify what does this mean for someone who’s depending on the system, so I feel like we’re not talking about the RSS metrics, but at the same time, say a resolver tries thrice to get an answer from the root, so it tries one, it fails, it goes to the second, it tires, and then it tries the third and then it gives up [inaudible].

What's the minimum sort of probability that that will happen in a joint sense? And what's the level of availability each server would need to have to prevent that probability from being like very miniscule, maybe it’s 2%? So I'm having trouble operationalizing that to get an answer, but that might be a way to kind of rationalize how are you building that availability number for RSO.

PAUL VIXIE: I can answer your question because you just said if it was 90% and the resolver would die after three tries, that gets you 99.9% right there, and that’s with the 90% threshold. We were talking about a 95% threshold, you're now at 99.995 or so.

So again, if we want to do the rationale mathematically that way, we can do that. But again, that’s from a rationale that’s quite different than what Matt had brought up, and sort of different from Liman because Liman's wasn’t even up or not, it was operationally, what if I do an operationally believable thing, like set the routes wrong and then see it?

LARS JOHAN LIMAN: I think you are right in taking that approach though, because that probably leads to numbers that are quite acceptable. And I really like that approach, because by combining the numbers for the [individual] root server operators, doing the math, we reach a number for the entire system and that's actually where we want to start and then do the math backwards as you suggested. So I actually support that view.

DUANE WESSELS: If we want to go this way, we can do that, and I would switch to presenting the slide deck that I was going to present tomorrow which has this math in it. I guess maybe we should do that and get it out of our system.

FRED BAKER: To see how it floats with the work party.

DUANE WESSELS: Yeah, okay.

BRAD VERD: Before I forget this comment, I like the idea of having monthly thresholds, and I only like it because I'm used to it in the real world. So maybe in the IETF or the technical world where people are used to daily or minutely or every five minutes or something, but real world – if I'm an Internet user, I can understand a 99% availability for the month, because kind of everything in my life is based around months. So it's just something to think about.

DUANE WESSELS: Okay. Thank you. That's fine. I've wondered the same thing, it's just so far we've only talked about days. But we can change to months.

UNIDENTIFIED MALE: [inaudible].

DUANE WESSELS: Or that. But it's going to affect probably people's answers to some of these questions. Well, Ozan, if I can ask you to put up that other slide deck, availability slide deck, then we'll go through this I guess since we need to. Can you put it on display mode, or does that not work? Or slideshow mode? If it doesn't work, we can live with this.

Okay. So this slide deck was, and is, mostly about RSS availability, but there's a little bit at the start about RSO availability, and maybe, as has been suggested, this will help us figure out the RSO availability from the RSS. So go ahead and go two slides down, to number three.

This is reiterating something that was presented earlier. Again, the RSO availability calculation is a straight fraction. It's number of responses divided by number of queries. And this is easily measurable, this is something that we can measure by sending queries and responses and so on. Go to the next one.

The spoiler is that measuring RSS availability s harder. This also is something that I mentioned a while ago, that the way RSO availability metric is defined, there are these elements between the vantage point and the server which are also essentially being measured. You're measuring the networks in-between and their availability to deliver a package to you.

I think it’s important to keep that in mind when coming up with thresholds. This is unfortunately the reality, I think, of the situation. At one point, we were talking about a scheme where the vantage points were essentially right next to the RSO to take these ISPs out of the equation, but we decided not to do that. Instead, we thought it was important to have some sense of externally verifiable metrics, so the vantage points are not adjacent to the RSO. There's a [inaudible] between that figures into these measurements.

Okay, next. RSS availability. So as we've talked a little bit about, if you do some research and whatnot, there are different models of availability for systems with multiple components, i.e. similar to the RSS. One of them is what they would call serial availability, that is you have some number of components and the path of execution or whatever flows through all of them so you need all of those to be up in order for the system as a whole to be up. That’s really not applicable here so it’s not really considered. There's what's called parallel availability, which is a little bit applicable. We’ll talk more about that in some upcoming slides.

There's [K out of N] which is sort of a derivative of the parallel availability. That’s probably the one that’s most relevant to what we want to talk about. And there's also called load sharing [K out of N,] which gets really hairy in terms of math and really complicated. But to be honest, it’s probably the one that’s most similar to our system. That means for example that if one of the components goes down, the others have to absorb its load, so that really does happen. But unfortunately, I think the math makes it not really a good model for us to consider in our discussions here. Okay, next.

So, simple parallel [availability] is when you have a system with components and they're all assumed to be fully independent, and the availability of any one of them is sufficient for the whole system to be up. And there's a formula there which says how you can predict or estimate the system availability from the individual component availability and there's a little graph that I stole from somewhere which shows how that changes depending on how many components are in your system. So this is pretty straightforward, and at the bottom, there's an example where if you have a system with 13 components and each individual component has 50% availability, your overall system availability still has almost four nines. So that’s what you get from n=13, which is pretty dramatic, right?

Okay. Any questions about this? So this is just FYI, we’re not really, I think, considering this further. Next. So K out of N, parallel availability is when you have N components and some smaller number, K, of them are required for the operation of the system. Again, in this formula model, all of the components are assumed to be identical. That is, they could all handle the same amount of load, they're all assumed to be independent. If one goes down, it doesn’t affect the others.

Classic example of this would be engines on an airplane. You might have an airplane that requires two engines to be able to fly but it’s built with more engines for reliability purposes.

Here's this more complex formula about how you can predict the availability of such a system, and here's another kind of si8lly example maybe, but if you have 13 components and you require 13 of them to be up for operation of the system and the availability of any of the individual components is 90%, then the overall availability of the system is 62%.

So this goes in the other direction, as K gets bigger here, the overall availability goes down. Everyone with me so far? Okay. So if you go to the next slide, then this is a table of some of those numbers that you get out of the formula.

DANIEL MIGAULT: I think there is an independence between the different measurements.

DUANE WESSELS: That’s right. In this model, it’s assumed that all the components are independent.

DANIEL MIGAULT: Yeah, which won't be the case under a DDoS.

DUANE WESSELS: I know.

DANIEL MIGAULT: Okay.

DUANE WESSELS: We have to make some simplifications here in order to get any progress at all, I think. Otherwise, it gets really complicated, and I don’t have the math background to talk about that.

So this is just a simple table of plugging numbers into that formula. Along the top are – really, it’s kind of labeled M there, but it should be the K to be consistent with the other formula. So across the top is values of K, one through 13. The rows going down are A, availability of an individual component ranging from 90 to 100%. And then you can see in the cell there the calculated overall system availability. But one thing that you'll see here that’s important to keep in mind is the way this formula sort of works, the value of K is kind of a design parameter. It’s an input. It’s not necessarily something that you can reverse out of this formula.

So for example, if we looked at one of these columns, say we get K=7, what we’re saying here is that in our system, it requires seven components to be up in order for the system to be functional. If it drops below that, then the system stops working. It’s not that it becomes less reliable, it’s that it stops working. Again, think of engines on an airplane. If you have a plane with four engines and it requires two to fly, if you lose an engine, it’s crashing. It’s going down.

Okay, so this is the table for, again, 90-100%. The next table is similar, except it just sort of zooms in on the 99 to 100% range to get a little more detail there. Okay, next slide, please., Ozan.

So some challenges with this model and this formula is that the formula can be used for predicting availability, but it doesn't really help us in calculating the achieved availability on a given day or given month or whatever from the measured components. As we said, it assumes all components are identical and independent, which we know is not necessarily going to be true. It assumes that the availability doesn't change as load increases, so I think under a DDoS, something like that, it may not really apply. Next slide, please.

So I spent a couple days looking at these tables and this equation, and really convinced myself that you can't use this model, this equation to tell you what value of K you need for other given values of availability and N. So as much as we might like to say, "Well, I want the overall system availability to be 100% and individual components should be 99%," what value of K can I tolerate or how many root servers do I need to be operational? It doesn't really tell you that. It'll tell you the probability of surviving some number of failures for given design parameters, but it doesn't tell you how many you need.

MATT LARSON: Duane?

DUANE WESSELS: Matt?

MATT LARSON: I have a question.

DUANE WESSELS: Yeah.

MATT LARSON: If your requirement is, say, a value of one, then you can look at your table that you just showed, if you back up to the table for example, any value of one is acceptable, right?

DUANE WESSELS: Essentially, yeah. There's rounding errors here, these are some number of nines, but yeah, they're not exactly [inaudible].

MATT LARSON: My point is – can you back into it? Can you look at, if you want one and you look at the values here, you can back into it?

DUANE WESSELS: Ozan, can you go up one more slide? I'll work through the example that I kept trying to convince myself on when I was looking at this. So let’s look at the row sort of in the middle where A is 95%. So if we say our individual component availability is 95%, just picking a number here, and I want the overall system availability to be 100%, so go across, and I end up with something like maybe five nines availability instead of 100%. I want five or six nines.

That would be like seven. So you could say, okay, that sounds reasonable. Seven servers, seven out of 13 at individual 95% gives me 100% availability. Okay, I can believe that. Now, what if my individual components are even better than that? What if instead of 95%, they're 99%? So I go down to 99 and I look for that place where it changes from one to a little bit less than one, and it’s nine. Does that mean I need nine servers out of 13 if my individual components are even more available?

So I improve the availability of the individual components, and this table tells me that now I need nine instead of seven.

UNIDENTIFIED MALE: [inaudible].

FRED BAKER: I think you hit on it when you said it’s a design parameter of the system. The availability of the components does not influence how many you need for the system to work. That’s just what it is.

DUANE WESSELS: Like you, Matt, I really wanted this to work that way, but I couldn’t make it work. It gives you misleading or incorrect conclusions. So if you scroll back down to whatever slide we were on, 13, next one, please, and in addition, this formula doesn’t help you calculate the daily availability. The work party needs a method for calculating overall system availability from individual measurements performed by vantage points. And my conclusion is that the only way we can do that is if we somehow otherwise agree on a value of K. if we can come up with the value of what the experts say the value of K should be, then all these other things can fall out of that.

Then we can say, “Okay, I want 100% system availability. That will tell me what my individual component availability needs to be.” But the hard thing is we have to come up with K first. We have to decide, is it one, is it 12, is it nine? Whatever. And I don't know how we’re going to rationalize that, again, other than maybe gut feel.

BRAD VERD: I agree with that. I think it’s going to be hard to provide a math equation for the community to read and say, “Yeah, okay, I get how you get this.” I think based upon – I don't remember the document we did, but I don’t think 13 is the number now. I think the number is 12 because we've stated that one can go down without it impacting the service, so our new baseline is 12. If you were to ask me what my gut feel is, it’s probably eight, and I've said that all along. I think it’s less than 13. I don’t think we need 13. I think it’s eight. But that would give you more that could – I think you need eight available in order to provide 100% service. That’s my ...

UNIDENTIFIED MALE: Is that eight no matter the number of N, or is that like half of N? So if in the future there were 20 RSOs, would it still be eight or would it be ten or ...

BRAD VERD: It would be two thirds.

UNIDENTIFIED MALE: So I'm just trying to get a sense of, is your gut feel a fixed number or is it a relative number depending on the number of operators?

BRAD VERD: I think it’s a fixed number. I think we need eight based upon the design points that I understand to be in the root today. But I'm not a math expert.

UNIDENTIFIED MALE: Methodology for determining K, it’s got – hypothetically here, but load divided by capacity. Now, we can't load, you’ve got to put in your DDoS and safety factor’s capacity, we’re going to have a number for that. But that ratio or some estimate of that might be a good way to – we think that given this load or expected load, you can do that entire load with K servers.

PAUL VIXIE: So it can't be eight. It’s got to be some fraction of the current population [inaudible] operators of the servers. Although since we've said that one can be down and we've got 12 servers that therefore have to be up, that’s the same as the number of operators. But just technically speaking, it is the number of servers, the population of servers, not the population of operators that matters. So eight makes sense to me because it is two thirds.

Now, there isn't an RFC out there that somehow requires or has the ability to interoperably prove that somebody complies with a protocol for trying the next server to get a timeout. So we all know that there are horror stories out there, there are certain recursive nameservers that will try them in order, starting with A, then B, then C, etc., which means that if the eight that are down are A through H, then that particular server is going to try nine servers before it finds one that works. That’s a long time out. There's no application in the world that will still be waiting for that answer. So we have to ignore the fact that there isn't a standard for choosing the next one if you get a retry or if you get a timeout, and just sort of pretend that it’s somewhat random. So if you don’t get a return from the one that you first try because it is one of the ones that are currently down, I would like there to be two to one odds that the next one – no, actually, one out of three odds that the next one that you try will be up. So actually, that would argue for seven rather than eight. But we just have to deal with it statistically. This is the wave function of resolvers, and so it needs to be some fraction of the population in order that we can then predict how many times they're going to have to retry before they get service. Thank you.

UNIDENTIFIED MALE: So Paul, just one wrinkle on that, which makes it a lot more liberal, is that for this document, we agreed that all of the resolvers are actually caching. So this would be a resolver that has a query that's not in its cache, that it wants to get an answer for, and whatever the last one it was talking to, which was probably only a moment ago given the ubiquitousness of Google Chrome, which is forcing us to keep going back all the time, that server getting a query that is not in its cache and having to go through them all. So it really is a question of at a given moment, have all of them gone down or not? At the same time, not during the month and not even during the day.

PAUL VIXIE: I understood that. Let me explain. If [at the moment] that they're trying to make a root query because they've gotten a cache miss and I guess we're hoping – although we don't have the document – that they're using QNAME minimization and they are using the DNSSEC spans in order to cache the negativity of certain parts of the namespace. But if they really are going to the root servers because they really do have a valid reason to need an answer, then they've got some system, maybe they're going to use the same one until it fails, maybe they're going to try them all and home in on the one that's fastest. So who knows exactly – I don't think we can model what it will choose next. And what that means is that it's a matter of pure – what do they call it when you stick one bullet in a gun and spin it? Russian roulette. So they are playing Russian roulette, and so what we need to know is how many chambers have bullets in them, because that's what indicates the probability that you're going to have success.

So we could say that half of them are down, half of them being down at the same time is not an operational problem because there's a 50% chance that the next one they try is going to work, or we could say that it has to be two thirds that are up so that it’s two out of three. That probability goes right across from the fraction. So if we say that eight have to be up out of 12, that’s two thirds that have to be up. I'm okay with that. But I think it’s a pretty arbitrary number.

LARS JOHAN LIMAN: I like that way of reasoning. I think it’s important to put that in the document that there is no way that we can know how different resolvers work. That part was very important, and what you said, Paul, I think, follows out from that in a good way, and that we, from that, have to more or less make up a number that we’re comfortable with, which is either pulling [inaudible] as you said, or actually do some kind of measurement or experiment to see what happens when a certain fraction is down or bigger fraction is down, or not.

And again, to get the K as a design parameter, because that’s – your math is obvious here. Thanks.

DUANE WESSELS: I'll take it from a different angle. Why can't K be one? This is really a question about recursive resolvers, and maybe I'm ratholing here, but – and Paul, I'm looking at you for help on this. Do recursive resolvers give up after a certain amount of nameservers in [an NS set] or fail?

PAUL VIXIE: There are actually two layers to that question. Recursive nameservers tend to be persistent. Once you give them some cache miss, they will utilize the last electron in the universe if necessary to go try that against every possible upstream, so every address hanging off of every name, whether IPv6, IPv4, whatever. It'll try them all.

However, long before that heat death can happen, the application will have retried once, sent a query, didn't get an answer because it doesn't get told, "Why don't you wait while I find it for you?" It just gets dead air, and at a certain point, they begin to interpret more dead air as "I'm never going to get an answer to this and I have to move on. I have to fail the transaction I'm in the middle of, put an error on the screen and move on." So that's going to happen before the recursive resolver finishes whatever work it attempts, and that's actually fairly portable. I believe all recursive resolvers have that property.

Now, if there were a standard and if there was a way to test conformity to that standard, it would probably be what BIND 4 did, which is to try them all, by which you eventually find which one's closest, and then you kind of decay it as you use it so that it eventually appears artificially further away than it is so that you try the others, and you just kind of home in. That's probably the standard that would be written, but there's no way to test a resolver to see if it does it, so there can never be a checklist item when you're ordering something from a vendor. So there just will never be a standard there.

So what we have to worry about if K was 1 is what happens to a resolver for whom that is the last address they will try, and what happens to the applications behind that resolver. And I think that we should look at that and be terrified by it and say that we don't want to be in that position. Thank you.

DUANE WESSELS: [inaudible] have two comments on K=1.

UNIDENTIFIED MALE: [inaudible].

DUANE WESSELS: Okay. Yeah. So K=1 is the same as the simple parallel availability, right? So if you said K=1, that means your individual components can be about 50% available, and your overall system for N=13 is still very high, still almost four nines. So while that’s what the math says, I'm not sure that’s what we would want to put as our thresholds or our availability thresholds.

And of course, the other one is that if K=1, then that implies that any one operator has the capacity to serve all the traffic.

UNIDENTIFIED MALE: So if I can follow up on it, I was kind of building on this. So we agree that K=1 is not good for some of the reasons that Paul mentioned, some of the reasons that you mentioned. Layering on what Paul explained here, a recursive will try and eventually give an answer back whether it’s gone through the whole NS set or not. What is that threshold by which we think most recursive will not have given up already?

And is two thirds the answers? I don't know what the typical resolver does and how long it waits or how many nameservers in an NS set it tries before it responds back with “It didn't work.”

PAUL VIXIE: I think you may have mixed your terminology there. So these processes are all asynchronous from each other. if you have a stub resolver, they're asking some question, they're getting dead air until an answer is available to them. They will give up and no longer care if an answer comes at some point.

So asynchronous from that, the recursive on their behalf is going to go search and chase CNAMEs and deal with timeouts and retries and t registry the next server and do pretty much an awful lot of work, possibly to eventually return an answer or a serve fail to a stub that has long since moved on and is never going to hear that answer.

So there's a loop within the loop here. That stub probably has several recursives it can try. It will after a few seconds say, “Well, I didn't get an answer so I'm going to try the next recursive in my list.” So it’s asking a whole bunch of different recursives to go do a bunch of work that can take an arbitrary amount of time to complete and it may still have given up on all resolvers and failed the transaction, put an error up on the screen and is no longer going to benefit from all of this work. So when I say two thirds have to be up, what I'm really hoping is that that gives them a very good chance that by the time the recursive has given up on the first server it was talking to and moved on to the next one, it will move on to one that is up and that within that, the one try and one retry, there will be an answer sent back to the stub. I think we have a reasonable shot of that happening if two thirds of the potential servers are operating at that instant, because we can't predict which one it's going to use. We should just have some standard by which the population can never fall below a certain fraction.

DUANE WESSELS: Alright. Thanks, everyone. So I think the only concrete-ish proposal I've heard is for the work party to settle on a value of K equals two thirds of N minus one, something like that? Is that something that everyone is more or less comfortable with?

UNIDENTIFIED MALE: One of the things that struck me as we went through this discussion is if we're going to, rather than use our bases as individual RSOs' availability combined in some mathematical way to reach the system availability, [that we're] going to start at the system and look at what the availability is there, and then, are we going to go back to the individual RSOs and come up with a minimum number there? I think that's what we've decided here.

DUANE WESSELS: It's doable. We can do that, yeah. Yeah, put up that first table, Ozan. For example, if we could agree on K=8, and if you wanted, say, five nines, that looks like it's about 96%. Did I do that right? So we're in that range. So that would imply that if you say K=8, and five nines availability for the RSS, then individual component availability needs to be 96%. That's where we would end up.

BRAD VERD: If I may, first, just going off what Paul – kind of adding to what Paul said, I think 100% is a marketing number, right? It's impossible to get there. But this assumes that everything is equal, and so I will throw the other grenade in.

UNIDENTIFIED MALE: [inaudible].

BRAD VERD: Right. So this also assumes that – I forgot which document we have. It might be RSSAC 001, or I know it was in 2870 back in the day, was that each individual root should be able to handle – was it three times the load of the whole system type of thing? It's hard for me to look at the load of the whole root server system and see that as like the design point. Or like Ken, you were saying that we could come up with K by using load divided by something, and the load on these systems is really small. But the availability has to be, I think, high, because if the root server system isn't there, then you've got a problem. So this goes back to the question that we pushed back on the ICANN board when the board asked us, can you survive a 1 terabit attack? And we went through all this discussion, and then we put into 37 that we can define how much, what the current load is, we can define what normal looks like, and then you, the board, has to define what risk level you're willing to accept. And the risk level is basically the DDoS, what that load on the system during – what do you want available under duress, under a bad day?

And Paul, going back to what you said earlier, we need to answer all of the valid questions, do we need to answer all of the valid questions under duress? If that's the case, then all these numbers kind of slide in one direction.

PAUL VIXIE: So I was probably the person who said let's multiply it by three and call it a day, and nobody knew better, so that is the kind of decision that just went through when it probably shouldn't have.

Obviously, DDoSes have changed everything. And I have a feeling that there's not one Anycast node among the 200 or so represented in this room that can't take 100 times more than its nominal volume before it begins to show stress. I don't know that any of us want to promise to always do that, but three times nominal is clearly an irresponsible number for the ICANN board to agree to if only they understood the issues. I think that we should treat that as the third rail of this conversation and not touch it.

BRAD VERD: I'm not trying to touch it, I'm trying to say that if we're trying to figure out if all things being equal, this is where the number is, what is the goal of our – trying to think of how to phrase this question. What should that number be under duress? Not under normal load. And should we address it like that or not? Is that unreasonable of me to ask?

PAUL VIXIE: We regularly see, like with the Mirai botnet, terabits of traffic in the hands of very low-skilled, low-energy attackers. So I don't believe that any of us can provision enough upstream capacity with enough paths play to be able to even receive a normal packet that is not part of the DDoS when somebody doesn't want us to. And the only reason that they don't carefully map out our infrastructure and attack all 200 at once is because that would not make them any money. And if they could get somebody to pay them not to do that, then we would be under regular attacks of that form. So no, I don't think that we should talk about duress capacity. I think that has to remain an unspoken best effort.

BRAD VERD: Just to be clear, never have I stated, nor would I ever state, that we should publish or talk about capacity. I'm talking about, is it something we think about here during our availability? Because our availability in peacetime is probably a lot less than what it is under wartime, let's say.

PAUL VIXIE: If there was a guy with a checkbook that Jeff could track down here, they would end up with some contract and in that would be something having to do with acts of god, force majeure, and everything you've said about duress is in that paragraph. I think we need to be talking to our insurance providers about it, not writing a document about it.

FRED BAKER: I wanted to respond to your comment, Brad, and I think it's a reasonable question to think about and discuss here. The thing that seems like a direction that we could go with it without explicitly saying it in the document is that this may become part of what gets eventually defined as good, as part of good, not as part of minimum. Now, we may not choose to do that, but that's a path that we've already kind of laid out in the document.

BRAD VERD: Yeah, I agree. I think kind of adding to what you just said, there's the reasonable man test. If there is an event that has negative impacts on the root server system, the question that will be asked from the community and the press and everywhere else is the reasonable man test. Did we, root server operators, reasonably spend enough on our resources to protect the system? I don't know how to put that into an availability number. That's all I was trying to do.

DUANE WESSELS: Okay. Thanks, Brad. Okay, so earlier, we were looking at this table and saying, okay, K=8, five nines gets us 96%. And Paul, you said that number has a marketing problem. Was that because it's not round enough or because it's not close enough to 100?

PAUL VIXIE: The latter. So I think if you have a number that will make sense after you get a chance to explain it to the people who object to it when they hear it, then you lose, because we're not going to get that chance. They're not going to read the paragraph that follows that explains how this multiplies out to a good number for the average end user. They're just going to say "You are giving yourselves an enormous carveout here and we don't like it."

LARS JOHAN LIMAN: So if we go back to my earlier statement and make the calculation from there, if we allow ourselves to make one mistake per time unit, whatever that is, and we take the K as design input K equals 8, what result do we end up with then in the table? As a final value for the entire system. First step, what value do we arrive at for the individual server, and what does that give you for the system as a whole? And I think we are home free there.

DUANE WESSELS: So your proposal was to say that the RSO – individual availability threshold is, I think, 15 minutes of downtime per month, but not 16 – or 29 but not 30, or something like that.

LARS JOHAN LIMAN: 15 is a good number. I'm actually happy to have it either per week or per month, but not per day, because it'll be too much if you do it per day. I don't want to be able to do this 15 minutes of downtime every day and still be under the bar. But once in a while – we can haggle about the while, but day is not appropriate, I think.

DUANE WESSELS: Okay. So I agree that is an approach we could take. One concern I would have is I don’t think it necessarily counts for what we talked about before, which is there are parts of this system under test that are not under your control, there is ISP [networks or] exchange points or something like that. So by the nature of the way these tests are designed, their availability is being measured as well. So it’s not just your downtime or your mistake. So maybe build that into ...

LARS JOHAN LIMAN: Sure, but how do you see that fit into the picture? I agree, it is a part of it, but what effect does it have?

DUANE WESSELS: I would have to look at the specific numbers. I don't know.

PAUL VIXIE: The way it affects it is that I think a lot of these discussions are coming based on something that Paul said, which I think we would all want, but not necessarily doing, which is if I'm going to pull down an instance under known circumstance, I'm going to withdraw its route first.

So if that’s available to every RSO, then really, all we are talking about is I've screwed up the routing for all or many of my nodes, or someone outside of me is doing that. It gets dicier because some RSOs are using other services that they don't control the routing on as well right now, such as – I won't say it because we all know what it is – who also have competitors who might be as clown car-ish or whatever. But that will be affected and we know that's affected if you look at the table that I gave there in the back. We know that some of the vantage points that I just happened to pick for this are obviously next to one of those because the RTT is less than a millisecond. Right? Whereas the normal one is at least 20. So we know that that is the case, and that brings in more risk for the people who are using those, and so I agree with Duane, I think that if you're going to say "I want to have the rationale for the metric is I have made this mistake this often," I think you also have to estimate somebody else making a mistake for you that is causing the routing to go bad. I don't know how to do that, but I also don't think that's impossible.

LARS JOHAN LIMAN: And to me, what I hear, without doing this on paper and numbers, what I hear is that the downtime should be allowed to be bigger than just the mistake by the operator. And I'm fine with that.

PAUL VIXIE: One thing I was going to say relative to – especially at the RSO level for someone else causing an RSO a problem that was in their provider path. It's kind of sort of a supply chain issue, and I don't know if it's a practical thing to say or not, but if this occurred to an operator, it's then incumbent upon the operator to find a more reliable way to do the upstream thing. So it's not totally out of the operator's control.

LARS JOHAN LIMAN: So let me throw the hand grenade. Sorry, Anand, but the RIPE NCC just messed up the RPKI system. How do I pick a different one?

UNIDENTIFIED MALE: Well, in that case you go to one of the other RIRs, because they're all 0.0.0.0.

DUANE WESSELS: Okay, Paul?

PAUL VIXIE: So even if we all had RPKI completely set up and most people were doing path validation as well as origin validation, we would still have some parts of the network that would be susceptible to BGP poisoning, and that is force majeure. We should note it in passing, and say that we are planning to have best practices as far as reducing net risk surface, but it will always exist no matter what we do. And we should certainly be transparent, make sure that anybody who wants to know that this problem exists and cannot be solved will learn it from us if from no other source, but we should not otherwise spend time on it.

BRAD VERD: I feel like there's a couple of those risks that maybe we should document somewhere. I don't know, but I just keep – I hate to say that, I just kind of say what he just said. This risk will always be here. We will continue to improve the system to mitigate it, but it will never go away, much like the botnets. There's no way each root server’s going to have 7 terabits or whatever to handle everything on a bad day, but there are things we are doing, like Anycast, bla bla, that mitigates it because it localizes the damage. That type of stuff.

But it sounds like we should maybe iterate through these so that we can point to them.

UNIDENTIFIED MALE: Brad, do you think they apply to individual of the metrics, or collectively to the entire set of metrics? Or do we know enough yet to say that one way or the other?

UNIDENTIFIED MALE: My quick answer to that is it applies to all the metrics other than correctness. If there is a break happening, you the root server operator may not be able to update all of your instances with the new root zone that you're handing, for example. And if that happens to be one of the ones next to the vantage point, then you're going to look really late even though you were trying.

So the reason I hit the button here, Duane, was we already have a note in section 4.8 – or I threw in a note this morning saying add the ability for the measuring party to be able to exclude some vantage points if they are impacted. That might need to expand to force of god, although some of us don’t want to use that term. If it would make it easier for this work party to come up with some numbers that could exclude force of god, I think it’s reasonable for the party that is measuring these thresholds, the ones who are going to take the threshold, and if they see a red, they're going to do something about you.

And we've talked about this in previous meetings, is you, Duane, now represent N root, and you’ve got a red for this, they come to you and say, “You’ve got a red, we’re going to throw you out.” And you can go, “You know what? What was happening was X.” Now, if it’s Liman's “I typed in the wrong routing thing and I never saw it,” the party might say, “Sorry, it’s still red, you are out.” But if it’s “The ISP there did and I couldn’t get to my router to ...” like I saw it but I couldn’t withdraw the route for a day because my ISP had screwed that up as well, it’s perfectly reasonable for them to say N root is allowed to still continue to be a root server operator.

DUANE WESSELS: And certainly, RSSAC 037 has that kind of language in it already. I think it’s not as cut and dry as one strike and you're out. Liman.

LARS JOHAN LIMAN: Two comments. One is that we’re starting to drift into another set of documents, which is the ones that will govern how the performance and measurement function is to operate. That’s a different issue, but I do agree it touches on how we define these metrics.

The other one is that with the experience from the CSC and how the IANA operates, I see these monthly reports and have exactly that situation where there are reasons for why the IANA doesn’t meet the levels that are specified, and it’s explained in the report and it’s in the procedures how we’re supposed to deal with that, and it actually works very well. So I would expect the same to be the case for the [PML] function.

DUANE WESSELS: Alright. So about four minutes until our brake time, and I feel like there's been a couple of suggestions for how we can sort of make progress here. One is we had the proposal for choosing K equals two thirds of N minus one. Liman was abdicating for something a little different which was downtime-based. One concern I have, I guess, is that we went to the exercise of –

LARS JOHAN LIMAN: Orthogonal.

DUANE WESSELS: Orthogonal, okay.

LARS JOHAN LIMAN: Yeah, because we cannot back into the system.

UNIDENTIFIED MALE: Mic, please, we want to get you recorded.

LARS JOHAN LIMAN: Thank you. It’s orthogonal because we can't back into the system. We have to design the K parameter regardless, and it sounds like we have an approach for that and that is perfectly fine.

When that’s done, then I propose that we do the downtime analysis and we take into account what other parts of the system routing and service providers and so on, and come up with a number from that, we back into the system to find the optimum number we need for the individual nodes. That was my proposal. But we need to define K.

DUANE WESSELS: Okay. So then the only concern I have, I guess, is that we discussed how [inaudible] exercise and we came up with 96% and people felt that was – to quote Paul, it had marketing issues, so I don't know how strongly you feel about that. I guess I would feel a little bit unfortunate that we did sort of a math-based approach and then we didn't like the answer that we got out of it and decided to do something different. So clarify, please.

PAUL VIXIE: So my proposal is that whatever it do be in one sentence. So some subordinate clause, comma, so we might say each Anycast node must have an uptime of at least 96%, in order to ensure that that address is available 100% of the time statistically speaking.

So we put it all together so there's no way to digest only half of it unless you're a hostile newsperson who wants to selectively quote. Then we can probably get away with 96%. But if it's a footnote, it won't get read. If it's the next paragraph, it won't get read. So this is a marketing concern only. I have no objection to 96% because I heard how you justify that. Others won't.

DUANE WESSELS: Alright. So I'm sensing rough-ish consensus for this approach. I guess we'll take it back to the work party and we need to write up some text around this. Go ahead, Brad.

BRAD VERD: I'm sorry, I just heard something different that is very pleasant to my ears. I just want to make sure I heard it properly, which is 96% availability for the individual instances and 100% availability for the RSO.

DUANE WESSELS: I think we would say five nines.

BRAD VERD: Okay, sure. Five nines.

UNIDENTIFIED MALE: Wait. That's not what I heard. So, okay, Paul, were you talking about instances, or were you talking about RSOs?

PAUL VIXIE: I was probably describing it the wrong way, because understanding we have Anycast nodes, we have names, and then we had the system as a whole, it's the system as a whole that needs five nines, and I believe that the way Duane explained the math is that at 96% for an individual Anycast node, we get five nines at the RSS layer. And if I said RSO layer before, I apologize.

UNIDENTIFIED MALE: [That's not what Duane said.]

UNIDENTIFIED MALE: Do you mean Anycast instance or RSO, Paul? Because like an Anycast instance down to the thousand that exist, it's crazy to try and micromanage those. I take mine offline for two days sometimes because I'm doing maintenance upgrades. It's not a problem because my address, my RSO is still 100% available.

PAUL VIXIE: Yes, and we've kind of said several times that this implies that you're going to make sure that your route is not being advertised when a given node is down. So we're assuming that that is not part of our percentage of downtime and that what we're talking about is the percentage of time during which queries that reach us will not be answered.

UNIDENTIFIED MALE: Alright, then phrase it carefully as currently up Anycast instances, or currently routed. Yes, okay.

DUANE WESSELS: Another wrinkle on this is that we were talking about N=13. You can maybe consider N=26 if you consider V4 and V6 together, in which case things only get better. If we plug N=26 into this formula, the system availability looks even better.

UNIDENTIFIED MALE: [inaudible].

DUANE WESSELS: They can be independent.

UNIDENTIFIED MALE: [inaudible].

DUANE WESSELS: They don't have to.

PAUL VIXIE: There are recursive servers who when they hear a serve fail from any of the addresses associated with a name will assume that that name is a multi-homed host and that it's going to get serve fail if it repeats the question to other addresses belonging to the same name regardless of what protocol is involved.

So I realize there are recursives that don't work that way, but we should not be treating a AAAA and a A as two Anycast nodes, because it might not be. Some of us might have implemented it that way, some of us might not have. So I think it would be gross overestimation to say that we had 26 because they're all dual homed or dual protocol.

DUANE WESSELS: Alright, thanks. We're a little bit over the break time, so we have more – after the break, we're going to continue talking about thresholds. We'll try to get back into the RSO thresholds again for some of the other metrics, and maybe we'll wrap up our understanding of this one when we come back from the break. Does that sound good? Okay.

Alright, Ozan, you want to – we're good, okay. Hello, everyone. Welcome back to the session four of today. We're going to continue talking about thresholds for some of the RSO metrics. First I wanted to try to resummarize where we left off at the end of the last session.

We had been talking about the RSO availability threshold, and decided that maybe one way to approach it is to settle on the RSS availability parameters first and see what that tells us about RSO threshold.

So we looked at the formula for parallel availability and said, "Well, let's make our target RSS availability five nines" I think is what we said, so almost 100%. We'll go with five nines, and per the discussion from Paul Vixie, we said, well, if a querier to the root server system, the first attempt they get a time out, the idea was that there should be a one out of three chance I believe you said, Paul, that the second query would be successful.

PAUL VIXIE: I wanted two out of three.

DUANE WESSELS: One in two out of three?

PAUL VIXIE: No, I wanted –

DUANE WESSELS: You wanted two out of three. Okay, you wanted two thirds.

PAUL VIXIE: So eight out of 12 being up is good.

DUANE WESSELS: Okay, thanks. So yeah, two out of three chance that the second query is up and since we have RSSAC – I forget the number, 02 something which says that the system can survive with one server being gone, we're setting our value of N to 12 and two thirds of that is eight. So by that, we settled on K=8, meaning eight root servers are required for the root server system to be operational, and when we plug those numbers in, we get an individual root server availability of 96%.

LARS JOHAN LIMAN: Individual root server operator.

DUANE WESSELS: Individual root server operator, yes, using the terminology in the document, RSO. An RSO availability of 96%.

LARS JOHAN LIMAN: Required availability.

DUANE WESSELS: Right. So that's how we –

UNIDENTIFIED MALE: As measured by ...

DUANE WESSELS: As measured by this metric in our document. So the metrics work party document has a method for measuring individual RSO availability.

UNIDENTIFIED MALE: On a network to be built.

DUANE WESSELS: On a system to be built, yes.

UNIDENTIFIED MALE: And did we decide to change it from daily to weekly or monthly?

DUANE WESSELS: I think that’s still a discussion to be had. That is something either to change it from daily to monthly or to add monthly in addition to daily.- Brad, go ahead.

BRAD VERD: Just to add to the recap, I think Paul Vixie said that with the 96th percentile, we have a messaging challenge.

UNIDENTIFIED MALE: Yeah, I debated whether to say [inaudible] but as I sat there and listened to your recap and heard it all in a relatively short time, the 96% certainly sounds kind of glaring. We all know how we got there, how the math works, that the chain of reasoning makes sense to us here in the room and is, I think, sensible and defensible, but I have to agree 96/ just doesn’t sound real good.

BRAD VERD: Maybe if we add up those force majeure risks and the reasonable man theory and whatnot, there's some sliding factor that you put in there to say the math says 96 but we believe it’s 98 when you put these factors that really you can never completely mitigate against but you spend a certain amount of resources to try to mitigate against. Something like that. I don't know.

DUANE WESSELS: Go ahead, Ken.

KEN RENARD: Yeah, just to maybe help everyone’s conscience here, yeah, 96% does not sound good, but that’s the next set of numbers down there. The good section is the 100%. So I think we have something defensible. Yes, it doesn’t sound good, but we’re talking the minimum requirements.

DUANE WESSELS: And also, Howard was bringing up in the discussion about the results you see from DNSPerf.com, and I've looked at the RIPE Atlas measurements quite a bit recently, and in RIPE Atlas I see similar things to I think what DNSPerf sees, which is typically, they report this availability in the 97% range, and even actually lower for v6. So I know that maybe the system that is yet to be built for RSSAC metrics will be held to a little bit higher standard, there’ll be fewer probes so that'll maybe be simpler, but I think the reality is that it’s going to be in that range. Do you have data, Paul, in your [inaudible] concept yet?

PAUL VIXIE: Not currently because I'm in the midst of changing it the way that we described it. What I was raising my hand about was even though we've just been talking about availability, remember there's four availabilities, and we have skipped over that, and yet we know there's differences. Just wanted to put that out there.

UNIDENTIFIED MALE: Is it just me or do we feel like there are deltas in the available methods of measurement and what we think to be the case, or at least possible ones? And do we want to call that out? Because if we're saying we are god's honest only going to be up 96%, it sounds horrible, whereas if we're saying – if we're going to be measured by systems operating in the incredibly complex environment of the actual functioning Internet, be aware that due to humanity and humidity and difficulty, the numbers are going to look like this. I'm just wondering whether we don't need to couch that a little. Because in a perfect world, 96 is horrible performance, but it's starting to look like that's just what we're going to get with measurements. What do you think?

BRAD VERD: This is just an observation, we continue to have the discussion about a second set of numbers but we just don't want to document it. We all say 96 is not a good number. It's the minimum. But we don't want to put in the document what the good number is, because we only want to talk minimums.

DUANE WESSELS: I think we want it both ways.

G1 if I recall correctly, what Paul Vixie was describing was 96% for an Anycast instance that resulted in an RSO having 100% –

LARS JOHAN LIMAN: No.

UNIDENTIFIED MALE: That was not it? Okay.

DUANE WESSELS: So nothing in the metrics work party is proposing measurements on individual Anycast instances. All of the queries go to the service address of a root server operator. The routing system does its job and delivers the query as it does, so we will have no visibility – especially, we're not going out of our way to try to identify individual instances of a root server operator. The most fine measurement level is on an individual RSO or a nameserver.

UNIDENTIFIED MALE: Thank you for that. So thinking about the granularity of your test, your probe sources, it has to be an order of magnitude more granular than the catchments themselves. In other words, if, say, RIPE has got 50 nodes out there, you're going to need more than 50 locations to make sure that you are actually talking to all 50 of theirs. I'm not sure you need 500 of them, but you need 100 of them if they've got 50 different Anycast nodes.

Have you figured out what your probe density has to be in order to make sure that you're testing a reasonable subset of the Anycast nodes?

DUANE WESSELS: We've talked about this a little bit, and I actually have some data from a simulation that I did and I can share, but again, within the work party, we've sort of settled on the idea that at least to start, there would be 20 vantage points, 20 probes. Realizing that that's not going to cover the entire system.

UNIDENTIFIED MALE: A follow-up. That seems like a reasonable starting point as long as you can describe the conditions under which you know you're not getting enough information so that when that happens, you'll know that you need to increase your density.

ABDALMONEM GALILA: I'd like to ask [inaudible] percentage for root server availability relative to the number of Anycast nodes [inaudible] measurement?

DUANE WESSELS: [I didn't catch the last – can you repeat the last part?]

ABDALMONEM GALILA: Yeah. What I would like to say is that, is this percentage of availability relative to the number of nodes nearest to the probe who make the measurement? Maybe I make the measurement from my location here in USA. If I have nearest nodes maybe 100 nodes, I will get a high percentage. When I'm far away from these Anycast nodes, I will get low percentage.

DUANE WESSELS: The system that we’re recommending does not attempt to do that. The thing that is being measured is the service address of a root server. So for A-root for example, it’s 198.41.0.4. We send a query to that IP address and get a response from that IP address. I don't care which instance answers, we’re not trying to count how many instances there are.

ABDALMONEM GALILA: [inaudible] reflects to the availability of the root server.

BRAD VERD: [inaudible] [Do you want to try that?]

UNIDENTIFIED MALE: Sure. I'm trying it because I actually came to this at the beginning before we had gotten to the “we were only doing one.” It might or it might not. And because we aren't even sure whether what you just said is true for all root servers, we went all the way to the other end of saying we’re just going to look at the service address. So for example, let’s say that there's a vantage point in this room and there happens to be an instance in that room of N root.

It doesn’t matter whether N root then doubles their number of instances. This vantage point is always going to have the same latency and the same uptime. So the fact that different root server operators have chosen to have more or fewer, dispersed or local, on islands or only in datacenters, actually, since we can't generalize that, we went to let's go as ungeneralized as we can, one address gets one measurement.

ABDALMONEM GALILA: [inaudible].

UNIDENTIFIED MALE: It could be minorly, but again, each vantage point is probably going to latch on to approximately exactly one instance of a particular root server. So if they add a whole bunch more, unless they add a whole bunch more right in the next room, the latency is also not going to matter.

DUANE WESSELS: Alright, I'm going to try to move on to some of the other thresholds that we wanted to talk about. Latency. Yes, latency should be fun. So this is about RSO latency, response latency, I should say. This is again over the course of a day, maybe in the future a month, but now over the course of the day aggregate all of the individual latency metrics and derivate the median value, and then apply the threshold to the median value.

If you look at the spreadsheet, the proposed thresholds range from I see one second, 250 milliseconds. What else is there? That's essentially the range I see between 250 and one second.

Some of the answers have different thresholds for UDP and TCP, mine included, because TCP requires one round trip time for connection setup, so essentially the threshold is doubled. Would anyone like to throw out their own threshold or make comments about these thresholds that have been provided by other people in the caucus?

DANIEL MIGAULT: Paul explained the reason they put one second, I guess. I’d like if any other people had any other reasons or reasoning behind to put their value. I used the same kind of thing, but I'm wondering if people had other thoughts to define the threshold they defined, the [inaudible].

DUANE WESSELS: Go ahead, Paul.

PAUL VIXIE: Speaking for myself, I entirely understand Paul Hoffman’s rationale for going around the global at least once.

UNIDENTIFIED MALE: But no more than twice.

PAUL VIXIE: But no more than twice. But in my mind since this is a median, we can do better than that. So that’s why mine is lower, it’s in the range of 250 milliseconds.

UNIDENTIFIED MALE: I think something in this magnitude is right. And the reason is that people are not making – we're hoping that our probes or the probes that are going to be operating in this system are a proxy for real traffic and that the behavior they experience will reveal what kind of behavior the Internet itself is getting from us.

And the Internet itself, when it makes a query, is not just doing it to probe and measure us, they're doing it because they've got some work they want to do, and that means that there is some held state somewhere down the rabbit hole of the initiator which we should treat as having mass and we should not want very much of that to be hanging around waiting to see if and when we're going to answer.

So I think anything lager than a second is out of the question just because of the amount of work that has to be held up waiting for an answer that takes that long from us. I would be fine with 250 milliseconds, I would be fine with the second. I would start to get hanky if it was over a second.

DANIEL MIGAULT: The exact rationale I used is I took the latency measurements in a far away island – I can't remember the name, but I put it in the comment – and I took that saying that considering that it's going to be the highest latency observed around the world. But given that now we have only 20 probes well located, we might go a little bit further.

UNIDENTIFIED MALE: I've put down 500 milliseconds for UDP. I like the rationale of going around the world, but I kind of think we could do better than that. I think somebody else put the justification, just this was documented I think in one of the gTLD SLEs or SLAs. We should at least be as good as them or should we be better? That sort of anchor.

DUANE WESSELS: Yeah, I think you're right. In the case of the gTLDs, I think it's 500 milliseconds. But what I don't remember off the top of my head is if it's also a median value of a distribution. It probably is, or a mean.

One of the reasons I think one second is too high is because this is a median. You could have much longer than that in your distribution and still have a one-second median.

HIRO HOTTA: Hi. May we ask Shinta what he wants to mean by [N/A?]

UNIDENTIFIED MALE: Yes.

DUANE WESSELS: Is Shinta still on the call? Yeah, okay. If I can guess, speculate, I think Shinta had in his e-mail, he suggested – well, that was about the RSS. That was not about this one.

UNIDENTIFIED MALE: [inaudible].

DUANE WESSELS: Okay, thank you. [inaudible].

SHINTA SATO: What I mentioned in that spreadsheet is that the response to the query is important, but the latency itself is less important, that if we think about the minimums thresholds, I can accept for if the response comes before the timeout. That is acceptable for me in this case. I'm not thinking with a median or something like that, but the thresholds with RSO latency, response to it and I'll say like this. IT's in the comment of the spreadsheet.

DUANE WESSELS: Thank you, Shinta. Your comment says that what's most important is to respond to the query, but the actual latency is less important. So it is acceptable if the response comes before the timeout. I guess my question to you –

SHINTA SATO: Depends on what we think for the minimum meanings.

DUANE WESSELS: The minimum, what we're looking for here is the minimum of the median, of the distribution.

SHINTA SATO: That's how we calculate, but what we mean for the median? Minimum, okay.

DUANE WESSELS: The way the metric is calculated is every five minutes, the probe sends queries to the root server and calculates the individual response times. At the end of the day, it should have about 5-6000 of those, and then it takes that bundle and calculates the median value. And then this threshold is a minimum, or I guess in this case a maximum threshold on the median response latency. Does that make sense to everyone?

ABDALMONEM GALILA: Sorry, for one second, there is for column for one second for latency. I think this is a huge time and this is my first concern. My second concern should differentiate between TCP and UTP. TCP maybe you'll have a larger response size of the response and it may take a longer time. So could we differentiate between TCP for IPv4 and IPv6? Something like this.

DUANE WESSELS: Yes, we sure can differentiate those. In the case of TCP, it wouldn't necessarily be because the response is bigger but it would be because there is a connection setup aspect to TCP that UDP doesn't have. I guess I also want to maybe if I can throw a grenade into the room, speaking of marketing problems, to me one second seems crazy. You would tell people that we'll respond to you within a second. If 96% is a marketing problem, I think one second is a marketing problem.

ABDALMONEM GALILA: [inaudible] this domain name or TLD is DNSSEC signed or not.

DUANE WESSELS: Only very slightly. The bulk of the latency is the transmission time, not in the size of the response. And I think as even written, this measurement does not request the DNSSEC data in the response. So these measurements I believe are unsigned, the responses. So I think it's negligible. We don't need to worry about that. Paul, I know you're next.

PAUL VIXIE: I'm actually answering for Kazunori there who said the same as gTLD. And people might find this interesting. So for DNS, in the SLAs, server availability has to be 99%, so 432 minutes of downtime per month. For TCP DNS, it's 1500 milliseconds for at least 95% of the queries. So 95th percentile, 1500. UDP is 500 milliseconds, so they actually gave three times, not twice.

DUANE WESSELS: And I guess both UDP and TCP are at 95th percentile instead of here we're talking about 50th percentile which you would hope is much lower than 95th percentile.

PAUL VIXIE: What Fujiwara-san had said – oh, I'm reading from the boilerplate agreement document that everyone has signed if they're going to be a gTLD – actually, all gTLDs, not just the new ones for these, I believe.

BRAD VERD: Registry agreement.

PAUL VIXIE: Yeah, it's the registry agreement. I believe they've actually wrapped them back to all of them at this point, minus possibly [arrow.]

BRAD VERD: I was going to go into the marketing aspect of a second. I agree it's a problem there. Is it possible or reasonable – and you can say no – in the document to be somewhere to say that our metrics are based on whatever the number, whatever the threshold is? And again, I'm going to go to two different numbers, but our goal would be that an Internet user would experience no more than like 150 millisecond response time to a root – not A root, but an individual root – and should somebody be experiencing more than that, then please work with the root server operators type of thing.

Because it would be nice to kind of state, "Look, given the way the Internet is and the way the monitoring is, all these different things that play into latency, you can't put enough probes out there to get this to be where the marketing numbers are happy, but you could state kind of a number or a goal of something that –" 150 millisecond to one of the 13 seems really reasonable.

DUANE WESSELS: I guess we can consider something like that. I think if I recall correctly, in the RSSAC FAQ, we have a number like that in one of the answers advising people – I think the question is, how do I know when I need a root server in my region? And I think there's a number there which I think is 100 milliseconds. So we have [said such numbers] before, so maybe we can say it again.

DANIEL MIGAULT: So the people operating or [fitting] higher number or lower number, so my guess is that, is there anyone that disagrees with the 200 or 250 milliseconds and 500 milliseconds? If everyone agrees, done.

The other thing is whatever threshold we're going to put, it's not a document, it's never going to be revised. In crypto, we have an update of the consideration every three years, which is approximately the time it needs to write a draft and get it through. So it's not that we're going to write things for a century ahead. So I think that's a way to address what Brad was also suggesting.

DUANE WESSELS: Yeah, I think that's a good point, although I would note that if a future revision revises these numbers upwards, should be very well justified why it's doing so.

So to Daniel’s question, if we settle on 250 to 500, is that something that everyone in this room is okay with? Does anyone want to argue for the need for a higher threshold than that? Robert?

ROBERT STORY: So if you want to talk about [marketing] and you look at it from the other aspect, if we’re talking about a median of 250, that means there are going to be outliers that are going to get above that. And they're just going to be 250 and they're going to start to complain, “You guys are not meeting the standard.” They're not going to look at that little word, “mean.” They may not understand what mean means. So that makes me a little nervous.

DUANE WESSELS: Can you explain a little bit more? Where would they get these numbers that are higher than 250? Because nothing in the metrics work party is proposing to publish even actual numbers or individual measurements. So they would have to come from somewhere else, I guess, in which case they can already see those.

ROBERT STORY: That’s a good point.

DUANE WESSELS: So you can go to RIPE Atlas and you can get these measurements, and I've done that, and I kind of get the feeling people don’t care, because no one has complained about them so far maybe. I don't know. But again, just to be clear, the metrics document does propose publishing things, but it's only a pass/fail of the threshold. At least for the RSOs. The actual numbers are not going to be published, and one of the reasons for that is because that bullet that says these metrics are not designed to compare RSOs to each other.

UNIDENTIFIED MALE: Just a thought here. I'm okay with us setting the numbers of 250, 500. That seems reasonable. But we're looking [at the constrained way of how we're] looking at the measurements. We have these 20 vantage points which are located in particular regions of the world and then we know what we are measuring, which is essentially the 13 root op, RSO.

So in some way, we have scoped the problem. So there should be a lower bound to this than just saying it's going to be half of what it is to go around the world or something like that. To say what could be the worst in terms of vantage point, trying to reach the instance of a root server operator that is farthest away and then sort of build our – because what we want info from this is when is something looking bad, when is an instance of something that we need to take an action?

So from that point of view, it's not just to satisfy this is a metric and we pass it, but at what point do we believe that the vantage point that we've set up is consistent with how we believe those responses would be returned? So I think we can do better than 250-500 in terms of setting a limit within the bounds of our test framework, but I don't know what the value is.

DUANE WESSELS: I think that's fair. Especially I’d kind of forgotten, Paul just reminded us about the gTLD thresholds which are at the 95th percentile, and I think given that what we’re proposing is median, 50% is very generous.

One thing you said that was talking about vantage points, a vantage point reaching a server that’s very far away, that’s an individual measurement result, but our thresholds have to be on the aggregated median value. I don’t think we want to impose any limits or thresholds on individual measurements. That would probably be disaster.

But if you can come up with a number other than 250, I guess we can discuss it, but I think we need something specific to fill the gap.

UNIDENTIFIED MALE: I'll give you an example from our recent experience. With a low number of nodes, we’re also going to have – probes, I'm sorry, vantage points, we also need to think about the upstream transit diversity because for example we turned up a new node and so a large provider is providing transit for that node and that provider we only have access to at that node.

So anybody else that has that provider as their transit is going to go to that node even if it bypasses both of our other nodes in the path to get there. So that’s an example where we have really horrible performance from distant locations that we can't do anything about because everybody is going to prefer their transit provider for exiting to get to us.

DUANE WESSELS: I guess that makes sense, but do you think that given the fact that these thresholds are unrealistic, or are they okay?

UNIDENTIFIED MALE: Well, you were asking about example of where we would get extreme values that are –

DUANE WESSELS: Just providing a real-world example of that.

UNIDENTIFIED MALE: Right. Yeah.

DUANE WESSELS: Okay. Got it.

UNIDENTIFIED MALE: So for instance, just theorizing again if – I think probably you're referring to, is it Brazil where that node is located?

UNIDENTIFIED MALE: Chile.

UNIDENTIFIED MALE: Okay. Someplace, another part of the world that had that provider, if that happened to be where the vantage point was located, that would be that long example. Have you guys collected any actual data to look at what latency would be to where you could say, “Well, yeah, that’s 250 and 500 is unreasonable” or “It’s probably okay?”

UNIDENTIFIED MALE: I think the highest latency I've seen in getting to that node from somewhere I think in Asia was on the order of 300, 350. So I think 500 is fine, and again, it depends on – we have the same problem with another node that has a different provider that we regularly get all the queries, again, going from Europe bypassing other nodes to get all the way across the country once it gets across the Atlantic and we have higher latency.

UNIDENTIFIED MALE: Sorry, UDP or TCP?

UNIDENTIFIED MALE: UDP.

UNIDENTIFIED MALE: Okay.

UNIDENTIFIED MALE: I feel like this was brought up before but I missed it because I'm really processing slow here. To the degree that we get measured on things and there are hits that are base on external factors, to what degree do we expect those to be something we can do?

For instance, we noticed a big failing recently on F root that is not measured by the Atlas probes but is shown as a really big hit on whatever I was looking at, DNSPerf, and it turns out it's probably part of a route leak that was a very aggressive Pakistani action, known, bla bla, whatever, but it hits and shows up as an absence of the node being live or responding.

So what is that? is that force majeure, an act of god? Is it something that's [an] adjustment or is contemplated as such?

UNIDENTIFIED MALE: I think both of these cases are dealt with by the fact that there are 20 probes, not two, that we know that the 20 probes, that unless all of your instances are on networks that are having those problems or whatever – I'm sorry, unless all the vantage points are on those – that's going to get averaged out by, again, we sort of chose then number 20 as a large enough number distributed across the world – unfortunately geographically not network-wise – to reduce those.

So when you look – let's say that you on a certain day said to the collector, "Hey, we want to see our data, we know we're passing but we want to see our data," you might see spikes. But it's likely you will continue to pass because unless all of them were maniacally on something that was bypassing, or in your case, was on something that –

UNIDENTIFIED MALE: It's hard to revert to 99 and change though. [If you don't revert to that, you kind of –]

UNIDENTIFIED MALE: I'm sorry, hard to revert to ... I thought we were talking about latency here.

UNIDENTIFIED MALE: I think I'm talking about availability.

UNIDENTIFIED MALE: Oh, sorry. Okay.

UNIDENTIFIED MALE: DNSPerf which I hadn't heard of until two hours ago.

UNIDENTIFIED MALE: Okay, we were on latency, but my answer does not [inaudible].

UNIDENTIFIED MALE: Okay. There was a blank spot. And like I said, I might be going back a little, but when these measurements show anomalies that look like bad performance, to what degree is the malicious act of a bad actor mitigating factor? Any? It just never occurred to me before.

DUANE WESSELS: Right. It's not the intention, Jeff, that the metrics would always – that you would be required to meet the metrics, the thresholds in all possible situations, right? So something like that, the intention is that you have a way to explain what happened and it doesn't sort of – it's not a strike on your record. But you can help us get the language in the document right to make sure that we capture this appropriately to cover the cases. I'm not sure that we're there yet. I'll take a read through it and see what it says, but if it doesn't sufficiently give you that out, then we need to add that into the document.

UNIDENTIFIED MALE: I'm sure we're all paranoid enough to imagine a scenario in which we're a little bit concerned that we need to hit a performance level and then somebody comes along and puts their thumb on the scale and our performance does not come up – that's – you can't talk about things like grades and passing and failing without worrying about failing.

UNIDENTIFIED MALE: Yeah. Take a look at section 4.8. That's the current one that deals with this type of anomaly, unexpected results.

UNIDENTIFIED MALE: So when we're talking about force majeure due to malicious actors, we're predicting some trouble on SLA negotiations and actual contracts or payments because it's very difficult to prove that the reason that you were not responsive from somebody's point of view is due to no fault of yours. I believe that that negotiation is going to end up including audit rights over the other guy's net flow and so forth so you can see the packets they saw and compare them to the questions that they didn't answer from your probes. So it's an information paradox involving other people’s networks, but I don’t think it concerns us much here. I think from the point of view here, we’re just trying to say that our performance is in terms of answering the questions that we hear, and if we don’t hear them all because of force majeure or our answers are not heard because of force majeure, that’s kind of off topic at the moment.

ANAND BUDDHDEV: Listening to Paul and other people, it just occurred to me that all the measurements we’re doing are only DNS measurements, and we’re not doing things like traceroute and other supporting measurements which might or might not assist with determining whether a particular measurement is flawed or not.

I'm sorry if I'm late to this discussion. And I don't know if something like this was discussed, but this might be useful, because when we use RIPE Atlas probes for doing DNS measurements, we often look at the associated traceroutes, and sometimes they reveal interesting things. So I don't know if this is worth considering and adding to the document.

UNIDENTIFIED MALE: [inaudible].

DUANE WESSELS: Yeah, this section that Russ referenced, it does very briefly mention traceroute in the context of unexpected results, so this is not specified in a lot of detail, but the idea is that if you do get unexpected results, then you might do a traceroute and record it.

if people think that the system should be doing traceroutes on an ongoing basis, we can consider that as well. That would be a little bit more of a data collection burden. So finding that balance is right, but please take a look at section 4.8 and make recommendations for how to make it better along those lines.

Okay, where were we? 250-500 ...

UNIDENTIFIED MALE: [inaudible].

UNIDENTIFIED MALE: Just looking at the numbers I've gotten so far, you don’t need to double for TCP.

DUANE WESSELS: You're talking about in your prototype?

UNIDENTIFIED MALE: Yeah, in my prototype, except for a couple of the root server operators. So I'm not saying don’t double. Most of them are in fact like barely more than 10 milliseconds more, but some of them seem closer to double. Not 250 and 500, but just if people are saying, “Why are you doubling?” It isn't really double for a lot of them, but it is double for some. And once I have more numbers, then maybe those will switch.

DUANE WESSELS: Okay. Yeah. Thanks for this data point. Should we try to move on to the last one? Okay. So the last one we wanted to cover today is the root server publication latency threshold. A lot of the responses here, I guess they're either one hour, 12 hours or 24 hours. There's quite a lot that are one-hour.

UNIDENTIFIED MALE: 24 hours [was used.]

DUANE WESSELS: Oh, that's the – well, somebody says one day. There's a coupe that say one day over there. So as a reminder, the way that this metric is proposed is there's a central processing system that sort of looks at all the [SOA] serial numbers and then from that, it can calculate the time that a new zone was put out, and the time that the root server started picking it up and publishing it.

this, again, is proposed as a median of those, all the times collected over the course of the day. I think an argument for one hour is that when we have sort of important changes to the zone, like somebody does a DNSSEC change that's important to them, maybe a key rollover, there was an example just recently where some TLD had changed their keys and was having problems. In cases like that, you want pretty low time, like one hour.

At other times, the root zone is kind of stable and doesn't really change a lot. There's not necessarily those critical changes, and in that case, you could maybe tolerate a longer publication latency. But I think that probably, we should focus on the more important case of changes that really matter.

Open up for comments on one hour, 12 hours, 24 hours. Robert, go ahead.

ROBERT STORY: I've put two hours because I agree with the comment that Mauricio had in his, 12 hours, which is that two updates a day, if you miss one notify, that might be okay, got lost in the mix somewhere. But if you miss two, then it's an issue.

DUANE WESSELS: On the other hand, the [SLA] refresh parameter is 30 minutes. so you shouldn't have to necessarily rely on the notifies, you should be doing the refresh anyway and you should pick it up within one or two refresh intervals, I would say.

And again, this is am median value, so half of the instances need to be at this level, the other half can be one week and you still meet the threshold. Not that that would happen.

I think we're all losing interest in talking about metrics. Everyone but me. I'm so excited about it.

UNIDENTIFIED MALE: I had one question that came to mind. I was going to ask Naela but she's not here right now. The other – what I would possibly describe as a critical distance besides a TLD key roll would be if a TLD changed all of their nameservers in one fell swoop. I don't have any idea how often that happens. Not at all. But does anybody have a clue?

UNIDENTIFIED MALE: Too bad Naela's not in the room, but I don't think IANA would let you do that. I mean they will try very hard to stop you from doing that. I don't know if they would relent if someone absolutely insisted, but ...

[SUZANNE WOOLF:] It's not happening except an emergency of some kind.

UNIDENTIFIED MALE: Right. The only thing I was thinking that might occur would be if there was some – I forget what the right terminology. Somebody had control of a TLD and the ICANN board agreed that it shouldn't belong to that entity, it should belong to another entity. But I can't imagine that that wouldn't be well coordinated and timed and so forth. That's the only one I could think of that might fall in that category.

BRAD VERD: We had a number of instances of emergency root zone changes during the DNS hijacking that quite honestly, according to the people calling in to get those DS records changed, an hour would have been unacceptable to them. So if we're considering our customers, that's one point of view.

UNIDENTIFIED MALE: One of the things thrown out there was if you're operating within [let's say a week old] zone and the way it's written, a year old would still pass the median test, should we have a 95 or 100% threshold.

DUANE WESSELS: You mean a threshold on the 95th percentile of the distribution of the ...

UNIDENTIFIED MALE: Right, which might be the week or much larger than the 50th percentile, but ...

DUANE WESSELS: Maybe. And also, of course in RSSAC 002, there's a similar thing that's self-reported, and there we use 95th percentile as the reporting [inaudible]. So that would be nice to align those two things maybe. But it would be the only thing on this document that's not median at this point.

UNIDENTIFIED MALE: Right, so the median's more how fast you do it, but how much you're dragging your feet. That's important, I think.

DUANE WESSELS: Yeah, I think if it was 95% instead of median, I think I would change my answer here. I wouldn't use the same thresholds. But we might want to think about it differently, or ...

UNIDENTIFIED MALE: I'm almost proposing a median threshold and a 95% threshold, or 100 for discussion. Maybe the week or something, an order of magnitude or so higher than the median.

UNIDENTIFIED MALE: But how would you measure that with only 20 probes?

UNIDENTIFIED MALE: Well, you can. Oh, I see. That's right, you still can only hit the sites that you can hit, but you can still calculate 95th percentile of the data that you have. But you're right, your coverage would be limited by the number of probes. But again, on the other hand, we do have the self-reported metrics from RSSAC 002.

UNIDENTIFIED MALE: Since the [RSIGs] have expiration dates, serving a zone past any RSIG's expiration date would be fatal. In fact, that's how people discovered, I think, if I understood the original report for the C-root not updating, that it was somebody who was on that threshold saying that. So I don't think we should allow anything past like the shortest of the expiration dates.

DUANE WESSELS: In that case it wasn’t a signature problem, it was a key problem. The root zone signatures are designed so that they're always longer than the [SOA] expire.

UNIDENTIFIED MALE: Right, but there's also signatures – is that true for all of the signatures in the zones? Okay, yes. Then [inaudible].

DUANE WESSELS: Their validity period is longer than a week.

UNIDENTIFIED MALE: Okay, yeah.

DUANE WESSELS: It sounds to me like where we are is to the extent that we would like to have a threshold based on the median of the distribution, one hour is about the right range. If we want to consider adding a second threshold at, say, 95th percentile, then we would need to open it up for discussion for a second threshold.

Are people comfortable with having two thresholds for this metric, t wo different thresholds?

UNIDENTIFIED MALE: [inaudible].

UNIDENTIFIED MALE: Or just make it one hour and use median. Would you be comfortable with one hour at [mean?]

BRAD VERD: [How can you not be? I'm sorry.] If one hour is the median, how could you reasonably object to that? Let me put it that way.

DUANE WESSELS: Is this one where we do not have a marketing problem? We're good marketing-wise?

UNIDENTIFIED MALE: Well, we do have several people both in the room and on the call that did say longer than an hour. Let's ask the question if anybody that was saying longer would have a problem if we said one hour for this. [inaudible]?

UNIDENTIFIED MALE: No.

DANIEL MIGAULT: So when I put an hour, I meant that the different instance would have different version of the zone. But yeah, obviously, I don't have any problem having less than an hour. It's just what the hour meant at the time, I [inaudible] because I have the impression it’s a little bit different. But I have no problem.

UNIDENTIFIED MALE: Okay. Thank you. So one hour, sounds like we've got agreement.

DUANE WESSELS: I think we've more or less accomplished what we set out to accomplish in the session and I'm sensing a lot of tiredness, so we should probably wrap it up today.

UNIDENTIFIED MALE: [inaudible].

[SUZANNE WOOLF:] [inaudible].

UNIDENTIFIED MALE: [You were going to wrap up between 4:30 and 5:00.]

DUANE WESSELS: I was?

UNIDENTIFIED MALE: [inaudible].

DUANE WESSELS: Carlos, [you're not in the room?] I noticed Naela came back. Do we want to ask Naela the question that we were asking when she was not here?

RUSS MUNDY: Naela, we were looking at in terms of the latency for an update, and one of the questions that came to my mind, is it ever something that occurs when all of the nameserver records for any given TLD are changed at once?

NAELA SARRAS: Yeah, thanks, Russ. Yes, it does. It happens. It’s actually not that rare. And generally, the root zone management system advises against it, it says this is a destabilizing event, you should consider staggering your nameservers. But if they push ahead and provide reasoning why, we let it happen.

[For] the gTLDs go through a whole change process, let’s say they change the backend registry operator and they go from one organization to another, they have a whole plan for how they introduce the new nameservers and have them parallel for a while, and then eliminate the old nameservers.

And that works pretty nicely. With ccTLDs, we often see cases where they're changing from one operator to another completely, and once that change goes through, they really want to just get the change done. Again, we advise against it, but we do do it. And we haven't seen any cases where the TLD goes dark or anything like that. It’s not a favorable action, but it does go through, yeah.

RUSS MUNDY: Okay. Thanks a lot, Naela. What we were really – our discussion was, is 12 hours or 24 hours an okay median time? We settled on one hour, so I think it lessened the impact of that a lot. Thank you.

DUANE WESSELS: I just want to follow up, Naela. Maybe you have some insight into expectations of TLD managers regarding root zone changes. Do they expect them to be made – I guess the best I could expect is with the same day, right? Except for the emergency changes.

NAELA SARRAS: Yeah. They expect them as quickly as possible, of course, but as Suzanne is saying, that's true, there is SLAs built in for each part of the process of how long something can take, up to X amount of time. It can't happen in one day unless it's an emergency, because there's different steps that we can't control. For example, each change request has to go through approval from the administrative and technical contact, and we can't impact that. They might take a day, they might take four days, four weeks.

So yes, my expectation is they come and they want it done ASAP, but a lot of that time is their time. [inaudible] Ryan Stephenson here. But yeah.

DUANE WESSELS: Alright, thanks.

BRAD VERD: If I could just add to that, she just described this process that they get very impatient because of the time it takes up, and we're not even at the point of publication yet. So once it gets there, they're like, "I want it now." Is that a reasonable thing to say?

NAELA SARRAS: Absolutely. Once we say we've submitted it to the root zone maintainer, they say, okay, so when will it be in the root zone? And [they're like looking at their watch.] So yes. Part of it is I think the community doesn't also understand the provisioning and the publication. I think part of it is that misunderstanding. But yeah.

WES HARDAKER: Do they have an understanding of TTL values and the fact that TTL values in the root are actually two days long and that there's a significant rollout with that?

NAELA SARRAS: Yes, more and more, I see a lot of this, especially in the gTLD world where they are changing backend registry operators and they understand that they need to stagger their nameserver changes to allow the TTLs to expire before their new nameservers start responding. I feel like there is a good understanding of that.

UNIDENTIFIED MALE: I would say yes, there's a good understanding, for exactly the opposite reason that you gave, which is if you have changed a ccTLD's – all of their NS records and you didn't see them go dark, that means both operators were continuing to serve them for however long the TTL was. So regardless of their desire of, "Oh, it's the new one who's real," if they didn't go dark, the old one was still serving.

WES HARDAKER: One other side point, we actually just are publishing an academic paper on studying the TTL values, and there's a lot of interesting data in it that shows some resolvers prefer the client's published TTL over the parent's, and some actually prefer the parent's. So even though like .ua has a very short TTL that they publish, some clients will actually cache it for longer than the client is. And the other thing is that we found that the A record and the NS record are often bound together in some cases too, especially when they're in bailiwick versus out of bailiwick cases. So TTLs are much more confusing, I think, than the average operator understands.

BRAD VERD: If I can just add one other thing. And Naela, you can back me up or correct me, which is we keep using the full NS set being swapped out as the example of an emergency change and we talk TTLs about that.

While I think that has happened and we've seen that, the most prevalent one that I think is a time-based one is the DS records being changed or removed, and not having two DS records in but one being removed and one being added. They want those immediately.

DUANE WESSELS: Alright. Thank you for that discussion. I believe Russ and I are done talking metrics for the day, so we'll head back over to you guys if you want to wrap up any more or not.

FRED BAKER: [I'm not sure that there's much to wrap up for anything.]

DUANE WESSELS: I would like to say thanks for the discussion. I think we made some good progress and I feel like we have some threshold values to sort of put into the document and we'll see if they stick, but we have a way to go on a way forward on these.

BRAD VERD: I think we definitely made some progress, and thank you for Russ and Duane putting that together. The dinner is tonight. I'm looking – Andrew?

OZAN SAHIN: Hi. Dinner tonight is at American Tap Room, and we can walk together if you all meet in the lobby by 6:45. It's just a five-minute walk from the hotel.

BRAD VERD: Okay. And then we're back here tomorrow same time, same bat channel. I will work on the wireless stuff and try to get that resent out to people. Anything else that I should address from food or building logistics, something like that? I don't know. Any issues?

RUSS MUNDY: Do we keep these?

BRAD VERD: No, you're going to have to turn your badges in when you leave. You'll get new ones tomorrow. Anything else? Fred? Alright. Thanks, guys.

[END OF TRANSCRIPTION]