OZAN SAHIN: Good morning, good afternoon, and good evening, this is the RSS Metrics Work Party teleconference held on the 17th of October 2019 at 1700 UTC. On the call today we have Duane Wessels, Russ Mundy, Keith Bluestein, Abdalmonem Galila, Chris Ishisoko, Jeff Osborn, Karl Reuss, Kazunori Fujiwara, Ken Renard, Kevin Wright, Paul Hoffman, Ray Bellis, Shinta Sato. From Staff we have Andrew Mcconachie and myself, Ozan Sahin. I just noticed that Jack Biesiadecki has joined us. I would like to remind you all to please state your names before speaking for transcription purposes. Thank you, over back to you, Duane.

DUANE WESSELS: Okay, thank you very much. Yeah, this is Duane, welcome everyone. It's great to see all the participants attending this meeting today. I think we've got a fair bit of material to go through so might be a little bit aggressive on the schedule. Essentially since the last work party meeting which was at the RSSAC workshop two weeks ago there have been a lot of things to update on the document and myself and others have been going through and doing that. So I will in this call today I will draw your attention to particular sections where change has been made and your feedback would be appreciated.

I noticed that sometime yesterday or this morning Daniel Migault who I don't see on the call, Daniel had gone through and added a bunch of comments and suggested texts. I think we'll save those to the end, even though they appear at the start of the document, so we'll skip over those. Alright, I'm sharing my screen, hopefully you can see my screen, the Google doc. I'm going to scroll down until I see something

interesting here. Alright, so this is one pending item that Russ and Steve are still working on. Then scrolling down there were some things that Daniel added that we're going to save until later.

Thank you, Paul, for also commenting on Daniel's changes, that's very helpful. One thing I did want to point out here. Ken, I think you're on the call here, you had made this comment about, this is in the section about vantage points and geographic location and what not, and we had discussion and there are some comments here in the document, I don't know if you had a chance to see those, but essentially the proposal is to remove the second sentence here and leave it at that. Would you be okay with that, Ken?

KEN RENARD: This is Ken, yeah, that's fine. That certainly belongs more in the recommendation section, but I was really trying to capture what Brad was saying, growing based on the geographic distribution or typological distribution are roughly the same ideas.

DUANE WESSELS: Yeah, my feeling on this is that this starts to head in a direction where we're being very descriptive and specific about how we want things to be distributed and I feel like that's maybe something for the future. At this point I think we need to focus a little more on the metrics and live with our admittedly more vague description of where our approach should be located for now.

KEN RENARD: That sounds fine.

DUANE WESSELS: Okay. So at the top of Section 4, there is a new Section 4.1 called Reporting, and this essentially outlines that reporting is going to happen on a monthly basis. The next paragraph talks about RSO metrics, the results are reported as pass or fail, and it gives a little bit of rationale for why that is. For those who have been participating for a long time would notice that this section does not make any references to concerns about gaming the system. It does talk about how these metrics are not designed to make performance comparisons between RSOs, so kind of the same thing, but just using a little bit different language. And then lastly it says for the RSS metrics that the actual VEZ will be recorded. Any comments about this new section?

Of course, throughout the document for each metric it also sort of repeats itself and says that they're all reported on a monthly basis in each metrics section. A little bit redundant, but that's okay. Alright, I'm not seeing any hands up so I'm going to move on.

Here in Section 4.3 which is about TCP, we cleaned this up a little bit. There used to be a sentence here that talked about sort of stopping the timer at the end of the data connection and we cleaned that up because the implementation may not actually have visibility into the TCP packets. But Fred's comment here and Paul's response match the intention that the timer can stop when response has been received, not necessarily when the connection gets closed.

Section 4.8 Unexpected Results, I feel like there is still some work to do here. Paul has this comment which has not been resolved. We need to write some more text about when a vantage point's measurements need to be excluded for various reasons. So that's still to come, I think. Paul, you hand is up?

PAUL HOFFMAN: Right, so, I don't know if you're waiting on me to do it, or if you're doing it, but just as a point, Ray Bellis brought up later in the document a case where the text that I'm proposing here would actually be valid, which is in your wild scenarios table. If a single vantage point only saw seven of the root server operators, Ray points out that's an indication that the vantage point is having routing problems and might need to be taken care of separately. So I do think we really do need to cover that here and to give whoever is running the collector leeway to exclude vantage points and such, as long as they're doing it in a transparent fashion.

DUANE WESSELS: Sure, I think we'll need to come up with a method or a post technique for figuring out when a vantage point is well connected that does not rely on it talking to root servers. We've talked about pinging other well known services or maybe have them do connectivity tests to each other to see if they're online. Okay, next, this is something that I expect we will have some discussion about. This Section 4.9 is new and this has come out of the meetings a couple weeks ago. And I really apologize, because if you had looked at this document sometime yesterday or this morning, this section may not have been there.

When I was preparing for the call today, I realized that this had accidentally been deleted somehow and I was able to recover it from the saved versions of the Google doc. But this is a key point from our discussions previously where we came up with a formula and a rationale for determining the number of RSOs required for reliable operation of the RSS.

There is a formula there which is K = 2/3 (n-1) and the explanation for that is that if a client of the DNS or a resolver gets a timeout from its first query, then its subsequent query should be have at least a 2/3 chance of being successful. And when we plug in the numbers for n=13, we get K=8. These numbers are used in some of the other metrics for determining availability thresholds and things like that. Some of you have seen this already because you participated previously in the workshop, but for some of you this may be new. I see Ray is making a comment, Ray, do you want to just speak to your comment now?

RAY BELLIS: It is unclear where the n-1 came from other than it happens with n=13, resulting in K being a whole number. So, it kind of looks like it was just made up as a way to come up with K being a whole number. Whereas I think if the intention is to get a whole number, it should be done with a more appropriate operator.

DUANE WESSELS: Yeah, so, it was my intention, and in the RSSAC meeting previously, I had, instead of using floor, I had used the ceiling operator to make it a

whole number, so it rounded up, but I couldn't, there is no ceiling operator symbol in Google docs, but that was the intention.

Regarding n-1, in the RSSAC discussion this was maybe a little bit confusing. I think when it proposed there was a mention of, I'm going to mis-remember the document, RSSAC023, whichever RSSAC document says that a loss of a single operator is not a big deal, and so that was one reason that we had n-1. But to me it makes more sense to think of it in terms of the fact that when you have a timeout, when you're a resolver and you have a timeout, you would exclude that, your first choice of resolver, you would exclude your first root server choice from your subsequent query, so instead your next query choosing from 13, you would choose from 12, or n-1.

SURESH KRISHNSWAMY: This is Suresh Krishnswamy here, I'm not on the Zoom but I'm on the phone line. I think one thing that was discussed in relation to the operator about rounding off to a whole number, I believe part of the discussion was that there are 13 root server operators, but in reality there are 12, because of the number of actual operators, if I'm recalling or remembering that part correctly. So it's actually a whole number when you do that computation. But I'm not sure if there was any other background associated with that.

DUANE WESSELS: Suresh, thanks, I don't remember the conversation going that way, I think, you remember what Paul Vixey was saying? Is that where you're remember this from?

SURESH KRISHNSWAMY: I'm not sure who made the comment, but I think it was like there are 13 instances but 12 of them are really effective, it's the effective count of operators, and so it's the number 12, two sets of 12, is how I remember that discussion, but I could be wrong.

DUANE WESSELS: Paul, go ahead.

PAUL HOFFMAN: This is Paul. Suresh, you do remember that wrong, that didn't come up, or else I would have jumped up, because K and A, the two that are run by VeriSign actually are considered different, they application in different places, they have different properties, so that discussion might have come up at some point, but it wasn't here. The n-1 is specifically for exactly what Duane said a few moments ago, which is any sane resolver software, if it gets a timeout and then is looking in the pool for what do I do next, it would have excluded that.

So, the n-1 is just due to the exclusion. Which means that let's say in the future we add another root server operator or we remove a root server operator, we will then get a fraction. So I agree with Ray that we actually do need to have a ceiling or a floor here, because it would make no sense to anybody reading this document if the requirement was that there be 8.5 root server operators or 7.8 root server operators.

RAY BELLIS: There are floor and ceiling characters in Google docs, I'm trying to find out how to do them just now. But what Paul was just saying, if we had another 14 root server operators, then K will actually become 9, rather than 8.

DUANE WESSELS: So, I included a graph, a chart in this section which I realize you weren't able to see, because I had to scroll down, but in this chart I had assumed that we would round up, so it has the ceiling operator. One reason I also wanted to include this chart is to make it clear to the reader that K=8 is not a fixed number for all time or for all situations, but it does depend on the number of operators. Russ, go ahead, I see your hand is up.

RUSS MUNDY: Thanks. I think Ray's point is good, and it seems like that has been maybe an unstated assumption that it would have to be an even number because the way we're dealing with the quantity of RSOs that are running. I don't know we want to even try to get in any kind of fractional percentage thing. So, it looks like the simplest choice would be to add a little bit more text that specifically said what you you've been saying, round up to the next whole number.

DUANE WESSELS: Okay, that sounds good, I can do that.

RAY BELLIS: I did [inaudible] the doc, Duane.

DUANE WESSELS: Alright, thanks Ray, good job. Then at the bottom of this section, also please pay attention to this paragraph because this tries to speak to the discussion that we had again about why is not just K=1, why is a response form one root server not sufficient to say that the system was available. So, this text is based on some words that I got from Tom, and I appreciate that very much, and it speaks again to the idea that these metrics are not designed to say what the end user experiences, but to provide useful metrics for the operators of the system. Alright, so unless there's any other big concerns with this, and again, please take your time to review this and make comments later, but I want to move on to some other things, because we have more to get through.

Alright, so this first highlighted sentence here in Section 5.1 which is about RSO availability, we added a comment that says that the availability of an RSO at any particular point in time depends not only on the RSO itself, but on the availability of the intermediate networks, as well. So, one reason for having this is as a rationale for why, for example why the thresholds will not necessarily be 100%, because the availability can be affected by some things outside of the operator's control.

Next, also in this section, this is something that we've talked about a couple times, and to be honest, we've sort of flip flopped, and maybe it's my fault, but the question is, in doing availability measurements,

how should the system treat a response that is something other than a successful response?

So remember, these are SOA queries and in my mind, for a legitimate response from a proper root server, these should always have R code = 0, you should never see refused or fail, or any of these other ones. Previously we had said that any response should be considered as contributing to availability. My opinion is that if you see anything other than R code = 0 then it didn't come from a real root server. So I guess I want to know if people are okay with this change, of you would rather it go back to the way it was before, where any response contributes to availability. Paul?

PAUL HOFFMAN: You're right that we keep going back and forth. I'm fine with either, and so I'm fine with this. Speaking for other people, the logic of well, if you got something back, it was available, I think was more based on the idea of well, what if it's truncated, like odd corner cases. But I think for a query for .soa, just looking at the R code and nothing else is sufficient to say this came from somebody who is likely to be the root server.

DUANE WESSELS: Okay, thank you. Ken?

KEN RENARD: I feel like this is conflating a little bit the correctness with the availability, but I'm fine with this, as well.

DUANE WESSELS: Okay. And I see Ray has made a comment that there are cases where you can get a serve fail from a proper root server if the zone has expired, had not been updated. That's a good point. Do you think, Ray, that should account towards availability?

RAY BELLIS: I'm still thinking about that one. In theory, is the zone is expired, then some time before that it's no longer correct in terms of the zone freshness. I'm just raising the point that a timeout is not necessarily appropriate in this case, because it doesn't actually mean the server is not responding, the server is responding but it's actually really not responding correctly anymore.

DUANE WESSELS: Okay. Alright, any other thoughts on this before we move on? Russ.

RUSS MUNDY: Yeah, it seems in that case what we were just talking about, if the zone data is in fact not available, the packets, yes, are moving back and forth and the machinery is running, but I think the fundamental point is that data that came back is not useable data and therefore saying that it is unavailable is a reasonable conclusion. So I certainly can build justification for considering a serve fail response to not be something that would be counted as an okay for availability.

DUANE WESSELS: Yeah, I think that's what makes this hard, you could argue it either way pretty easily.

RUSS MUNDY: Yes, yes. So if we go with R code 0 and stay where we are, the point today then let folks field the argument for it to go the other way, is that okay, if we leave it that way for now?

DUANE WESSELS: Sure, it's fine with me. Next in this section and in all the sections now, there is a threshold paragraph. This one is particularly interesting, I guess, because the threshold here was determined using the numbers that we talked about previously in 4.9. so, based on the discussions at the RSSAC workshop, the recommended threshold for this metric is 96% and the way that we arrived at that was by applying this formula for [inaudible] parallel availability and plugging in K=8 and n=13, and stating that the desired overall system availability was five 9s.

So, if you look for example at a table of availabilities using those values, you will see that the individual availability, a, set to 96% will give you the desired overall system availability of five 9s. So we talked again at the workshop how 96% is, the word we used was it has a "marketing problem," it might look bad to some people as not a very high bar, but this is what the math gives us, and I guess until we have some, I know Paul has some implementation of these metrics and Paul, I don't believe you've shared recently your values yet, but I would say we're in the right ballpark, right?

PAUL HOFFMAN: Yeah, I have not shared values because I actually don't have them. I've seen sort of head down and trying to make correctness work, and boy, is that fun. I strongly hope to have values before we meet again in two weeks. I'm not sure if I will, or not.

DUANE WESSELS: Okay.

PAUL HOFFMAN: But, from the previous values before we gotten this complicated, I think nearly everyone would pass 96. One wrinkle here, which I mentioned earlier, is that this is a threshold for each of the 4 values and I think we did see some cases where V6 was worse, but I don't think it necessarily got worse than 96.

DUANE WESSELS: Okay, that's a good point, thank you. Okay, so following this, there is a paragraph here that again reminds the reader that this threshold may need to be adjusted in the future if the number of operators changes, and again we have the formula, and thank you Ray for updated that. There is also a chart here which shows how these different parameters relate to each other. One way to use this graph is to say if our desired availability is five 9s, which is the very top of the graph, you can look at these different colored lines for different values of n and see where they intersect the top of that graph. So, for example, n=13 is the green

line, and it intersects on the X axis with an individual availability of something like 95.6%, but essentially we're rounding it up to 96%.

Alright, we're going to move on down. So, the next section is Response Latency, and again, there is a new subsection here about the threshold. Based on the discussions previously we had settled on 250 msec for UDP and 500 msec for TCP and the rationale is that it's based on the amount of time to circumnavigate the earth twice at the speed of light and there is a little formula there which lays that out, and it also says that the TCP threshold is twice the UDP due to connection setup latencies.

Next, this section 5.3 has changed a lot because what we had decided was previously you may remember there were two separate correctness metrics. There was one that using data set validation and one that we were calling exact matching correctness, and we had decided to combine those into one, so that has been done here. A lot of this text is different. We no longer use the phrase DNS correctness or matching correctness, it's just correctness.

There is description of the types of queries to send, essentially with a 90% probability the proposal is to send queries for which you expect a positive answer, or essentially a delegation to a TLD, and with the other 10% probability send a query for a name that is expected to be an ex domain, not in the zone. And then it talks about how to form those query names so that they provide good coverage of the name space and allow you to test the gaps in between the existing TLDs.

UNKNOWN SPEAKER: Duane, I just realized actually we have an error in that second bullet, where I say [inaudible] with dash test and then the examples do not [inaudible] with dash test.

DUANE WESSELS: Ah yes, we dropped that, didn't we. But you have test as the second component.

UNKNOWN SPEAKER: Right, so I propose just removing the [inaudible] with dash test since like you say, test in the second level would certainly pull these out.

DUANE WESSELS: Okay, yeah, we need to update that.

RAY BELLIS: Sorry, what's the rationale for removing the dash test? Do you specifically not want these queries to be identifiable by the RSOs? I'm just concerned that with the [inaudible] that we do get on the root servers, it would be kind of nice from my point of view to be able to say actually I know what that piece of running crap is.

UNKNOWN SPEAKER: It's certainly not to hide it. In fact the idea was of having test as the second level to make it very clear, that is you see www.test.exactly10letters, that it's a test. Do you think that it would be

easier if you saw it with dash test? That seems further than is necessary.

RAY BELLIS: Yeah, okay. I guess that's probably okay. I'm just wondering how likely we might be to see other people constructing queries like www.test, I guess not that likely. Probably not a big deal.

UNKNOWN SPEAKER: So, Ray, are you suggesting to remove the suffix?

RAY BELLIS: No, I was suggesting that actually having a specific identifying string in the suffix would be useful. But given that if the www.test is constant, I'm not that bothered. It's not as if these queries are going to be [inaudible], because they're specific queries for specific name.

DUANE WESSELS: One thing we talked about a little bit at the workshop, Ray, was whether these queries should stand out like they are now where they have tests, or if they should blend in more with regular traffic to make them harder to cheat or whatever.

RAY BELLIS: I don't see honestly how any RSO could feasibly manipulate the traffic based on the presence of that, but when people are doing subsequent analyses on root server traffic and they see a whole lot of random crap

come up, it would be nice if they had the ability to remove the stuff that is known to be from this test system.

DUANE WESSELS: Would it be better if instead of just the word test we had something like RSSAC metric test?

RAY BELLIS: Yeah, why not.

DUANE WESSELS: Make it a little less likely to collide? Is that alright with you, Paul? Make it a little more unique there?

PAUL HOFFMAN: Sure. How about, Ray, is RSSAC-test good enough?

RAY BELLIS: Yeah, that works for me.

PAUL HOFFMAN: Okay, so I won't have that in the current data I'm doing, but that's okay.

DUANE WESSELS: Alright, so we can work on that later. So again, this whole section is new and there is a set of rules which I see Paul has been making edits

to, thank you Paul, that describe how to process and do the correctness matching. Currently one thing Paul and I have talked about in working on these is that the matching is done based on the actual response and not necessarily the expected response, because we didn't want to be too strict with how up to date a root zone TLD list needs to be when you're generating the queries, and so on, so if you generated a query assuming that TLD was in the zone but then it got deleted, and then your response may not necessarily match your expectation. And I guess, Paul, I think you're updating this based on your implementation, right? So your implementation experience is sort of driving these rules.

PAUL HOFFMAN: Yes, but just to be clear, the changes that you're seeing there are to simplify so that somebody who is reading this from top to bottom has fewer indented bulleted lists. All the changes there are identical to what has been there for the last week, it's just removing a level of bullets so that someone reading from top to bottom who said does it have this rule, is more likely to see it.

DUANE WESSELS: Okay, thank you.

PAUL HOFFMAN: Having said that, with lots of people on the call who are deeply involved with looking at DNSSEC and such, I would love to know if these rules are correct. That is, they look correct to me, and that doesn't mean much, so if people could look at this and look at some responses to queries

and such like that, even for very edge case ones, we want to have these matching rules be exactly what one would expect in an implementation.

DUANE WESSELS: Alright, great. So the threshold for this metric, as for all the correctness metrics here is 100%. The expectation is that root servers also serve correct responses. So that is something that everyone has agreed on to date, so that's not really controversial, I don't think.

I'm going to skip down, the next section is Publication Latency. The method here has not changed. I see there are some comments about notify, that's good, we can work on that later. The threshold part is new. Based on the discussions from two weeks ago, we settled on a recommended threshold of one hour, and remember, this is the median value of all the update times from all the probes. Ray, given that it's the median, does that change your concern at all, or not?

RAY BELLIS: Yes, it does make it less of a concern, a little bit at least in any case. I've been reviewing our own low time RSSAC statistics and I think from the 95th percentile, and I have seen a few outliers which I can't explain, I need to look into it to find out why they were so low, I'm looking at some that has taken a few hours on occasion, and I'm not sure why that is. We have actually, full disclosure on this one, recently taken the decision to use max refresh time setting to force our secondaries to proactively poll the zone much more frequently than is specified in the SLA. Actually there is a late comment which you may have not seen the

detail of yet, which I think was on that same comment train, yes, that one at the bottom there, that everyone can now see.

DUANE WESSELS: Right, you've seen the rationale here that the threshold is twice the SOA refresh parameter which is 30 minutes, and that's a median.

RAY BELLIS: But my experienced has been, I'm trying to think, 95th percentile, would that be better or worse?

DUANE WESSELS: That would be worse.

RAY BELLIS: The 95th percentile is a much higher, yeah, so 50th percentile should be okay.

DUANE WESSELS: I think if we were considering 95th percentile in this document instead of median, I think one hour would be too low. I think it needs to be higher than that, as well.

RAY BELLIS: Our median, I'm not too worried.

UNKNOWN SPEAKER: Just to drop the notify, I mean this notify is expected to be either between the distribution master from the root zone maintainer, or the one from your distribution master and your different root servers. So, it's unlikely that an attacker could drop that packet.

RAY BELLIS: No, simply because it's UDP, they just get dropped.

UNKNOWN SPEAKER: Yeah, okay. But it's limited to this UDP thing.

RAY BELLIS: Yeah, notify does not use a reliable transport.

PAUL HOFFMAN: This is Paul, Duane, can you put on your RZM maintainer hat for a moment?

DUANE WESSELS: Sure, yeah.

PAUL HOFFMAN: Why are those notifies sent over UDP instead of TCP?

DUANE WESSELS: Well, that's the way the protocol works. But I can tell you that it's not a single notify message, I believe we have almost 40 servers and I think that all of them send out notifies. So, you get kind of a blast of 40 notifies each time the zone is changed.

UNKNOWN SPEAKER: Ah, thank you. To me, that completely ameliorates any concern I have. My concern which sort of went along with Ray's was well, what if you miss a notify, are you screwed until the next SOA? And the answer is if you miss the notify, you're doing something really wrong.

DUANE WESSELS: Well, if you miss the notify, you query at the refresh time, which is 30 minutes. You query every 30 minutes whether you get a notify or not.

UNKNOWN SPEAKER: Yep, I agree.

DUANE WESSELS: Even if there are no notifies, you're still polling every 30 minutes.

UNKNOWN SPEAKER: Okay, yeah, to me that's an hour, even if it was at the 95th percentile, would probably be okay.

DUANE WESSELS: So, we can add some text here that explains this better about notifies and polling and stuff like that, I think that would be good.

RAY BELLIS: Paul, that's where my second comment, the last comment on that page, came in. We do have to be very careful about the notification and propagation. Every primary server has to have a list of explicit, also notify secondaries, if you're using your own internal distribution master, as we do. So, if for whatever reason you fail to add a server to that list, then it will fall back to refresh time instead of notifies. And in fact, I'm considering the appropriate of not bothering with explicit notify lists or our new servers going forward, and just relying on faking the refresh timer, so that every secondary just polls every 2 minutes and say have you got a new SOA for me. Because that removes a lot of mountains overhead in trying to create and maintain those notify lists.

DUANE WESSELS: Alright, I'm going to push forward because we've only got 15 minutes left here. The next section, we get into the RSS metrics and this first one is very different, because again, we're using these equations and formulas from the discussion about the number of operators needed for reliable service. So, we had a lot of discussion about this two weeks ago at the RSSAC workshop and where we settled was that, it's a little bit complicated, but the central collector system will analyze all of the individual availability measurements and for each time interval and for each vantage point, it will tally up the number of RSOs that responded

to the availability query from the earlier section 5.1, and then it calculates an aggregated RSS availability based on this formula.

So, it sums up these R sub TV values which was capped at K, so for example in one interval if a vantage point receives responses from all 13, for that interval it's actually capped at 8, so 8 is a maximum that you can get at any interval, and is sums up all those individual measurements and divides that by the sum of the total possible measurements, so that basically the maximum which is K, and that's the aggregated availability for the month. This table down here has some example scenarios, a little bit farfetched, but they demonstrate how this works.

So, first up, if you have a hypothetical attack that lasts a whole month and entirely takes out a single RSO only, that doesn't affect the availability because that's just taking out one, and you only just need 8, so the measured availability is 100%. If you also have a month long attack that takes out 5 RSOs entirely, you can still have 100% availability because again you only need 8. If this month long attack happens to take out 6 RSOs entirely, then the availability drops down to 7/8, because for every interval the measured value was only 7, instead of 8. If there is a 24-hour attack that takes out all the root servers entirely, the measured availability is 96.66%, which is 29/30.

The largest availability that you can have that is less than 100% is this next one, which is in one 5-minute interval, 1 vantage point can only reach 7 RSOs, and it's 6/9 availability. The point at which you get less than six 9s, where you start to get five 9s is something like 7 vantage points can reach no RSOs. So where you subtract 14 from the

numerator there. Does that make sense to everyone? I know that we had some discussions and not everyone agreed with this method, but this is where we settled at the end of the workshop.

RAY BELLIS: Ray here. I think the method is fine, the false positives is of concern. Paul mentioned this comment earlier in the meeting.

DUANE WESSELS: You're talking about when your vantage points are disconnected and providing bad data?

RAY BELLIS: Or partially disconnected at least, yes. If a vantage point for example has two upstream transits and one of those transits has an issue, it could potentially result in some of the RSOs looking as if they're unavailable when they're actually not, at least with respect to that point.

DUANE WESSELS: Yeah, so I guess we can get into this a little bit. Do you think, Ray, that it would be sufficient if the vantage points tested connectivity to each other? And if they met some threshold then they would be considered functional and working?

RAY BELLIS: I think that would probably be as good as anything. I think it's probably more reliable then for example trying to ping Google or any other global scale system. And least they're end points that would be under the control of whoever is actually running the system.

DUANE WESSELS: Alright, so I guess we'll have to come back with a more concrete proposal on that in the future. Paul, your hand is up?

PAUL HOFFMAN: Yeah, I'm fine with them all trying to check each other, but I think at the end of the day, well, before the end of the day, the person running the collector should have the ability to determine that a vantage point is not doing what we expect for any reason and pull that out of the measurements, as long as that's done in an open and transparent fashion. And in this particular case I could totally believe that the vantage points could all talk to each other and still have the possibility that this one vantage point has a disconnect from a bunch, but not all the root servers.

RAY BELLIS: It's theoretically possible, but I don't think particularly likely.

PAUL HOFFMAN: Well, I don't think any of this is particularly likely, but in the case that a non likely thing happens, that can be explained and everyone goes, like, oh, oh, I get that now. I really do believe that whoever is running the

controller and therefore is generating the metrics, should have the ability to say we could see that something was broken at this vantage point, not for this whole root server operator, but for this vantage point, or this set of vantage points, and take them out of the calculation. So that if we had 20, that there might be only 19 for a day because of something that is defined and reported.

DUANE WESSELS: Okay, thanks. Russ?

RUSS MUNDY: I was going to say something along those lines, but I was trying to describe it somewhat differently. And that is, it seems to me since we do expect for the official system, or whatever becomes the official system, to be under the operational control of one entity, even the probes would be spread about and so forth, and perhaps a way to handle this problem is include somewhere in the document a requirement for the operator of the system to remove probes, somehow make sure that the results the probes are seeing actually reflect the state of the RSO operations or the RSS operation and not problems with either the probes or the monitoring system itself.

So, without trying to solve the problem of how they would go about doing it, it might be simple as if we state that this is a requirement for the monitoring system implementation and operation, they can identify themselves, problems with their monitoring system that are not reflective of the RSS or RSO individuals.

DUANE WESSELS: Alright, thank you Russ. Daniel?

DANIEL MIGAULT: Yeah, so I think I'm not going to repeat what Russ or Paul is saying, but I think we have to assume that the vantage points are well connected, and it's up the vantage point operators to define how to meet that requirement. I don't think we should dig into that, at least in that document.

DUANE WESSELS: Okay, thank you. I'm going to scroll down a little bit. So, in this, for the RSS availability, the document here says that the recommended threshold is five 9s and this is actually I think a little bit different than the way left things from the RSSAC workshop. I remember at the RSSAC workshop we said 99% and as I was writing this, I was struggling to reconcile that 99% here with the desired availability of five 9s in section 5.1, where we came up with the individual RSS availability. So, I definitely would like to hear people's thoughts on this, especially those that were at the workshop and maybe remember the same as me or different than me. Alright, no comments, so I guess we'll proceed with this for now, unless someone comes up with a reason why it should not be that way.

So, RSS response latency. Here is a little bit new, too. The proposal from the meeting was in each measurement interval you find the best K response latencies for each vantage point, and then aggregate those.

So, rather than aggregating all the measurements, in which case the RSS latency is just a median of all of them, you take the median of the best one, the best K. That has a nice property, it sort of mimics a little bit the way that we expect some recursive name service to work, in that they choose lower latency servers for their queries and then the thresholds can be lower than the individual RSS thresholds. So, the recommended thresholds for this case are 150 msec for UDP and 300 msec for TCP, and the rationales that these are based on, the individual latencies, but multiplied by a factor of K/n, approximately. Any comments about RSS response latency?

Alright, RSS correctness is straightforward, it's just a simple aggregation of all the individual correctness measurements and again, the threshold is 100%, I believe not too controversial.

RSS publication latency is something we added back in. Very early in the work party we decided this didn't need to be here, but at the workshop we felt that its absence was sort of strange, and so it's back in here and the recommended threshold in this case is 30 minutes, which is 1 SOA retry value. Again, for all the servers being measured, the median publication latency threshold would be 30 minutes.

Alright, so we made it to the end of the metrics. The recommendations here have been updated a little bit. For example you can see in this one, there is some text that has supposed to have been remove. This is because we have agreed now that the raw data will be available to anyone in the interest of transparency, that's the third bullet, so that third parties will be able to identify the computation of the metrics and operators also can get a heads up on advance notice if they want to on

how their metrics might turn out for the month. Because of that, proposing to remove this sentence in the 4th bullet that talks gaming system and what not.

Section 7 has some new tables summarizing all the threshold values and you can see those here. I believe they're all in sync, they're all correct. Section 8 has the example results which we've agreed to keep in the document. And that's pretty much it. There was some stuff, I think Paul, you're right, I think we need to clean that up, that doesn't need to be there anymore. I think we're done. So, Daniel, I don't know if you were at the start of the call, but did acknowledge that you had made some comments at the start, but since they came in late, I wasn't planning on talking to those today. So, we'll have to talk about those in the document or on the mailing list, if this I know with you.

PAUL HOFFMAN: Yeah, that's perfect.

DUANE WESSELS: So, our goal for this document was to have a final version to look at in two weeks, also note that in two weeks the meeting will e moved from Thursday to Tuesday. So the proposed date for the next meeting is October 29th. I think everyone has received a meeting invite already for that. If not, please let myself or one of the staff members now, and we can get you that meeting invite. Also, you can expect to see this work presented at the upcoming ICANN meeting, there will be an RSSAC caucus meeting there and also at the ITF meeting there is another RSSAC caucus meeting. Thanks for coming everyone, if you have

comments about anything you've seen today, please add them to the document, post them on the list or send them to me privately and we'll get them addressed.

[END OF TRANSCRIPTION]