HYDERABAD – How It Works: Root Server Operations
Thursday, November 03, 2016 – 15:15 to 16:45 IST
ICANN57 | Hyderabad, India

UNIDENTIFIED MALE: All right. If we could take our seats, we're going to get started on the next session. Troublemaker won't sit down.

UNIDENTIFIED MALE: [inaudible]

UNIDENTIFIED MALE: All right. You made it this far. I appreciate you all coming back. I see some new faces here as well, so welcome. This is "How It Works session: The Root Server System," presented by RSSAC (Root Server Security Advisory Committee). Jetlag is still kicking in, so I'm probably – oh, no, it's written right there. I said it right. Huh. Okay.

We have Wes Hardaker and Lars Liman here, both operators of various letters. It's kind of like James Bond, where people are Q. We have B. We have I. So they can be super-secret people.

I'm going to let them run with it, so I'll turn it over to Liman and to Wes.

LARS-JOHAN LIMAN: Hello. Welcome to you all. My name is Lars-Johan Liman. I work for a small company in Sweden, Northern Europe, far away from here. When I left home, we had our first snow, so I'm kind of happy to be here. We operate one of the root name servers, the letter I.

I will very soon hand over to Wes Hardaker, but I will go to ask for the next slide and just do a quick run of what we're going to talk about here today.

We're going to look at the root server system and try to do an overview of it. That ties into the domain name system from the technical standpoint: how does it all work? What's the underlying technology that carries the domain names that you deal with here?

That leads into a brief history. People often wonder: how did we end up here where we are? There is a history, and it's fairly long.

We're also going to cover the root server system today – how it works and what the features are – and we're going to explain the term "Anycast" that many of you probably have heard, but it may not be all clear to you what it actually entails. Then we're going to talk about the Root Server System Advisory Committee,

which is one of the Advisory Committees within ICANN, its publication and its current activities.

Wes is going to start with the overview of the domain name system, so please go, Wes.

WES HARDAKER: All right. Thank you. I'm Wes Hardaker. I'm with the University of Southern California, and as Lars said, I run the B-root – or I'm one of the people that help maintain it.

We're going to go briefly into the overview of the DNS and what it actually is. How many people were in the Overview of the DNS this morning? Some of you? Okay. How many of you have never been to this presentation before? Quite a few. Good. All right. By the end of the day, you'll learn a lot more.

One more. A couple of quick things. The identifiers on the Internet were created because every host needs to talk to another host, and that's all done via numbers. It's all done via IPv4 and IPv6 numbers, Internet addresses.

The reality is that humans don't do numbers that well. It's hard enough for us to memorize telephone numbers. Imagine trying to memorize telephone numbers. Imagine trying to memorize all of the IP addresses for every host you want to talk to on the Internet. It's nearly impossible.

These addresses are uniquely allocated by the IANA RAR system to everybody that needs and Internet address. But when it comes to humans, we have a hard time looking that up, so we even invented a naming system that goes along with it.

So the original problem is you have an IP address and it's hard to remember. What name can we assign? We'll go into the history later, but what was creatively came up with is that it creates a Distributed Naming System that exists from the root and comes on down. We'll talk about what DNS and we'll talk about what the root is and where a resolver starts their lookup process, which is really what the roots are.

Next. Fundamentally, the DNS can be thought of as a tree structure – an upside down tree. It actually looks more like the roots of a tree, to be honest. It's a lookup mechanism. When you need to look up some object that has a name associated with it and you need to translate that into something else – normally that's IPv4 and IPv6 addresses, but many thing are stored in the DNS, including mail servers and other stuff; we'll come back to that in a minute – you would always start where you don't know. Where you don't know, consider the root. So it's before .com, it's before .edu, it's before .mail, it's before .uk, when you don't even know where they are.

We'll talk about this more in a later slide, but fundamentally you always start at the beginning and you work your way down to the answer that you might need to know. For example, on this left-hand side of this diagram, you would start at the root and ask the root, "Where is edu?" Then you would ask edu, "Where is cmu?" which is Carnegie Mellon University. Then you would ask Carnegie Mellon University, "Where is www?" It's a little bit more complex than that, and we'll get into the details later. But fundamentally, you're always chaining down this tree.

When you, sitting at your computer, want to type in an address – say, www.example.com – into your web browser, you're not actually asking all of these machines yourself. You're asking a recursive resolver, most of the time, that has cache information – we'll talk about that in a minute – that is responsible for doing all of that for you.

So you only send it one thing. You say, "I need to go to www.example.com, please," and it's going to go off and it's going to go talk to the root server and say, "Can you tell me where www.example.com is?" The root server is going to say, "No, I can't, but I can tell you where com is, so you can go talk to them next." Excuse me – this is example.org. I keep saying com.

So it's going to chain off and it's going to go to the example.org servers and say, "Hey. Can you tell me where www.example.org

is?" The org servers are going to go, "No. I don't know anything about example.org and what's inside of it. I can only tell you how to get to the name servers that can answer that for you. So here's the name servers for example.org"

If you want to come to The Beginner's Guide to DNSSEC, which is occurring I think Monday. I forget which day it is. I think you're right – I think it's tomorrow. There'll actually be a little skit that demonstrates all of this for you, with people handing pieces of paper around.

The main point of this slide is that all of that information is actually cached. If for some reason somebody wanted to go very quickly after that and ask the resolver, "Where is www.ICANN.org?" that caching resolver is not going to talk to the root again. Not all the questions start at the root because that caching resolver is going to know, "Oh, wait a minute. I already know where .org is, so I'm going to go ask. org directly where ICANN.org is. I don't need to go all the way and start at the top again. That's very important and it's one of the reasons that the DNS has scaled so well over the years.

So the root servers only need to know where you go next. It's just a starting point. If you go ask them for stuff, they're going to hand you a list of where to talk to next – .com, .net, .org. There's a list of servers for each of those and for all of the other TLDs –

the ccTLDs, the gTLDs. The root has a list of where to go talk to next.

Again, caching means that you don't need to go back. Those records in the root zone are actually cached for a very long time – at least a day.

There's been some recent refinements to the DNS over the years that do affect the roots as well. First off, there was security extensions called DNSSEC that was introduced – we'll go through the history a little bit later and you'll see the exact dates – in the last decade and that's been ramping out. They're really just cryptographic signatures on the DNS. That allows you to prove that the person you're getting the answer from is not lying to you. If you want to know more about that, again, go to the session tomorrow night. And it reduces the risk of spoofing, so nobody can tell you an invalid answer.

There's also been some recent work on privacy enhancements. If you don't necessarily want somebody to know exactly where you're going, you don't actually give that query out to everybody. Some people have that as a concern, and it's being worked on right now in the IETF. Hopefully we'll see some deployment of those solutions in the near future.

Finally, Anycast, which I'm not going to talk much about now because Lars is going to talk about it later, is really where

multiple servers are sharing a single IP address and multiple servers can respond to you and respond locally as well. That improves latency and resilience, and it also protects about DDoS attacks. We'll go into that later.

There's two things that are frequently flung around, terminology-wise, and they're actually distinctly different. One's the root zone, and one's the root zone servers. The root zone is the data, is the starting point. It's the list of all of the TLDs and all of the name servers associated with them that I talked about earlier – .com, .org, .net. It's managed by ICANN. So the contents of it are managed by ICANN per community policy. That's one of the things that ICANN as the larger organization does. It's compiled and distributed by the root zone maintainer to all the root zone operators. So the root zone operators each get a current version of that and redistribute to you, the rest of the world.

The root servers then – Lars, go ahead.

LARS-JOHAN LIMAN: I just would add a small comment here regarding the root zone. The root zone is a file. It is absolutely and totally public. You can download this from a server. Using your web browser, you can look at the exact file that is being served from the root servers. It's being updated twice a day, if I remember correctly, and also

uploaded to the servers. So you can look at it in your browser and see exactly that data that we serve. There's nothing secret there.

WES HARDAKER: Yeah, very good point. Very good point. Not only that, but DNSSEC allows you to check that nobody is lying to you about that they're distributing different data from that file. Thank you, Lars.

So the root servers are the ones that respond when you go ask them the starting question of "Where is .com?" for example, and they're currently distributed from 13 different identities, the letters –A through M.rootservers.net. They're the name server records for the top level where you start. So there's 13 different name identities.

The root servers are a purely technical role. They just serve the root zone. They don't do anything else. They're the responsibility of the root one operators. They don't modify the data and they don't do other things. They're only serving the data that was given to them, and you can check them using DNSSEC as well.

Who are the root server operators? They are 12 different professional engineering groups focused on making sure that

reliability and stability of the service is maintained, that it's accessible to all the Internet users, that everybody within the Internet has equal access. Technical cooperation – they have to work together. And they're all very professional organizations. They [send] out as the best quality service that they can.

They're very diverse. This is actually a strength of the types of root server operators that they are. They are all technically different. They are organizationally different, and they're geographically different, which brings a strength in that there's not a single point of failure within the system because everybody is slightly different from each other.

Next, please. An important thing that the root zone operators are not involved in is policy-making. That's what happens at ICANN. They're only technical. And they don't do data modification. They are just the publishers. The root zone operators don't actually change the data. That's, as I said before, ICANN's job.

The operators are involved in the care for operations and the evolution of the service. As the demand expands, they have to expand. They evaluate and deploy suggested technical modifications. As the DNS changes over time, they have to make those modifications within the servers that run the root system, and they make every effort to make sure that everything is

stable, robust, and reliable. There's an excellent track of that that will be documented in the history as well.

Next. With that, we'll look back. So that's technically how it works. Now we're going to look at the history of how it got to where we are today with 13 organizations. It's pretty amazing to me that that system has scaled from a very, very small Internet many, many years ago – many decades ago – to one that now expands everything that we have. That comes down to the history and where things came from, and Lars is going to go into that next.

LARS-JOHAM LIMAN: Thanks, Wes. Next slide, please.

UNIDENTIFIED MALE: We're having some latency, so when you say "next," give us for seconds or so.

LARS-JOHAN LIMAN: Sure. Absolutely. The DNS system was invented in the early 1980s. The problem at hand was that the namespace was flat. There were no dots in the names. Every computer had a unique name. The number of computers increased, which led to a couple of problems.

First, you couldn't have the name that you wanted on your computer because that was already taken by someone else. Two, you had to have a central repository of names. You had to have someone who kept track of all the names, somebody to make sure that there were no collisions. The SRI International Research Corporation in Menlo Park, just next to Stanford University, that actually was a spin-off from Stanford University long ago, had a contract. They were assigned to keep track of this.

The bigger the database got, the more changes always happened. If you have three computers, the chance that one of them will change its name today is very small. But if you have 1,000 or 2,000 computers, there's always something that needs to be changed.

So SRI was faced with having to do more and more updates. You also had to distribute this list of hosts across the entire network. When they started to discover that only sharing this gigantic file of all the hosts on the Internet actually took up a noticeable capacity of the Internet itself, they started to realize that this was probably not a good path forward.

People sat down and started to think about a database approach, and that ended up being the DNS. Two of the most well-known names driving here were Jon Postel and Paul

Mockapetris. They got together. They wrote documents within the IETF – RFC documents – that described ideas for how to create this database, and they also produced software.

Paul Mockapetris wrote a piece of software called Jeeves, which was the first implementation of a DNS server. With this [inaudible] came also the concept of a multi-level system – but that was described before – where the different levels in the tree where separated by dots.

Now, when you created this hierarchy and the database approach where you had to be referred to the different levels in the tree, they also needed to have servers at the top. So they created the first set of root servers back in 1983. The very first one was stood up at where Wes works today, and it was followed by a few others.

This was quite sufficient for the Internet of the day. Now, you have to go back to 1983. You could count the hosts on the Internet in the thousands. It was a totally different environment. Computers were the size of two of these tables and this high. So it was a totally different thing.

Well, things developed. People experimented with this, and it got more developed and it got more spread. Next slide, please.

They added servers. This is still Jon Postel, mostly Paul Mockapetris, rolling out their idea. It's getting traction. It's getting good feedback from the Internet Engineering Task Force, the standardizations body.

Servers were added at NYSERNet, University of Maryland, and also in other military installations because the fist interpretations of the Internet had its roots in military money at least. I would say that much of it was developed at universities but by using military money, so this was a cooperation between the military and the academic side. That's why y0u a spread here between academic sides and military sides.

This continued to develop. Next slide, please. In 1991, the first root server that was operated by an organization outside the United States was established, and that was actually our server in Stockholm. Those of you do their math quickly realize that we celebrated our 25th anniversary this summer. Now, that's not a scary number. The scary number is that I've been around for 14-and-a-half of them. So I started to work at the Network Operations Center for NORDUnet in 1992. I no longer work there, but that's where it started.

So that was the first step outside. This has its roots in the fact that NORDUnet, the Nordic University Network, which is a pan-Nordic network covering the national academic networks in

Sweden, Norway, Denmark, Finland, and Iceland was the biggest contiguous patch of Internet connectivity in Europe at that time. So it made sense to put a server in the biggest patch in Europe at that time.

But things continued to evolve. Next slide, please. In 1993, we ran into a size limitation in the DNS. The DNS standard limits of the size of original DNS packets. This has been modified and upgraded since, but back in 1993, the allowed size of a DNS packet on the network was 512 characters, which is fairly small.

Now, when one of these caching resolvers – the helper server that looks for information in the DNS tree – starts up, it needs to know where the root name servers are. It actually has a list of them, but it doesn't really know that that list is fresh and up-to-date.

So one of the very first thing it does is to ask a root server from the list, "Can I please have a fresh copy?" The root servers always know the freshest and most up-to-date version of the list of root servers, of course. This is referred to as the priming query.

We really, really wanted this to fit into one packet – the list of root name servers should go into one packet. We also wanted to deploy more root servers. By using a clever idea by Bill, Mark, and Paul here, by just changing the names of the servers, we

could take advantage of how these DNS packets are constructed. When the computer says, "I want to put this into a DNS packet," there's rules about how to stuff this stuff into the packet. This was a clever idea – to take advantage of that algorithm [for] how the packet is constructed. That saved some space in the packets. We could actually fit in four more servers.

So the servers renamed in 1995, and they all ended up in the rootservers.net. Next slide, please. They were all renamed according to this schedule. Here you can see at the bottom that nic.nordu.net, the former name of our server, became i.rootserver.net. [inaudible] ISI.edu become b.rootserver.net, etc.

Next slide, please. So by using this trick and renaming the servers, there was room for four more servers before we would, again, hit the 512 bytes limit. Jon Postel, who was in charge of handling this, set up a set a criteria for how to select new root server operators, and they were quite simple and straightforward.

The first one was need. Where is the Internet population not properly served? Again, go back to 1995. This is 20 years ago. The Internet was something completely different than what it is today. So where was there need for better service? Where would a good place be if you looked at connectivity? It would have to

be easy to get to this root server to get a response from it. You could have an underserved area in the absolute northern parts of Norway, but it doesn't make sense to put a root server there with this old technology because nobody else would be able to go there from Africa or from South America. So he wanted places where a lot of people could reach it easily, so it had to have good connectivity.

The organization operating would have to commit to send and respond to traffic without filtering and modifications, so he was looking for people who had the right kind of mental approach to doing this service. Also, he wanted a community consensus around that operator, so he would like to see that a lot of people got together and said, "Yes, please have these guys do it for us because they are good guys."

So these are the criteria he used, and then he asked around and asked for proposals.

Next slide, please. In Europe, the RIPE NCC (RIPE Network Coordination Centre), which is the entity that hands out IP addresses to Internet service providers for very large definition of the European area, does a lot of good networking. They're well-situated, and they had good connections with the London Internet Exchange, which is one of the major ones in Europe. So

they placed their server in London, but it was operated from Amsterdam.

In Asia, the WIDE project was chosen to run M-root on similar grounds. J-root has stayed at Network Solutions, and there were plans to find a better home for it. L-root was transformed to ICANN as part of the founding of ICANN. When ICANN was founded in the late 1990s, it made sense that ICANN should operate one of the letters.

Next slide, please. All this was planned and handled by Jon Postel. Jon was a very considerate person and an extremely good diplomat. I really respected him. He made very good things happen. The only problem was that he died very suddenly and at a very bad time, actually, because ICANN was just about to be formed. Jon was very much part of the formation of ICANN. I wouldn't say it was only his idea, but he realized that "So much hinges on my single person on this big and expanding Internet. This cannot be right." So he was very much helping to found ICANN and was hoping to give his authority to ICANN.

When he suddenly died after heart surgery, the root server operators realized, "Hmm. We don't have this central figure anymore. There's no central function that keeps us together and leads us." So the root server operators, for the first time, actually, all met together during an IETF [meeting] in 1998.

We got together and we set down and said, “So now what do we do? We’ve lost our leader, so to speak, so let’s sit together and talk and decide how we should continue to operate this.”

Out of that came five internal statements. We all agreed that we wanted to operate for the common good of the Internet, so this is a service which we don’t do for money, really. We all agreed that the IANA should be the source of the root data, so we all take the same root zone file that I described before. We don’t take it from different sources. We don’t modify it. We all use exactly the data that the IANA gives us.

We all agreed to sufficiently invest in the operations and to operate responsibly to make sure that we were in line with the demands from the Internet for the service. Should the case be that we no longer wanted or could live up to that, we would give the other operators and the Internet at large very proper notice and say, “In a very long time, we won’t be able to sustain the line operation that we do. So we will have to see.” So the Internet would have sufficient time to figure out how to handle that.

We also agreed to recognize the other operators. We have this mutual understanding that, yes, Wes is responsible for operating B-root, and I am responsible for operating I-Root. Also, Brad is part of A- and J-Root and so on. Tripti is for D-Root. So we recognize that these organizations actually do this work.

Next slide, please. Today we have a system that evolved from that meeting in 1998. Wes is going to talk more about that.

WES HARDAKER: I think first we actually have one question that came in online.

LARS-JOHAN LIMAN: Ah.

UNIDENTIFIED MALE: Yeah we have actually some multiple typing right now, so maybe we can split some questions, if there are more, between the floor and the remote. The one that I have right now is from – and I apologize if I pronounce this wrong – [Mohibalah Mankil]. His question is, "As there are 13 root servers, how is it categorized for which request should go to which root server?"

LARS-JOHAN LIMAN: Technology is your part.

WES HARDAKER: That's a good thing that actually we are missing in our slides to some extent: we don't talk much about how the resolvers work. Fundamentally, when a resolver gets a list of name servers, I will start by saying it picks it randomly from them and says, "You

know what? I'm going to pick #13" – is that my phone? Great. "I'm going to pick #13 and I'm going to ask it. If I don't get a response from it, I'm going to go ask somebody else." So it picks randomly. Later on, it tends to prioritize a little bit with the fastest response.

Liman, go to the next question.

UNIDENTIFIED MALE: Any questions from the floor at this point? There's a couple. There's one back where we need to get a mic back there. Then we'll take another one remote

[inaudible] It was here somewhere.

UNIDENTIFIED MALE: Please state your name.

UNIDENTIFIED FEMALE: Hi, I'm [Rishmi]. I've just one query. As per my understanding, the number of root servers is limited to 13 because of IPv4 addresses and limitations of the UDP protocol. Since our world has been migrating to IPv6, my understanding is that 14 root servers is possible. If that is the case, then where would it be launched? Or are there any plans in the near future?

LARS-JOHAN LIMAN: Shall I? Okay. Thank you. First, the limitation is not in IPv4 or UDP. The limitation is in the size of the packet that contains the list of the servers. Whether that goes over IPv4 or v6 is not relevant.

That said, the DNS protocol has evolved – I'll [look] forward so I see more people – since 1993/'95 when this happened, and today we have something called extended DNS, which can actually make room for longer such lists.

We have a big problem. We are now seeing the possibility of creating more letters, but we don't have a process for doing so. So the first thing we need to do is create a process for adding more letters. When we do so, one of the first things should be the criteria for creating more letters. What's the reason for doing so?

WES HARDAKER: Just as importantly, there's actually a large belief that 13 is too many or may be too many. So there hasn't been a huge amount of technical amount of evaluation in terms of what's the right number, but the reality is that Anycast, which we'll get into shortly, is actually a better answer, that it's important to have more addresses near you and that the number of letters is actually less important than the ability to reach a good number of addresses.

We'll come back on that for a little bit more. If at the end of the entire session you still have a question related to it that doesn't seem right from a technical point of view – remember that this is a very technical presentation. So there's no understanding right now that there's a need technically for more letters. Politically or communally, that might be different.

LARS-JOHAN LIMAN: Also, again, there is no limitation to 13 servers. It's a limitation to 13 letters. By using Anycast, which I will describe shortly, we can have many, many, many more servers. Already today I will jump into my [part]. We already have something around 600 servers operating today. So it's not a limitation in number of servers.

WES HARDAKER: Yeah. We'll see that again in a minute. Did you want to go to the next question?

UNIDENTIFIED MALE: Well, oddly enough, through some sort of predictive answering, you've answered exactly the question that was also asked online. So at this moment, we are good, unless we have some more questions from the floor.

WES HARDAKER: Again, we're going to answer more questions going forward, so unless you have –

LARS-JOHAN LIMAN: Yes. There will be more chances.

WES HARDAKER: Okay. So we're going to talk about the root server system today. You just heard the history of it and you heard how DNS works. Now we're going to go onto what it looks like today in the next few slides. Go ahead. We'll look at what happens today.

Today we have A through M in terms of letter. You can see all of the root zone operators, on the right-hand side, organizationally. You can see that, on the IP addresses, we have a ton of IPv4 and IPv6. In fact, every letter has both a v4 and a v6 address today, so no matter which protocol you are using to send the packets, you're going to get 13 different random things to go ask the query to.

Next. This is actually map. If you go to rootservers.org, you will see this map. You can go look at it. It's a map of where all of the root zone operators have instances today.

You'll note that there is 13 letters, but there is over 600 – it's actually kind of cut off down there – instances around the world.

The root zone operators try very hard to balance those around where there is the most need. You can see that the spread is quite good. You probably can't read the numbers from the very, very back, but each one of those circles is filled with all sorts of numbers, ranging from 3,4 to 39,66, and 58 in various regions so that we serve everybody equally.

Going onto the root zone management, how is it managed? Where does the data come from? Who's serving it? What actually happens? This is the breakdown within the ICANN community of how the process works.

The TLD operators on the far left-hand side – .com, .sce, .horses – all have a list of name servers and a list of addresses that have to go into the root zone in order for people to find them. Those change requests, when they might want to change their name servers or change their address, go to IANA, which is a part of ICANN. IANA will make the change.

This slide is actually a little bit old because it still says NTIA on it, and as you all know, the transition just happened. So we need to update this diagram – oh, is it in the next one? Okay. I'm sorry. So this is actually prior to IANA. I should have read the title. Prior to IANA, the NTIA would oversee those changes and make sure that all of the changes were proper and good. IANA would make the change, and the root zone maintainer, which is Verisign at

this time, would then distribute those to all of the root zones, which are to the right-hand side of that squirrely line going down the middle.

All of the letters would the redistribute that data to all of their individual servers. As I said, there's over 600 total across the entire world. This propagation happens over the course a day or two for those changes to be made.

Again, on the far right-hand side, all of the DNS resolvers will go make queries to one of those 600 instances, and one of those 13 v6 addresses actually propagate to many, many, many more instances using Anycast, which, again, we'll go into in a little bit.

Next. Today, now that the IANA stewardship transition has happened, the NTIA bubble that you saw at the bottom is not there anymore. So the IANA makes changes and the root zone maintainer distributes those changes. Now there's a community oversight that makes sure that this whole process is still fluid and happens properly.

Next. Let's talk about the features of the root zone operators today. There's still diversity. This is still a strength. By having diversity, it ensures that there's no one political power that has the ability to take over everything. There's a diversity of organizational structures. There's still some government labs. There's still universities. There's still for-profit companies.

There's not-for-profit services. There's diversity in the operational history. Each organization has their own distinct history of how that root zone operator has held its systems over the years and years. Some are newer. Some are older.

There's a diversity in hardware and software that are in use. Different organizations buy different vendors' hardware. Different operating systems support different name servers' software.

Then there's best common practices that refer to a minimum level of what their physical systems security is, how they over-provision how many Anycast instances they have, and how they make sure they handle well under a load. Then they have a very different set of professional and trusted staff that is in use.

All of this diversity actually adds up to a strength in the long-run.

Next, please. They all perform various levels of cooperation and coordination amongst themselves and with the community at large. The cooperation takes place at many, many meetings that you are all aware of, like ICANN and IETF or the common ones that many people in here associate with, but also the operational communities, like RIPE and NANOG, and some research communities, like DNS-OARC, and then the RARs as well, like APNICK and ARIN and AFNOG, which is another operational one. Also in the Internet itself you can see forums

where the root zone organizations are actually responding to questions.

There's permanent infrastructure to respond to the possible emergencies that are in place. There's communication paths between the root zone operators on how they talk during an emergency. That includes everything from telephone bridges to mail to other forms of communication.

There's periodic activities to support the emergency response capabilities to ensure that we're all running with up-to-date practices – I'm so sorry. We're going to turn off now.

There's coordination with the established Internet bodies. RSSAC within ICANN, which we'll talk about in greater depth later, is the Root Server Stability Advisory Committee. We also participate in the standardization bodies, especially within the IETF, about how the DNS changes over time to make sure that the root server system can be up to date with the latest changes in the DNS technology and be able to serve that properly.

Then there's data sharing through DNS-OARC to do research studies and things like that on how the root server system can be studies for its research-related activities in terms of being able to see how it's being used effectively and to see what conclusions we can draw that will actually affect the future of the system as well.

Next, please. As things evolve, new requirements are constantly put on the root DNS system as a whole, and that includes the root server system. We constantly analyze the impact and adopt new uses and protocol extensions. Examples are the IDNs and DNSSEC and IPv6 and packet sizes and negotiation of what packet size a resolver can handle is.

We're constantly increasing robustness and responsiveness and the resilience. We do that through the wide deployment of Anycast – again, over 600 instances around the planet – and the general consensus is that the best operationally that you can get is to have more instances of not just the root zone but more instances of any zones that you use on a regular basis in order to get proper latency response and proper serving in your local area.

Next, please. There are a number of myths that are out there and float around. We'd like to talk about those for a minute to make sure that we correct some notions that we constantly here that are actually incorrect.

One myth is that root servers do control where the Internet traffic goes. They don't. All the root servers do is just answer a response to a query. A lot of times, those are addresses, but we don't actually control where you send traffic to that address. The Internet traffic is actually controlled by routers that control

where stuff happens and goes on the Internet once you have an address.

Most DNS queries, as I mentioned before, are not seen by the root servers. Because of that caching mechanism and because the TLDs are cached for such a long period time, the TLDs actually see a lot more traffic than the root servers. That's just where to go ask the first question. After that, you remember it for 24 hours and you don't go talk to them again.

The administration of the root zone is separate from the service provisioning. ICANN administers the root zone. The root zone operators are responsible for serving data that comes out of IANA. The administration of that data is not done at all by the root server operators.

None of the root server letters are special. It's commonly believed that A is the master. It's not. All of the letters actually serve the exact same data, and A is treated in the same way as the rest of them. That's a myth that we still hear constantly.

The root zone operators are not hobbyists. They're all professional organizations that are in it for the long haul and have been for a long time and are committed to serving the Internet users worldwide.

There are not more than 13 servers. There's 13 technical identifiers. There's A through M. There's over 600 instances of those various letters around the planet.

The collective root zone operators coordinate as a whole. Each individual root zone operator operates their own system without help from others. They communicate on a regular basis, but one of the important elements of that diversity that I spoke of earlier is that each operation does its own thing and towards what it believes is needed for the Internet users. That diversity is actually a prime benefit of having that spread.

We're going to dive down next into Anycast. We've talked about it a lot. There is a large number of instances. Lars is going to go into exactly how that works.

There's a question first.

UNIDENTIFIED MALE: We have a comment and a question remote. I don't if we want to split the difference between anyone on the floor first since we took a remote first last time.

WES HARDAKER: Okay. Where's the one on the floor?

UNIDENTIFIED MALE: I don't know. Do we have any – okay.

UNIDENTIFIED MALE: I got it.

UNIDENTIFIED MALE: Hi. My name is [Zanucom]. I'm from the Internet Society, Kolkata chapter. We host one of the instances of L, and we are also working on hosting one of the instances of J on an IXP model.

What I have seen is that most of the burden of 600 instances is taken by other L or J, whereas there are other mirrors like A or the one operated by NASA. They don't have much instances across, so is there any plan to streamline that piece so that everybody has an equal amount of mirrors? Or why is it so that it is not so?

WES HARDAKER: That's an excellent question. Speaking as one of the letters that don't have many, I feel obliged to answer it. Again, the diversity is that each organization chooses how many that they want.

The hard thing about that question is: is there too many of a particular letter? [Is there] actually getting benefit? So the right thing to do is actually to do a study in each local area and find out, "Are you being properly served?" and if not, absolutely

reach out to the various root zone operators to ask, "Would you be willing to have me host something locally?" That's a prime thing to do.

It's unlikely that you need 13 right next to you, but there's probably a magical number that you do need near you. Each ISP might have a different answer for that, or each region within the country, depending on how well they're connected.

There's also some new technologies within the IETF about how, if you're running behind a very, very poor service, where you can't get anything, there's actually a loopback RFC, which is number…

UNIDENTIFIED MALE: I think it's 7706.

WES HARDAKER: It might be 7706, which talks about how to deal with running in a very poorly constrained environment as well. So you may look into that.

UNIDENTIFIED MALE: We have a comment from Davey Song. He says, "Root servers is just a network system to replicate and distribute the root zone file around the world. You can set up your own root server for

yourself or your customers by downloading the root zone file and validating the data with the signature all by yourself."

LARS-JOHAN LIMAN: Yes, you can, but…It's technically possible to do that, and Davey Song works for an organization that actually does a lot of experimentation with that – the Beijing Internet Institute.

The problem is to coordinate so that everything runs at the same pace, that everything works so everything is updated at the same time everywhere.

If you have a local root name server and decide to do that, you possibly create – when you deal with Anycast, there are all kinds of extremely complicated problematic situations that you can end up in that are really hard to predict. If you go that path, you create something where your users cannot predict what's going to happen.

If they expect to talk to a root name server which has certain content and there is someone else a root name server which doesn't have the same content because it's maybe not updated at the same time, then you have an inconsistency on the network that is very negative for the end users. That goes back to the old criteria that Jon Postel set up for running in a root server operator to make sure that the data is consistent.

That said, Davey Song is actually technically correct. The content of the zone, if you use the authentic root zone, which you can download, which is public, you can create a system where it is possible to have the correct data and have it verified.

But having done the journey for the past 25 years, I can tell you that making that work is not something that you want to try, other than on your own desk at home. If you want to provide that to any subgroup of people, you have to be extremely careful so that you provide the correct data.

If you start to use our IP address, we will come after you because the IP address that we use is a service mark for us for I-Root. So we want to know that people who use our IP address come to our service so we can guarantee the service back to them. So if someone else tries to do that, we will be rather angry.

WES HARDAKER: Sorry. Are you done? Because I have one more point to add.

LARS-JOHAN LIMAN: I guess I am done.

WES HARDAKER: Okay. One more important thing is that, if you try to stand up your own root server system, which you can do, the first thing

that happens is that, when your local resolvers try to query it, they're going to get the list of current name servers that you have in there. If you're serving the IANA one, it will be a signed copy of the current 13 letters. They'll immediately direct traffic away from you to the current 13 letters.

So if you were going to do that, you would actually have to replace the list of NSSETs, which would actually not validate properly according to DNSSEC. So there's a whole bunch of other things to consider about that.

UNIDENTIFIED MALE: We do have two more questions remote, but I want to balance that. Are there any questions in the room here? We may have a couple back there, and then I'll read another remote one.

PARMINDER JEET SINGH: I'm Parminder. I just want to know how the DNSSEC complicates the possibilities of, as somebody was saying, someone to run one's own root server – because you wouldn't get a signed security certificate for that, right? – and whether that issue complicates alternative possibilities.

Second is – though I should have asked this question after the definition of Anycast, but I'll do it now since I hold the mic – that the 13 root servers are replicating the authoritative root zone

files, republishing it. The Anycasts is republishing the data from these 13. What is the difference between the first kind of republishing and the second kind of republishing? I understand both are not automatic. One has a manual step. I would like to know what the difference is between the two kinds of republishing.

LARS-JOHAN LIMAN: If you take the DNSSEC part, I'll do the Anycast part.

WES HARDAKER: Fair enough, fair enough. The one thing about DNSSEC is, no matter if you could have it copied from one person to the next to the next to the next to the next, if you are using the IANA DNSSEC root key, it doesn't matter how many times you copy it. You could tell if anything in it was modified. Anything.

So DNSSEC requires that all of the data in any zone from the top all the way down be not modified. That is what it's guaranteeing you for.

Your question was about if you run your own root server. What would happen is, if you distribute the current IANA file – again, it has the 13 names – the 13 addresses in there would not be the ones that you would be hosting. So you would have to replace the name server record in the file that you were hosting yourself,

and it would be immediately noticed by anybody that was a validating resolver as not authentic and it would be ignored. So you really can't do that from a DNSSEC point of view.

LARS-JOHAN LIMAN: Speaking to your other question regarding the difference between the letters and the servers, this picture may give you slightly the wrong impression. The root zone maintainer makes the zone file. The zone file is edited. The content of the data is handled by IANA. The root zone maintainer makes that data available to the operators.

So the letters here are not servers. They are organizations that operate the letter. They pull the data from this distribution system that the root zone maintainer provides. This is not one machine. This is a set of machines for redundancy.

Each organization copies the data to its own distribution system, which then further copies the data out to the actual servers that provide the service.

So when you say a letter, that's not one server. It's any one of these servers. If this is a D, then this is operator D, the organization that handles D-Root. Then any one of these is a D-Root, and they are absolutely identical. That's part of the work for a root server operator: to make sure that these are absolutely

identical. You will not be able to tell the difference. The data here is exactly the same as the data here because it has been copied all the way through here.

Again, M-Root will have its set of servers, which has exactly the same data because we copied from the same source. We only transfer it through our own system. It's just a pipe that pushes it out to all the individual servers.

Now, A has a set of servers. B has a set of servers. C has a set of servers. And they are all working with identical data. There is no difference.

Also, now I'll speaking only for I-Root because I don't have authority to speak for the other operators. In the case of I-Root there is no manual step involved here. When the root zone maintainer has generated a new version of the zone file, we receive an automatic notification to our system, which then goes and fetches the zone file from the root zone maintainer and automatically sends out notification to all our more than 50 instances, which then come and pick it up from our servers in Stockholm. There are also more than one.

So the file is copied, copied, copied, copied, copied, copied automatically out to all these end nodes, and there is no server here that you can send queries to. There's only a set of identical servers here, and none of them is better than the other because,

when you come to digital data, it's master and copy, because if you have a file and you copy the file, it is identical. You cannot tell the difference, and if you cannot tell the difference, you cannot tell which one was master and copy. It's irrelevant. It doesn't make sense to talk about it.

WES HARDAKER: One more thing about the manual. It's pretty easy to tell that there is no manual steps, because if you go send a query to all the root letters at once and you query for the SOA record, which is the very top-level record in a zone file, you can get the serial number.

I challenge you to go find the case where one server is actually responding with a different serial number than the rest. If there was a manual step involved, you would see a delay in how those serial numbers are propagated.

And it's not to say that there may not be a delay due to technical reasons one way or the other, but that should not generally happen. Generally, when you go query them all, they should all be serving the exact same copy at the zone within a very small period of time, which I'm not going to define because I haven't done the measurement recently.

LARS-JOHAN LIMAN: I can speak for I-Root again. For I-Root, the normal delay and the normal operation from – we received the notification. We do all the copying out to all the instances. The slowest instance gets updated in seven seconds.

UNIDENTIFIED MALE: I have a total of four questions. I see two on the floor and two here in the chat room. The next questions from the chat room is from [Vivani Shinkur]. He asks, "Will we run out of IPv6 addresses somebody and take IPv8? Or is IPv6 never going to be filled for the next century?"

LARS-JOHAN LIMAN: I'm old enough to remember the days when IPv6 was designed. One of the problems was that we thought we would run out of IPv4 addresses. This was back in 1992/'93/'94. So there were lots of proposals for IP version 6, and eventually they honed in on one of the proposals and made they made the negotiations and so on. That's what we're running today.

They also needed to design the length of the address, how many bits should there be in the address. Now, you have to remember, the old IPv4 addresses have 32 bites, 32 1s and 0s. That gives us roughly 4 billion hosts, theoretically.

If you add one single bit, one 0 or 1, you have doubled that number to 8 billion hosts.

Okay. So how many do we need? Someone said 64 bits. Now remember, that's not twice what we used to have. That's 4 billion times what we used to have. That's a large number.

Eventually they went back and forth and ended with 128 bits. That number has more digits than my age. Someone did a calculation and said, "With that many addresses, we can put an address on every electron on every atom in the entire universe several times over." We have a lot of addresses to play with.

I'm not going to promise you that we will never run out of them because one of the reasons we are running out of IPv4 is because we were stupid in the beginning and the way we handed out addresses was not very smart. But I think we can live with the IPv6 addresses for a very long time to come.

UNIDENTIFIED MALE: We had a question in the back on this side of the room. Whoever had the question on this side, please raise your hand again. If not, I'm going to the other side.

UNIDENTIFIED FEMALE: [inaudible]

UNIDENTIFIED MALE: Okay. Right there, then. Anupom.

ANUPOM DEY: Just a follow-up question. This RZERC and RSSAC – is it a subset or a whole [delegated]? Who are the people in that?

WES HARDAKER: That's a good question. RZERC is a very new organization. I'm going to defer that until after the last set of slides, where we're going to talk about RSSAC specifically. Then, if you still have the question, come back.

UNIDENTIFIED MALE: A question online from [Sayid Sharajuden]. "How do you ensure that zone files are properly replicated? Is there any chance a root zone file is getting corrupted by any attack or malware?"

LARS-JOHAN LIMAN: The DNS system has a number of security additions that have been added since the start. We've already talked about DNSSEC, but that is one way to make sure that the data you receive hasn't been modified anywhere in the system, regardless if it was the transfer to the root servers or if it's the query cycle where you talk to your caching resolver. The data actually moves from IANA

all the way to the end user. The end user can check all the steps in between by using DNSSEC.

In addition to that, there is something called transaction signatures, and that's something that are used in these steps here. When we as an operator transfer the zone file from the root zone maintainer, the root zone maintainer adds crypto-signatures to file transfer. We can then verify that when we receive…Hello. I was trying to use the picture. I was trying to use the picture. Can you please put slide 26 back?

We can verify when we receive here that the crypto-signatures are okay. That's a guarantee that nothing was modified in the one transfer. That's how we use root server operators ensure that we operate the data that is provided from this source.

We use the same mechanism in the next step when we distribute from our servers in Stockholm to the various Anycast instances. We use the same type of crypto-signatures.

This crypto-key that we use for these signatures is negotiated by the root server operators all together, and it's changed on a regular basis. So we are pretty certain that we actually serve the correct data here.

WES HARDAKER: Okay. One more question. Then we need to actually move on because we have a lot left to cover.

UNIDENTIFIED MALE: Okay. It's a very simple question. I'm just wondering, in the post-IANA transition, the NTIA is gone. So where is Verisign? Is there any [root]?

WES HARDAKER: Verisign is still the root zone maintainer right now, but that is a role that is designed to change if need be in the future. That's something that ICANN as a community can change. There's a contract with Verisign right now as the current root zone maintainer. Whether they stay that way for ad nauseam, that's up to the community at large.

LARS-JOHAN LIMAN: I'll try to continue on how Anycast works. Next slide, please. There are several, but in this case here for the DNS, there are two alternative mechanisms to provide the service from an instance.

Now, Unicast and Anycast are pure networking things. They can be used for any service, and it's not specific to the DNS. The DNS was a good poster child. It was a good place to start using this service.

The traditional way of providing service on the Internet is Unicast. You have one single host, which has a unique IP address, and that's where your servers. It could be a web server or a mail server or a DNS server. Anyone on the Internet who sends a packet towards that server address will eventually reach that one single server if everything works as intended.

That means that it receives all traffic that's going to that IP address. That also means that, if you have some kind of attack – one common type of attack is Distributed Denial of Service attack (DDoS), where a lot of hosts send traffic towards a specific IP address. When that all gets close to that server, it will jam that server and it will not be able to operate properly.

Now, Anycast is a trick, if you wish, that we use on the Internet. It has nothing to do with the DNS server, per se. Anycast is implemented in the routers, the switches on the Internet. What it's all about is that we can have two servers – we start off with two – that actually have the exact same IP address, and we let the routers on the Internet decide where to send the packets. So the sender doesn't know which one of these two packets he'll end up with. If I send a packet from Stockholm, I don't know if it will end up in Washington or in Tokyo.

The good news is that it doesn't matter because the server in Tokyo and the server in Washington are kept in absolute sync.

So I get exactly the same response, regardless of whether I'm talking to Tokyo or Washington.

If you have a distributed denial-of-service attack, you have lots of hosts on the Internet trying to reach the same IP address, but they cannot decide which one of these two hosts to hit because the network in between makes the decision for them. So in good numbers, half of the traffic will end up in Tokyo and the other half in Washington.

Now, if we multiply this and say, "We don't only have two. We have 55 of these sites," the routing system on the Internet will carry your traffic to the nearest server. Now, "nearest" is a very strange term on the Internet, so don't be surprised if you end up in very strange places. The idea is that I as an end user will reach the nearest instance of the service that I'm trying to reach.

I'll try to explain this on the next slide and the following. If we look at the Unicast case first – this is the traditional – we have a source – the green dot that sends a packet towards the blue dot. Regardless of which way in the network this packet goes, it will always end up at this unique host.

If you send a packet from up here, it will take a different route through the network, but it will end up here. So this is the unique host serving this. That means, if you have a large number

of sources, a large number of clients, they all end up here, and it's easy to overload this link by just sending many packets on it.

Next slide, please. In the Anycast case, you can have, in the example here, three destinations that are absolutely identical. They have the same IP address, and the router in between here or here will make the decision on which one is the closest one. So when the source sends its packet, it comes here, and this one decides, "Oh, the nearest blue dot is going downwards." And this one can actually make three decisions: either downwards, going to the right, or going to the upper right.

There is a way to reach a blue dot there. There is a way to reach a blue dot here, and there is a way to reach a blue dot here. But it decides that this one is the closest one and sends the packet here. This one provides the service. It sends the response back, and the client gets his response.

So the routers, the switches on the network, make the decision on which one is the closest one. To be quite honest, this one doesn't know that there are three blue dots. It thinks it's the same blue dot and that there are three different paths to reach the same blue dot, and one path is much shorter. The fact that they are different actual servers is totally hidden to this machine here, this router. So it think it has redundant paths to these multiple instances.

Next slide, please. Now, if you have someone who tries to attack, they will only reach the nearest server. The client over here will still get good service from this destination. If you have a lot of red dots that try to attack, only the red dots that are close to this blue dot will hit this. The red ones up here will hit that one. The red ones down here will hit this one. So only a small part of the traffic will hit this one. Another small part will hit this one. Another small part will hit that one. So we distribute the attack over multiple hosts. By doing that, we can increase the capacity in the overall system, the entire system.

If you think now that you have 600 blue dots in total of an entire network – they're divided into subgroups per IP address – we can withstand a lot of traffic.

Let's try the next slide. I'm not quite certain – oh. Please back up again then. I want to say one more thing here – oh. That's – one more. 34. That's good.

Now, if this server was to disappear from the network – it crashes or it's taken down for service or some component breaks here – and this router will no longer believe that it can send packets to the blue dot this way. But it still has this way. This is another way to the blue dot, and this is an automatic failover system.

So if we take out this server for some reason – crash, service, attack, or whatever – the traffic will automatically find its way to another one of these Anycast instances. That's a very, very good relief for us as operators.

Netnod brings down servers every week – often several of them every week – to do upgrades, maintenance, and service. Did you ever notice? Probably not. I hope not. Then we put them back again, and suddenly this one realizes, "Oh, here it is. There's a much shorter way to the blue dot. Okay. It's back. Good. I have a better way. I'll send the packet that way instead."

So this all happens automatically on the network by the routers receiving configuration information from the blue dots about where this service is available, and they will choose the nearest one.

Thanks. Shall we jump into the next chapter and do final questions afterwards?

WES HARDAKER: Okay. We're going to dive now into the ICANN policy realm and into RSSAC and what that organization is and what it has been doing recently.

Next. I'm going to read this straight off because this is functionally 100% true in terms of it being very short and brief and is exactly what RSSAC is supposed to be.

"The role of the Root Server System Advisory Committee, otherwise known as RSSAC, is to advise the ICANN community and Board on matters relating to the operation, administration, security, and integrity of the Internet's root server system."

It's a very narrow scope it doesn't do a huge amount of stuff. It is only talking about the root server system and how to advise the ICANN community and the ICANN Board.

Next. RSSAC is a committee that produces advice primarily to the Board but also to other ICANN bodies and other organizations within ICANN involved in the overall DNS business at the root level. The root server operators are represented inside RSSAC. Each organization has a representative within RSSAC.

But the RSSAC does not actually involve itself in operational matters. RSSAC is not involved with the technical operation of the root server system. That's an important distinction because up until now we've been talking a lot about the technical stuff, about how the root server system runs. RSSAC does not actually manage the technical operation of the root server system.

In terms of the organizational chart within ICANN, we are over on the right-hand side, along with the other ACs over there. It's composed of three primary sources of membership; one, appointed representatives of the root server operators. Two, each organization also has a primary and an alternate representation. Then there are liaisons to some other organizations, both within ICANN and without. That includes the ICANN Board, for example, as well as without, including the Internet Architecture Board (IAB).

There's also an RSSAC Caucus. This is actually where, really, all of the work gets done. It's a body of volunteer subject matter experts that become a member. Anybody can become a member of the Caucus and help work on the documents that RSSAC produces.

The members are confirmed by RSSAC based on a statement of interest. So to become a Caucus member, you need to fill out a form that says you have interest in working on the stuff that RSSAC produces, and then you can become a Caucus member and help work out on the various work parties.

The current RSSAC Co-Chairs are Brad Verd from Verisign and Tripti. Can you guys raise your hands just to say hi? They're from A-Root and J-Root and D-Root.

Next, please. For the RSSAC liaisons that I referred to earlier, this is actually the complete list, including the IANA functions operator, which is now PTI, the root zone maintainer, the Internet Architecture Board, SSAC, the ICANN Board, the Nominating Committee, the Customer Standing Committee, which is fairly new (CSC), the Root Zone Evolutionary Review Committee (RZERC), which was mentioned before – do you have a comment?

UNIDENTIFIED MALE: [inaudible]

WES HARDAKER: Oh, okay. There's our CSC representative there. Next, please. The Caucus members are anybody with technical expertise [inaudible] to which do not actually work in the root zone operations themselves. They declare a public statement of interest about why they want to be a part of the Caucus. All of the documents produced include names of the people that actually help produce that document within the Caucus.

With the RSSAC documents, if you go look at the bottom of them, you'll find it doesn't say the list of the RSSAC members. It actually lists all of the Caucus members that contributed to the document.

The purpose of the Caucus is to bring together a diverse level of expertise who can work on these publications. We'll talk a more a bit later about some of the recent work that has been done. It's important from a transparency point of view to have the Caucus be a fairly open body. If you have DNS expertise in the technical community, you, too, can become a member and help participate in the construction of these documents.

The Caucus is really a framework for getting all of this done. It's a framework for enabling the community at large to actually help work on the documents that define the stability of the root server system and how it evolves over time.

If you want to apply, there's a website about it. If you search for RSSAC and ICANN, you'll find it. I suppose you could search for Caucus or you can e-mail that address as well.

Recently, there's a couple of things that have come out most recently since the last ICANN meeting. RSSAC Workshop 2 on the RSSARC report has come out. RSSAC is continuing to hold some on-the-side workshops in order to do some brainstorming about the future of the root server system. That second report came out in June.

There has also been some statements that have come out. There's a response to the GNSO Policy Development Process Working Group on their new gTLD subsequent procedures and

the fact that we might be going into a second round of gTLDs and our view on that subject and our view on that subject.

There's also an RSSAC statement concerning the impact of the unavailability of a single root server. Some analysis was done to look at past events and see, if one server goes down for a while, if any of the Internet users be actually visible affected. The answer is no. At least a temporarily outage of a single letter would not cause an impact to pretty much the entire Internet, due to our over-provisioning that we provide within the root server system.

There was also a RSSAC statement on the client side reliability of root DNS data.

Next slide, please. The RSSAC Workshop 2 was held in May of 2016 in Reston, Virginia. It focused on three themes: the architecture, evolution, and the reinventing of RSSAC itself; were changes needed organizationally?

One of the important agreements that came out of this is that RSSAC agreed to be the front door to technical questions about the root server system. If you had a technical question about the whole root server system and how it operates and it needed answers from the root zone operators, you could ask RSSAC. The Chairs of RSSAC would relay that question to the operators and make sure that you got an answer back if it was within scope.

There were three statements that were conceived at the workshop: the RSSAC statement concerning the impact of the unavailability of a single root server system that I mentioned already, the RSSAC statement on the client reliability of root DNS data, and the key technical elements of potential root zone operators. That one has not been released. The Caucus is still working on that, but it is coming out very shortly.

For the RSSAC statement on the client reliability of root data, that was published in June of 2016. It reiterated a few things. It reiterated that the operators are committed to serving the IANA global root DNS namespace. The root zone operators are only reserving what IANA has given them.

All of those answers are complete and unmodified and DNSSEC-signed. So they served the DNSSEC-signed copy of the root zone file. Thus, you can guarantee that the data has not been modified by anybody in that little organization diagram that you saw earlier.

That's it. Thanks.

One of the other documents that came out was the unavailability of a single root server system that I mentioned earlier. It came out in September. We looked at existing data in terms of what has happened in past attacks. Because of the high redundancy – those 600 instances that are serving root zone

data – as well as the fact that records are cached for so long, the loss of a single root server would not cause any immediate stability issues within the root server system.

There have been outages of various letters over time. Looking at that data, we could actually not find any user-visible detection that would have occurred that would have actually prevented somebody from accessing the Internet.

The response to the GNSO PDP on the new gTLD process. If future plans were for more top-level domains, more gTLDs were going to come out of an ICANN process, as long as it was consistent with the way that the past gTLD program was done; that we don't see any problems with it impacting the root zone stability.

Most importantly, we recommend that, of course, the root zone management partners and the root zone operators continue to be engaged in that collaborative process to make sure that we don't see anything, as the New gTLD Program continues, that might impact it that was not seen at this time.

The current work that RSSAC is working right now: there is a history of the root server system that documents a lot of what Lars actually went over earlier in terms of the evolution over time but in even greater detail that you might want to read if you're a history buff.

There's the root server naming scheme. There's a discussion of what should be the – right now, everything is a.rootservers.net, b.rootservers.net. Is that still the right naming scheme to continue into the future?

And then there's the key technical elements of potential root operators. What's the minimum criteria that you must do in order to be a root zone operator?

Finally, there's the distribution of Anycast instances. This is where local feedback from local ISPs is very important: that Anycast be widely distributed and make sure that everybody has available service locally to themselves. There's a study looking into what is the best way that we can deploy Anycast among all the root server operators.

Going further into those, the history of the root server system contains a chronological history, so you can actually follow along over time on how the root zone has changed and how the name servers have changed within it. It contains a description of who the current operators are, not just now, but at any particular point along the change of root zone operators over time.

Next. The root server naming scheme that I mentioned earlier documents the technical history of the names that were assigned to the individual operators. It considers various

alternatives. Should the names change? Should we change from b.rootservers.net to whack-a-mole?

They considered a bunch of different options, including just having no suffixes – so a letter name server straight in the root zone to a number of other things – and they analyzed the pros and cons of all those choices and then performed a risk analysis and made a recommendation in the long-run about what future work needs to be done in order to complete that task.

There is also, as I mentioned, the key technical elements of potential root zone operators. It lists the important things that root zone operators must do in order to be an effective root zone operator to make sure that they are serving the Internet community at large.

It's based originally on RSSAC001 and RFC7720 as starting points. Those are older documents. One of the reasons that this work was done was to revamp that list and bring it up to date with current design methodology.

It contains all sorts of things, including design, experience, networking, diversity, documentation, data, measurement, and all those technical requirements. It's just a technical document of what a root server operator must possess.

Finally, there's the distribution of Anycast instances. This work is really trying to decide what the best way is to distribute Anycast instances in order to properly serve the global community, and, given the current state of the Internet today, what the maximum latency is that somebody ought to be experiencing. Would adding more instances in different locations help you? If root operators were to coordinate their developments of Anycast instances, what consideration should be contemplated? Is coordination necessary? Would there be conflicts if there's no coordination done, or if there is? Is there any regional or global risks if only a subset of operators deploy Anycast instances? Which goes back to the earlier question of, do you need an instance of every single letter near you?

Next. That brings the end of the discussion on RSSAC. If there is any more information that you might want to look at, there's a couple of important things. One, we have webpage. Also there's a webpage with a list of all of our documents that we have published to date. If you're interested in joining the Caucus, that information is there as well.

I encourage you to come to the RSSAC public meeting, which is on Sunday at 1:45, if you're interested in more information about what the RSSAC has done recently. We'll go into some of these topics a little bit more in depth as well.

I think that's the last slide. That's the last slide. Is there any questions?

UNIDENTIFIED FEMALE: Can somehow DNSSEC be not protected from the fast flux and the double fast flux?

WES HARDAKER: From which attacks?

UNIDENTIFIED FEMALE: Fast flux and double-fast flux, where the SO [inaudible] changes [around] the DNS.

LARS-JOHAN LIMAN: To be frank, I have not a detailed knowledge about the fast flux system, but my guess is that it does not because fast flux builds on authentic updates. So these are true updates that go into the system, and they are distributed through their SO system. They happen not at the root level. They happen further down the tree.

Warren?

WARREN KUMARI: Sorry. I was just going to say the same thing, that they're much further down in the DNS hierarchy. There isn't fast flux at the root, so the root doesn't notice or care.

LARS-JOHAN LIMAN: But the question was actually whether DNSSEC provides any protection, and it does not because it's the correct data. It just changes very quickly.

WES HARDAKER: Okay. Any other questions online?

UNIDENTIFIED MALE: There are no questions online. Any other questions in the room? One way back there.

WES HARDAKER: That's okay.

ABDALMONEM GALILA: I am Abdalmonem Galila, a second-time Fellow. I have a question. You said that we have 13 root servers. From a technical review, regarding performance, [is it] okay – performance and the resiliency. So is the number of Anycast nodes unlimited or limited to a specific number?

LARS-JOHAN LIMAN: I would say, in principle, it's unlimited. There is no limitation in the DNS protocol for how many Anycast instances we can have for any given letter. So we have lots of headroom there. We can deploy lots more Anycasts instances.

Speaking for I-Root, I can tell you that we are interested in doing that, definitely. We try to maintain a dialogue with people with people from all over the world to find out where the suitable place is to deploy these new instances because there's a lot of detail that you need to figure out. One of them is actually finance. So there is a lot of room for many, many, many, many more Anycast instances.

WES HARDAKER: One important consideration. A colleague of mine at USC actually did a study on how many more Anycast instances help you. John Heidemann is his name. You could go look up his research paper. He shows that, after some point, you really don't get any benefit from a latency point of view. If you're in an area that has poor routing and things like that, that's a different question. But from a latency point of view, you actually don't get a huge amount of benefit by adding a significant number more if they're properly distributed.

You have to balance that question against the stability. If I had two millions servers running nodes, would that be stable? So there's a balance of there of: where's the exact ideal number? That's a good question, but I'm not sure it's quite as straightforward as a million.

Any other questions? I think we are about out of time.

LARS-JOJAN LIMAN: I will remain here for the rest of the week. I'm tall. You will find me in the hallways. I'm happy to talk to any one of you, always. Please come and talk to me if you have questions, if you have ideas, if you want to talk about the root servers or anything.

UNIDENTIFIED MALE: They will also be here at the exact same time tomorrow, so if you wake up jetlagging and you have some kind of epiphany question, come join us tomorrow, too.

WES HARDAKER: There'll be different people presenting that day, so you will hear a different set of words. We do thank you for putting up with our jetlag from today.

UNIDENTIFIED MALE: Let's please thank Liman and Wes – oh, one more question. Sorry. Liman, a question around that side.

LARS-JOHAN LIMAN: Yes?

WES HARDAKER: I need my exercise for the day.

LARS-JOHAN LIMAN: He runs faster than I do.

VALERIE FAST HORSE: Hi. I'm Valerie Fast Horse. I'm from the Coeur d'Alene Tribe in north Idaho. I had a question on the participation requirement. If we were to sign up or send a delegate to participate in the Caucus, what would be the requirements?

WES HARDAKER: I think if you look on the RSSAC Caucus website, which I can't quote from memory, it has the list of requirements. You have to have some technical knowledge of the DNS and be able to understand.

VALERIE FAST HORSE: I'm sorry. I didn't mean technical requirements. I meant time requirements.

WES HARDAKER: Good question. The expectation is that, if you become a Caucus member, then you can contribute to work parties. If you are only contributing to a work party just to listen, you're expected to contribute. I don't think that we have a hard guidelines on "You must contribute at least so many hours," but realize that it is a work party to get stuff done. The goal is that everybody should have a voice in that opinion, and if there's not a voice being heard, then we question why you're doing it.

LARS-JOHAN LIMAN: But that doesn't mean that you have to participate in every work party every time because there are different types of issues. Some things may be within your area expertise, and then there are two parties where you feel that you have less to contribute, and then there's another one where you can [yourself].

So there are no strict rules, but there is an expectation that one will contribute when one joins.

WES HARDAKER: Very good point.

UNIDENTIFIED MALE: Last chance for questions. Anyone else?

LARS-JOHAN LIMAN: Going, going…

UNIDENTIFIED MALE: I'd like to thank Liman and Wes for today.

I hate sitting down and speaking. I want to thank you guys all. Many of you have spent the whole day with us, and I appreciate that. Many of you have joined for the last couple sessions. I also appreciate that.

If you've missed any session today that you wish that you had seen, we are doing the exact same schedule tomorrow, starting with DNS fundamentals with Internet workings and routing and things like that with the Registry Operations Workshop, and then again with the root server operators but with different root server operators, different secret letters.

Please come and join us tomorrow if you missed any session. Otherwise, tell you friends and have them come and join our session.

Have a great evening, and we'll see some of you tomorrow.

[END OF TRANSCRIPTION]