HYDERABAD – How It Works: Root Server Operations
Friday, November 04, 2016 – 15:15 to 16:45 IST
ICANN57 | Hyderabad, India

UNIDENTIFIED MALE: Okay. If you've made it this far, I guarantee you it's worth your while. RSSAC has traditionally given us super-interesting slides and presentations. We did this yesterday, and they didn't make us upset. It was a really interesting session. So I hope you enjoy it today.

We have Daniel Migault and Brian Reid today, and they will presenting on the various aspects of the Root Server System Advisory Committee – I'm jetlagging; pardon me. Without further ado, I'm going to go ahead and hand over to Daniel. Thank you.

DANIEL MIGAULT: Thank you. Good evening, good morning, everyone. Today we're going to [handle with] a tutorial on the root server system. It's going to be presented by the Root Server System Advisory Committee. So we're going to talk about the root server system and also present the Root Server System Advisory Committee.

In the first part, we're going to see an overview of the domain name system – that's just to recap a little bit on how DNS works

– and also to position where the root service stays. From that, as the Internet has evolved, the DNS service/the DNS system has also evolved, so we'll go back and present some history of the root server system and spend a little bit of time to explain how the root server system is today with the new features that it was not originally designed for. One of these features is Anycast, so we're going to provide a little bit of explanation on what Anycast is.

Then we're going to switch from technical aspects to more – it's not administrative, but non-technical aspects – and present RSSAC – our activities, where we are in ICANN, and the publications we're working on.

Let's start with an overview of the DNS system and the root servers. When you're attached to a network, usually you can be reached through an IP address. If I know this IP address, then I can send you any information I want.

Usually it's uniquely assigned. So you're a host. You get an IP address. If I want to reach you, I just have to send you a packet to this IP address. The IP address can be IPv4 or IPv6, and they are uniquely assigned.

Why do we need DNS after all? Well, the purpose of DNS is really to make communications between names instead of IP addresses. First of all, IP addresses are quite hard to remember.

It's like a phone number. It's easier to know you're calling someone than knowing the phone number of someone, and it makes more sense.

How has the Internet has evolved, the use of names can be seen as a better human-readable representation of an IP address. But with how did the networking evolved, the identity and the [rootability] – the locators – have different spaces. So you can have one name that can be associated with multiple IP addresses. On the other hand, you can also have multiple services hosted that are sharing one IP address. So the binding is not one-to-one, and so it's good to have these two different spaces.

The primary purpose of DNS is to make a binding between an identity – usually a human-readable identity – and some locations' information, like an IP address. This is mostly how DNS is being used, but it can also make different bindings, such as mail servers or IP addresses.

The key idea is that, when you want to reach www.example.org, you get the associated IP address for that server.

The DNS system really can be seen like a phone book. So you have all the names and the corresponding connectivity information. So for the whole Internet, it's a book, and because

all information can be updated, it must be a very dynamic database that it scalable.

In order to have something scalable that can match this dynamic requirement, DNS has been designed as a hierarchical distributed database. The names, so the identifiers, start from the root, and then you have top-level domains, and then secondary-level domains, until you have the last leaves.

At the end of the day, the database is formed of small entities that are really [manageable] and spread over the Internet.

Let's see an example. You're a browser and you want to reach one server. Suppose you don't know the IP address of that server. That's quite a common-use case. So you want to be connected to example.org, but you don't have the IP address. That's bad because you need an IP address to connect to that server.

You're going to ask the DNS system, "What is the IP address of this server?" At first, you're going to ask something where the [signature] is a caching server. So you're going to ask this, "What is the IP address of www.example.org?" Well, the caching server is going to check in its cache, and if he has got the IP address, he's going to provide you the response. But if he doesn't have the IP address, then he's going to proceed to a resolution.

So suppose the cache is empty. It has no information. It has received the request to resolve www.example.org. It's going to start by the entry point of the phone book, which is the root DNS server. It's going to send a query to the root DNS server: "What is the IP address of www.example.org? What the root DNS server is going to answer is, "I don't know. But I know who is hosting the .org information, and here is the IP address for the .org DNS server."

So the resolver is going to ask .org DNS server: "What is the IP address of www.example.org?" and the server is going to respond, "I don't know, but I know who is responsible for .example.org."

So the resolver is going to ask .example.org: "What is the IP of www.example.org?" This server is going to send back the expected IP address.

The resolver is going to store this response into its cache in case the request is being made later and provide the response to the end user. Once the end user has got the IP address, at that time he can start the [http] connection.

The important thing to note is that, because we have a caching server, it means that, when the end user a few minutes later wants to reconnect to www.example.org, he has forgot the IP address. So he needs to consult the DNS system and he's going

to ask the cache server. But at that time, the cache server has the information, so it's going to send the response without proceeding to the resolution. Well, in this case, it has considerably reduced the load.

Similarly, if the caching server receives a request for a mail.org or mail.example.org, it's not going to request to the root server again and again because it doesn't have to start from the entry point, but instead, because it's mail.example.org, it knows that it just has to ask the example.org DNS server.

So this caching system considerably reduced the load of the DNS system.

Over time, DNS has somehow changed to adapt to the evolution of the Internet, and security has become quite a concern. The first extension – and a big evolution in this area – has been the introduction of DNSSEC.

DNSSEC enables the resolver to check that it actually received a legitimate response. So it's based on cryptographic signatures. It avoids someone providing an illegitimate response or a spoof response or any kind of things like that. We provide all the information, and the resolver has to perform the check.

Another big concern has been the privacy. As we mentioned earlier, the same question is requested to all the different

servers. In our case, we saw that "What is the IP address of www.example.org?" has been requested of the DNS root server. But we know that the DNS root server is not going to provide this IP address. Its scope is limited to the top-level domain.

If you're taking a taxi downtown, you don't say, "I want to go to the Novotel because my room is 552." No, you just say, "I want to go to the Novotel because the taxi driver doesn't need to know which is your room number.

It's the same here [for] privacy enhancement. We don't want to provide information that is not useful for the resolution.

DNS and privacy enhancement are discussed and standardized at the IETF.

Another big evolution is Anycast. Brian is going to present Anycast in more detail. The principle is that, when someone is sending a request to an IP address, the request is going to a local node, which is very close to you, which means that, in our case, if someone in New Zealand is sending an IP packet to a given IP address, if someone in Alaska is sending to the same IP address, they are actually not reaching the same node, even though they have the same IP address.

It's very useful to prevent DDoS attacks – I mean, it does not prevent. It's to protect against DDoS attacks because then, with

one IP address, it only has a local impact. But again, this is going to be explained in more detail later.

Now, root zones and root servers. You'll remember the entry point is represented by the root server of the DNS system. That was the first request we sent: "What is the IP address of www.example.org?"

As we mentioned, these root servers are only providing information about the top-level domain. But what is different is responding and the information that is being provided. So that's clearly two different things.

The data is actually hosted in what we call the root zone. The data associated with the root zone is being defined here at the IANA. The root server has no action over this zone. It is not able to modify that zone. It's just information it receives and it serves to the end users. So the root servers provision the zone and serve the zone.

Currently, we have 13 root servers. They're designated with a letter, A to M. It's not only a letter because it's a letter.rootservers.net. Well, the real purpose is only to serve a zone.

Just to make sure the difference between the root zone and root server is clearly understood, it's like if you were in a restaurant. If

you want to complain about your meal, it's not actually to the waiter that brings the meal from the kitchen to your table you have to complain. It's to the chef. If the plate falls down, you may complain to the waiter. But those are different things, different roles.

As I mentioned, there are 13 root servers. These are administrated by 12 organizations. These organizations are independent and very diverse. Their role is purely technical. Their goal is, for each of these letters, to make the system reliable and accessible all over the Internet.

Overall, what is also important is that, even though each organization has got some responsibility for this letter, the main responsibility they have is to make the root service available. That's the key role.

As I mentioned, the root server operators – that is, the organization administrating the root servers – are not involved in the data they're serving. That should be clear. Their real goal is just to serve the zone. That's what their there for.

I think that now, if you have any questions, please raise your hand. Then we're going to switch to history.

UNIDENTIFIED MALE: We do have a question on the floor here.

UNIDENTIFIED MALE: Thank you. I have a question. The different BIND and other software works differently. If you want to install a root server local or global in our country, which one should I prefer? Because it's not possible to install all the 13 root servers. So if I want to go with a single root server, which software should I prefer, BIND or some other? Because they work differently. Like, [some work on a randomly-selected] root server, and some software on the basis of roundtrip time or something. That's my question.

DANIEL MIGAULT: What you're asking is: in a resolver, which software should you use?

UNIDENTIFIED MALE: Yes, yes.

DANIEL MIGAULT: Okay. It should not really matter, because as long as you have a resolver, it's going to reach one letter, and as long as you reach once, you will have the same answer – oh, it's not about the answer?

UNIDENTIFIED MALE: I'm sorry. We're streaming and recording, so I want to make sure we have you on microphone.

UNIDENTIFIED MALE: It's about, like you mentioned, increasing the latency. "How can I increase the latency in that term?" is what I'm asking.

DANIEL MIGAULT: To decrease the latency? Well, that's the network, then.

UNIDENTIFIED MALE: Okay. So you just explained to me how it elects from the 13. Whenever a new query comes, there's [no] information in the cache regarding the 13 root servers, so how it selects. In the earlier session, they explained that some select on the basis of randomly electing last time.

UNIDENTIFIED MALE: Oh. Do you want to…?

UNIDENTIFIED MALE: And there's one more issue: the countries whose Internet exchange is not interconnected, they face a [install]. It's a local root server. I [install] that in the location of one city. So that serves to that particular region. If you make it global, it starts

resolving queries to the neighboring countries also. So sometimes it's got [inaudible]. The bandwidth we can increase. So what will be the better choice? Still we are not able to get an exact answer to these questions.

UNIDENTIFIED MALE: One of the things that, in fact, you and I talked about yesterday was that there's multiple steps that happen. When a new resolver starts up, it will query all of the root servers for their information. You will get to a bunch of instances of which it will do a measurement. It will determine which ones you can get to the fastest. Almost all software these days does this, and then it remembers that order. So it randomly queries in the beginning, but then it keeps a memory of where's the fastest one.

So even if, say, you can only get a few of them, and the top two or three are the fastest, those are the ones that most software will query to today. So you actually don't need reachability to all of them. You need decent, low latency to a small number of them, if that makes sense.

UNIDENTIFIED MALE: So it is advisable to interconnect the Internet exchange? Or if install my root server in my Mumbai location, I have to interconnect that exchange to the other exchange points? Or it

will serve the purpose to the whole or I should make that root server global?

UNIDENTIFIED MALE: The more routes that you can get to an instance that might be hosted near you, the better off all of your users will be. That's true not of just DNS but all infrastructure. The more routes you can get, the faster you're going to have access to the data.

BRIAN REID: I'm the next speaker, and I'm going to start my part by a different version of the answer to your question. I'm Brian Reid. I'm in charge of F-root, which basically means that I help pay the people who actually run it. I have been known to log on it and type the root password.

I know how to install root servers. I'm really good at it. I do it basically for a living. I don't really spend any time teaching other people how to do it because they should be doing something else.

We are specialists. Those of us who are root server operators make all the decisions you're asking about. We work with the local host to [site] it, but we are the experts in where to place a root server and whether to make it local and all that stuff. I

guarantee you that all of the other root letters are just as good as we are.

So the real answer to your question is: you don't have to worry about that stuff. What you have to do is find a root server operator who will do the worrying for you. Everything that those guys said is also true.

How come this clicker is not clicking? Oh, okay. It's got to have an actual target.

In the beginning, before there was such a thing as DNS, there was a host file that was maintained in Menlo Park, California, by a woman named Nancy Dorio. If you wanted to put a host on the Internet, you phoned her up and said, "Hey, Nan. Assign me an address and put in the host file." It used to get updated. So nobody really needed a root.

But then, when the DNS came along, there was this chicken and egg problem that Wes was referring to, which is, when you get started, how do you actually launch the beginning of the resolver before it finds a root server? That's complicated, and it's in the RFCs. It's not really part of the root service protocol. It's part of how resolver work and not how root servers work.

The root servers make the assumption that you're going to be able to find them and that, if you can't find them, you should not

be talking to root server operators. You should be talking to the resolver designers and purveyors.

In the beginning, people figured out that there needed to be root servers. Back then, NET10 was a really net and not a NAT net, and so the first root server actually was on NET10. It was sitting on the floor in Nancy Dorio's office at SRI.

I don't remember where NET26 was – probably near the Pentagon – but, slowly, as it became obvious that it would be useful to have more and more root servers, they started coming online. By about 1986, everything was up to date and we had all the root servers we needed, which was four of them. One of them was at a military installation, so it was safe. One of them was at SRI, which meant it was surrounded by experts.

So the technology worked, but then it started – come on, laptop; talk to me. Okay. This is defective technology.

So it became obvious, as the number of users of the Internet increased, that having more root servers was going to be better. Four wasn't quite enough.

So there was one added at NYSERNet in upstate New York. There was one on the campus of the University of Maryland – still is. I don't recognize the gunter-adam.arpa address. That's not around anymore under that name. NASA's Ames [in] Sunnyvale,

California – I guess Mountain View – maybe had one because they had a need and they had money.

Where's the clicker repairperson? Or could you please click?

UNIDENTIFIED MALE: [inaudible]

BRIAN REID: Alright. He said – does anybody here have the ability to make this thing move to the next slide?

So more and more start to get added. This is actually pretty complex. Can you – yeah, stop right there.

I was watching this go by. I thought I knew the names of all of them – thank you for changing the batteries – but it was getting out of control. So at some point in there, smart people decided that they had to bring a little bit of order to what was supposed to be – oh, I guess I should point out that the red one down there at the bottom was actually outside the United States. NORDUnet is probably in Norway, certainly in Scandinavia.

Change. Change. There we go.

When you send a UDP question, you get back an answer in a datagram. Datagrams are 512 bytes. Going along with the DNS

protocol, that really limited you to 13 answers, so there was a limit of 13 root servers.

People began to bring order to it. Vixie and Manning and Kosters pulled together this plan to give them names: A-Root, B-Root, C-Root, D-Root. Until then, they were just Joe's Root and Mary's Root and Bill's Root and Sam's Root and BRL's Root and so on. It was exceedingly informal.

IANA approved the plan. I think that, at that time, IANA was just one person or maybe two persons. I don't remember exactly what year it was that John died, but until Jon Postel died, he and IANA were the same thing. One of the challenges of ICANN was to try to replace the Postel function with the IANA function.

Any remember what year he died? Was it before or after 1995? So when it says IANA approved the plan, that means John said, "Yeah. It's okay."

I'm going to ignore this last bullet point because it's not really important. Speaking of ignore – ugh.

So these various machines in these various places were given A, B, C, D, E, F. The first few versions of these slides that I reviewed didn't have our root, F, on it. I just kept quiet about it because it didn't really matter because if one root server fails, you've got a bunch more and no one ever even notices.

But still, there's a disturbing lack of them outside of the United States. But NORDUnet is Netnod – okay. Change. Oh. It's says here, all formal, "Jon Postel used a set of criteria to select new root server operators."

Raise your hand in here if you've ever actually met and talked to Jon Postel. He didn't use a set of criteria. He just did what he felt was right. He was a really wise person, and he was a total anarchist. He also knew something about NASA, and he knew something about reliability. He wanted to make sure that the root system that he was helping put together was resistant to all sorts of attacks, including political attacks.

So one of his things was that he didn't want a root server organization. He wanted 13 organizations that were willing to talk to each other.

Wes here runs another one of the roots and when he and I first met, we didn't really have much to say to each other. We were glaring at each other because, although we're in the same business, we're not friends and we probably shouldn't be because if he falls for some stupid [DDoS] trick and his root goes down, then I didn't fall for it and my root my not have gone down.

So keeping the root server organizations spiritually, administratively, and politically separate from one another and

not having a single point of political failure was a really big piece of Jon Postel's plan for all of this. He never really wrote it down, but he certainly did it and talked about it to everybody who was around him.

But the root server operators that he chose also had to be able to serve the root in addition from being different from each other. So they had to have access to fast networks. They had to have access to people who would get out of bed at 3:00 in the morning to fix it. And they had to be people that the Internet community got a long with.

What'd you say? Yeah. RIPE I think is the Reseaux IP Europeens. It's the European Internet club. WIDE is a Japanese consortium of universities and smart people. As part of the founding of ICANN, the L-server was transferred to ICANN. So there was a little bit of settling out of the name servers as actual names and rules and policies got – oh, come on. I'm 18 inches – there we go. It just takes time.

So the root server operators all met in person and jointly agreed on principles. It's really nice when that particular ragtag group of people can agree on anything because each of the root server operators feels as though they are personally responsible for saving the world. I've often thought of them as being like The Avengers of comic book fame. They get along because they have

to all fight the bad guys, but they're certainly not going to be dear friends, and they're certainly not going to have tea together. But they will operate for the common good of Internet reliability. They all agreed to get their roots from the same place.

I know a goodly number of people, included myself, who have experimented with alternative non-standard fake roots, just to see what would happen. In case you care, it turns out that, as long as there are transparent proxies in the world, then you can't use alternate roots.

Also, you can't run a root server unless you have the horsepower to do it. You have to have the people, the bandwidth, the computers, the building, the air conditioning, and the beeper to wake you up when it breaks. Also, you have to be willing to cooperate with a plan that will move you out and move somebody else in. You actually also have to be nice to your fellow root operators.

UNIDENTIFIED MALE: I is there, so it's your turn again.

UNIDENTIFIED MALE: We do have a question on the floor here and we have a question on the web, so we'll take the floor and then I'll pass it to [Kathy] to ask for the web.

UNIDENTIFIED MALE: Could you go back to Slide #7?

BRIAN REID: I don't know. Maybe he can. I can't, but he can.

UNIDENTIFIED MALE: 7. Yeah, this one. [I expect that, in the caching resolver, if the IP of, for example, the [com] cached inside the resolver, is 1-2-3-4, I assume that if this IP is changed at example.org the DNS server]. So if the user asks the resolver for www.example.com, the resolver will go back again to the client with the wrong IP. So how should be a resolver be updated instantaneously with the exact server of www.example.org?

BRIAN REID: One of the things that you have to do when you're caching is know when to quit. Every piece of information in the entire DNS that is cached has an expiration time associated with it, a declaration of "you're not allowed to keep it any longer than this."

People who maintain DNS records are responsible for putting time-to-live fields in them. If I own the domain example.com, I can choose myself whether it's a one-hour or a one-minute or a

one-second timeout, and that controls how long the information is cached.

If you change an IP address, you're going to get bad data until all of the old data times out because it's in the cache and there's nothing you can do about it. That's what caches are. If you had to stop and look to see if what was in cache was still valid every time you used it, there'd be no point in caching.

So, yes, if you change an IP address, you will get wrong answers for a while. This is one of the reasons why people don't change IP addresses very often.

I find, when I'm running a domain, that I usually have a couple of days' notice of when something might change. I'll go in and put a five-minute TTL on those records so that, when it does come time to make the change, it'll be wrong for just five minutes instead of a day or whatever.

The ability to specify time-to-live on individual records is a big part of DNS operation, although it has nothing to do with root servers.

UNIDENTIFIED MALE: Yeah, last question. Could you elaborate more about the difference between a maintainer and an operator of the root zones, the root servers?

BRIAN REID: For every zone, including the root, the maintainer is the person who creates the entries. When I'm maintaining my own personal zone, I sit down at a text editor and I type in a name and an IP address and a TTL and then whatever else, depending on what kind of record it is. Then I get up out of that chair and get back in the same chair, wearing my DNS operator hat. I tell my name server, which is not serving the root but serving my personal domain, "Hey, you. I just changed this zone file. Please go read another copy." So I both am the maintainer and also the distributor because it's my personal zone.

The root zone is more important than that. It's nobody's personal zone, so there is a group of people who are responsible for maintaining the content of the root zone. They are, interestingly enough, called the root zone maintainers.

Then there is another set of people, which at the amount I think is just one person, who takes the zone as maintained and gets them put in the format of zone files and computes all the DNSSEC signatures and gets it ready for distribution. Then that cooked-in, ready-to-eat root zone file is distributed to all the places where the root servers will look for it, and they will put it, when they're ready, from that place. But the root zone

maintainer is quite separate from the root zone distribution system.

UNIDENTIFIED MALE: [Kathy], we had a question online?

[KATHY]: We have a question from [Afifa Abbas] from Dhaka, Bangladesh. "In the diagram you showed in the beginning slides, which one is the authoritative server, and which one is the recursive server? Until what time can a caching server store the information?"

BRIAN REID: Let's go back to it. Is that the picture you're talking about? In that picture right there, the brown servers are all authoritative, and the blue server is caching. The authoritative servers might or might not cache inside themselves. That's an implementation decision of the software inside the server. But in terms of caching in the DNS protocol, the blue caching server shown on that slide is where it's caching.

Is that the whole question? Did I leave a piece out?

[KATHY]: The second question was, "For what time can the caching server store the information?"

BRIAN REID: You can cache data as long as the creator of the data allows you to do so. Every piece of data comes with an attached Time-To-Live (TTL) value. The protocol allows you to keep that data in the cache exactly as long as the TTL value permits and not longer.

Ten years ago, there was a major American computer manufacturer whose engineers chose to ignore the TTL value. They kept stuff in their cache until you rebooted, and this caused a lot of strange things to happen.

The right thing to do, the standards-compliant thing to do, is to cache data until the time-to-live expires and then to pretend that you don't have it anymore.

UNIDENTIFIED MALE: Any other questions for now? Oh, we do have one more questions, and then we'll move onto the next session.

UNIDENTIFIED MALE: My name is [Abeeshake]. I'm sorry. Might be a dumb question, but who decides where the root server is stationed in which country?

BRIAN REID: There are 12 organizations that operate root servers. I work for one of them, and I can only really speak for myself. I can tell you that F-Root is very happy to work with pretty much anybody to put an F-Root server in your country in a place of your choosing.

We have a list of requirements, such as it has to have heating and cooling and security and electricity and not very many rats and that sort of thing.

We operate the root servers, but we encourage other people to host the root servers. What it means to host one is that you put it in the building of your choice with network connectivity of your choice and work with us to make sure that it obeys all the root service protocols. We'll talk more about what that's all about in Section 4 of this, but we've got to get through Section 3 first.

That's not at all a dumb question. It's a very critical question, and I'm glad you asked it. But let's keep moving.

DANIEL MIGAULT: Thank you. Mostly I think we're going to repeat most of the things Brian already said. It's how the root server system is today and some of the features.

As you can see here, you have 13 root servers, and you have here the IP addresses and the organizations on the left – or the other left. It depends.

Even though you have 13 organizations, 12 operators but 13 root servers, it doesn't mean you have only 13 computers, 13 nodes running. In fact, each organization has got multiple nodes. At the end of the day, it happens that over 600 instances of root servers are running all around the world.

This means that, when someone in Australia is sending the same request as someone in Alaska, they're probably not going to reach the same node. They're going to reach the service instance that is closer to them.

This diagram shows how the system is being provisioned. This is a diagram prior to the IANA stewardship transition. The IANA is responsible for providing the data to Verisign. Verisign is responsible for building the zone according to the data provided by IANA. Then NTIA was validating that the data was appropriate and that the zone was appropriate.

Once a zone file is being built, there is a notification to the root servers that the zone is available. The distribution system provides the zone to all these root servers.

What is important to say is that there is no one letter that has priority access to the root zone. They're just provisioned the same way. Each of them is responsible to provision the zone within all the instances it owns. [That's an operation issue, not an issue of operations.]

So that was before the transition. After the transition? Well, there is no NTIA anymore in this scheme. The root zone maintainer is responsible for aggregating the data received by the IANA to build the zone file, to sign it, and to make the zone file available for the root servers. Then you have the same distribution system providing the zone file to each root server, which are then internally responsible for distributing the zone file into the different instances they own.

Just to make it clear, it's two different things about building the zone and distributing the zone/serving the zone. Anything related to the zone is handled by RZERC, and anything about the root system distribution is handled by RSSAC of ICANN.

As Brian said earlier, all these root server operators are very diverse. They have different histories, different operations. They use different hardware. There is no one root server operator supervising the others, but they have some common practice. They have some physical system security. They're providing much more capacities than needed and have some experience and professional staffs and a lot of operations and so on.

It's not that they are different, that are ignoring each other, because they're serving the same goal, which is providing the roots' service. They cooperate through industry meetings. It includes ICANN, IETF, RIPE, and NANOG, etc., to talk to each

other, eventually. They have phone bridges, mailing lists. It's not that they're working on their own.

They're also involved in Internet standards that define, for example, the DNS protocols or at ICANN or when they share data and analysis on the root system at DNS-OARC.

So there is some cooperation. They work really together for one common goal, which is serving the zone.

As DNS evolves, they have to always take care of the impact of all these evolutions on the root service. For example, when IPv6 was introduced, how was it going to affect the root service with DNSSEC, and similarly with all the new gTLDs, the IDNs, and so on?

This is quite an important slide now because there are a lot of myths running about serving the root zone. We would like to clarify that a little bit.

The first thing is: serving the root zone does not mean you have any impact on the Internet traffic. If you remember, when you're asking for the IP address of www.example.org, the root zone is only providing the information about who is responsible for .org. He has no idea after what you're doing, and he has no idea which path is going to take your request to www.example.org, the [http] queries. That's the [rooting] system. It's not DNS. DNS

is only providing the IP address. Then you're going to use this IP address and it's up to the [rooting] system to send this packet.

The other thing we saw is that most of the queries are not going to the root servers. In fact, most of the information provided by the root is handled no longer than one day. It basically means that, any time you have a TLD (Top-Level Domain) you've never received, you will ask the root servers. Otherwise, it's in cache. So very little traffic is actually reaching the DNS root server.

Another point that we would like to make very clear is that serving the zone has nothing to do with designing the root zone. It's completely different things. When you have a newspaper, it's not the one that delivers the newspaper – the mailman – he has nothing to do with what is written in the newspaper. Well, it's kind of the same thing for the root zone servers.

Letters are equal. There is no one letter that got information prior to the others. Root servers operators are not hobbyists. It means they're doing the job seriously. They're over-provisioning. They're monitoring. They're interacting with many bodies in the Internet community. So it's something they're doing seriously.

There are 13 letters. It doesn't mean there are 13 instances. There are 600, about. These root operators coordinate because one goal is to serve the zone.

Oh – and now that's Anycast.

UNIDENTIFIED MALE: Any questions? No questions? Okay – oh, one question from [GZ].

[GZ KABIR]: I'd just like to know, if the letters don't differ from each other, why should we host more than one root server in a particular position? [For a server I know, my IX] was K-Root server, L-Root server, F-Root server, and one instance from Verisign as well. So why are there a number of different root servers in a particular region?

One more question: why 13?

DANIEL MIGAULT: Sorry?

UNIDENTIFIED MALE: Why 13? Why not 12? Why not 14? Why not 10?

DANIEL MIGAULT: For the first question, I would say it's more for redundancy. You have different organizations, so if one fails, then you have another one.

So why 13? I think it's a good question. Do you want to answer?

BRIAN REID: [inaudible]

DANIEL MIGAULT: Oh. Maybe Lars wants to answer.

UNIDENTIFIED MALE: I have a mic here. I can pass it to Liman.

DANIEL MIGAULT: The number 13 comes from the size of the UDP packet.

LARS-JOHAN LIMAN: My name is Lars Liman. I'm in charge of running I-Root, the one operated by Netnod in Sweden, not Norway. The number 13 is very old. It was part of the explanation that Brian gave, the historic part where the number 13 was as many as we could fit into the first packet that the DNS client wants to see.

When this recursive server, the caching resolver, starts up, it needs to have the list of all the root servers. We wanted to fit that list into one single packet for efficiency. If you look at the DNS standard, which is very old by now, it actually limits the size of one such UPD DNS packet to 512 bytes, 512 characters. So

that's what we have to play with to fit the list of all servers and all their IP addresses. And that's really tight.

By changing the name of the servers so that they end in rootservers.net, we were able to go from 9, which was the previous limit, to 13. We could squeeze four more in by just changing the name because we took advantage of the internal structure of the packet – these were real, real, deep-down DNS experts who sat down and thought, "Hmm. If we change the name, the algorithm that puts together the packet will actually be more efficient, and we could fit four more in."

Since then, the DNS protocol has developed. But there's also the question of: how many do we need? Back in the old days when this happened, when we put 13 in there, there were 13 instances. There were 13 machines on the Internet – that's not many – that provided this service. Today we have 600. Today the letters represent organizations operating parts of these 600 instances of servers.

So it's quite possible that, today, we don't need 13 after all. It could service with 5. I'm just saying numbers because we don't know. RSSAC is looking at trying to investigate and find what the right number is – if there is a right number – and what would be the changes if we go upwards, if we go downwards. It's much

more important that you have close access to one or two or maybe three letters than to have more letters.

There's a mapping between letters and the organizations that operates them, so what you're asking with more letters today is: do we need more organizations that operate root name servers? And that's not quite obvious.

What we do need to provide better service to the Internet is to make sure that we deploy more instances of the existing letters in more places and that every user on the Internet has good access to a few of these letters. It doesn't matter which because they're all alike. That way, we can meet the operation requirements from the Internet at large and use the system that we have today.

The number 600 is not really limited, so we can expand on that and we can put many, many more servers out there. But we can do that from the existing set of server identities that we have.

UNIDENTIFIED MALE: Thank you, Liman. Actually, I have two more questions, if we have time. We've got 30 minutes. Three more questions.

I'm going to take two and then we'll keep going, and then we'll end with one more Q&A session.

UNIDENTIFIED MALE: Could you go back to Slide 26, please? It's about the IANA transition. Other than the government seizing control and NTIA going out of the picture – there's no oversight of government in it – what else has changed here? Is there anything else which has changed, other than the –

UNIDENTIFIED MALE: No. Lars says yes.

[BRIAN REID]: I guess some of the people are involved are different.

DANIEL MIGAULT: Could you back up one more slide? Yes. In this slide, we have the blue square, NTIA, which is involved with the oversight of the root zone provisioning. If you go forward one slide again, please. In this slide, NTIA disappears, but you also will see the term RZERC at the top. RZERC is a new body that has been defined in the context of the IANA transition as the IANA stewardship of the…that's too long-term. There's the IANA stewardship transition.

So the RZERC is a body which is comprised by representatives from many corners of ICANN that is tasked with overseeing the

technical evolution of the root zones. So this is the Root Zone Evolution Review Committee – thank you. Brad is a member of the committee. That has, I wouldn't say, entirely replaced, but it's filling some of the functions that the NTIA had in the past. But it is now a multi-stakeholder body instead of a government representation.

BRIAN REID: Everything still happens the same way. A couple of people's job titles have changed. But pretty much, the process remains the same.

We have time for one more question. Section 4 of this is going to be what most of you actually want to hear, so let's get to it. One more question over here.

UNIDENTIFIED MALE: Thank you. I'm [honored]. I'm asking you about the new gTLDs and the root zone system. [I knew that the new gTLDs, .home and .[cop]], has been suspended for the reason of namespace collision. Right?

BRIAN REID: That's completely outside the scope of this meeting and this group.

UNIDENTIFIED MALE: No, no, no. I wonder if the basic system, the rooting system, can handle this problem.

BRIAN REID: I don't see that there's any issue here. This is an ICANN issue and not an RSSAC issue. If it's in the root zone, we'll serve it.

UNIDENTIFIED MALE: Yes. Two letters had – due to the reason of a namespace collision in the [interprocess – already used]. So I wonder if the technology of the root zone system can handle this problem. I'm looking forward for when the two letters can be introduced to new gTLDs an opened for auction. Thank you.

DANIEL MIGAULT: I will pass the mic to Warren, but the thing is, it's about defining the root zone – what you're asking – and this is out of scope of RSSAC.

WARREN KUMARI: Hi. Warren Kumari. I help out with F-Root. Yeah, that's information that goes in the zone file, and RSSAC has no real opinion on the stuff that goes in the root zone. That's entirely up to ICANN. So it's an ICANN issue.

The root servers simply serve the information in the zone file that's given to us by ICANN [inaudible].

BRIAN REID: I've heard a bunch of question and I've seen a bunch of faces and I've been doing this for a while. I think that a lot of people have in their minds this question: "Why do you have a root server and I don't?"

The answer to that is really, "Because you haven't got one yet." Lots of countries have national airlines. Very few countries manufacturer airplanes. So if you have the national airline of Shangri-La, you are probably not going to start an airplane manufacturing facility in order to set it up. You're going to buy aircraft from one of the half-dozen or so countries that have a well-established aircraft manufacturing facility.

Just like there's no real link, no important link, between manufacturing airplanes and running international airlines, there is no important link between being a root server operator and hosting a root server in your place or country of business.

What I want to do in this section, using what little time we have left – the questions have eaten into it – is talk about Anycast. I think I'm going to skip all the written slides. They're on the

website if you want them. I'll be brief because this is really pretty simple.

I want to start by saying that, in the beginning, there were 13 actual physical root servers, and the organization did what they did. Then one of the root letters had the idea, "Hey, we could Anycast. We're going to do this." The other root server operators basically said, "You're crazy. You're nuts. It won't work. The world will end if you do this. Don't do it." They did it anyhow, and it worked fine. This was a good example of why having root operators not be beholden to one another is a good thing.

If you have an IP address and you send it out of your computer, it will go wherever the network routes it to. That's what networks do. You buy routers from router manufacturing companies and you configure them to know where the U.K. is and to know where British Honduras is and what have you.

If people who run networks get crafty and put in little blips so that the same address can appear in more than one place, then if you send a packet, it will reach the first one of those that it comes to. That's all Anycast is. It's a cheap routing trick designed to get packets to go to one of many different versions of the same IP address, rather than to a specific computer.

There's a really fancy PowerPoint diagram here that allegedly says it, but it's just not that complicated. If the people who

configured the network can make it so that you can have 19 different copies of this address and you are somewhere in the network, yours will go to the one that is closest.

So what Anycasting a root server does is it takes a single root server IP address – whatever that might be – and makes many, many, many copies of it. I know one root letter that has 120 different Anycast instances with the same IP address. I also know of a root server that has exactly one. It's up to the root server operator how many they make.

But hosting a root server an operating a root server are very different things. Anybody can host a root server. All you need is a building that will make it first and network people who are crafty enough that they can make the network surrounding it send packets to that version of it instead of somebody else.

Daniel in one of the earlier segments mentioned that Anycast is used to guard against DDoS. The way that works is, if you have 100 different machines that have the same IP address and some bad actor is sending too many packets to that address, it's not going to all of them. It's just going to go to one of them. That means that the others will not be swamped.

But if the denial of service attack is coming from lots of different place and that it might impact first this one and then that one, Anycast guards against denial of service attacks by providing

multiple targets with the same name. It also provides multiple servers with the same IP address, which means that it can be closer to you.

In answer to a question earlier about why you want more than a couple of root servers, root servers are done by humans using software written by humans that can have bugs in it. I would encourage anybody who believes that it's a matter of life and death to have at least three different pieces of software and hardware manufactured by three different companies and put them in different buildings so that a single fire won't wreck all of it.

Diversity is good – diversity of software, physical location, and all that stuff. It starts to get excessive if you have more than about four root servers in the same metropolitan area. But certainly there's a lot of arguments to be made in favor four.

I think that that's all there is to say about Anycast. It's just not complicated. It's very simple. It's not done by servers. It's done by network operators, and they do it for the purpose of making room for multiple copies of the same server.

If you really want your own root server and you don't understand why I have and you don't, then call me or call any other root letter operator who's doing Anycast installations.

We'll make it work. It's just really very simple. There's nothing to it.

For the same reason that you don't want to start your own airplane manufacturing plant in order to make an airline, you don't want to develop your own root server operator system in order to have a root server. It's just ridiculous. Too much overhead.

Any questions? Yeah? Somebody with a microphone?

UNIDENTIFIED MALE: Yeah. Paul, I'll get to you. I do have a question that we put on pause after, so I'll get to you next. How's that?

Sir, did you still have a question? Okay.

UNIDENTIFIED MALE: Two questions. One question is regarding the timelines. RSSAC is evaluating the numbers two or three or four. Is there any timeline decided?

The second question is regarding the distribution. When is the contract with IANA and Verisign going to expire?

BRIAN REID: I do not understand the question.

UNIDENTIFIED MALE: Right now, Verisign is designated as [distribution] [inaudible]. In your diagram, in your graph, in the post-IANA transition, the NTIA is gone. Can you move to the slide?

BRIAN REID: Yeah, but I can't answer your question. I'm a root server operator, not a politician, and that's a contract question. I'll see if I can find the slide.

[WARREN KUMARI]: I'll speak very briefly from ICANN's perspective. The IANA is the IANA functions operator, currently under ICANN. The root zone maintainer is currently under contract with Verisign. I'm a techie. I'm not going to even try to guess what the terms of those contracts are or what the terms of the functions operations are. This is a technical forum here. That would be a question for other forums during this congress. Alright?

UNIDENTIFIED MALE: The first question, and then [inaudible]

UNIDENTIFIED MALE: The first question is on the number of root servers, or number of letters. RSSAC is evaluating whether two will be there or three will be there or 14 will be there. So [inaudible].

UNIDENTIFIED MALE: I believe, actually, that's the next section. You guys are talking about some RSSAC publications? So maybe we can get to that question in the next session. I'll go to [Paulo], and then we'll go to the RSSAC activities section. Hopefully, that'll answer your question.

[Paulo]?

[PAULO JORGE]: Hello. I actually host a mirror of the L-Root server with ICANN. My question is concerning maybe contingency measures in light of what happened to Dyn – the attack on Dyn's DNS. With the proliferation of IoT devices everywhere, there could be perhaps – this is just a hypothetical – a situation whereby maybe attackers are able to actually use those IoT advices to attack perhaps all the instances of Anycast.

Have you done a contingency assessment on the possibility that perhaps IoT advices can be used to actually bring down whole sections of the root zones?

BRIAN REID: We spent a lot of time talking about this. I'm not about to tell you how to do it. It is a real problem. It has no obvious solution and lots of unobvious solutions that clever people are working on.

But, yes, the growing problem of distributed denial of service attacks is real and it's getting worse. There's a lot of discussion among the experts of the Internet as to what ought to be done about it and what can be done about it.

I will point out that the attack on Dyn was not an attack on the root. It was an attack on second-level domains because that's who Dyn's customers mostly are. It had the same effect that an attack on the root would have had.

So the Internet is vulnerable, and the good guys try to stay ahead of the bad guys, but the race never stops.

DANIEL MIGAULT: Now let's see a little bit more about what RSSAC is and how we work and what our different activities are.

RSSAC stands for the Root Server System Advisory Committee, and the mission is to advise the ICANN community and the

Board on matters relating to the operation, the administration, the security, and the integrity of the Internet root server system.

Did I miss a slide?

Okay. RSSAC provides advisory to the Board. Here's where RSSAC is. There's a link to the Board. And we have a liaison, one person coming from RSSAC attending the Board meetings.

RSSAC is composed of actually two main bodies. One body is composed of the root server operators. For each letter, you have two persons representing the letters and some liaisons. You have incoming liaisons of RSSAC; that is, people coming from external bodies attending RSSAC meetings.

The other things RSSAC is composed of is a huge set of experts that are coming from the entire community. It's a technical group of experts that are supporting the questions that the advisory RSSAC is providing.

The way it works: there are some work parties that are organized, where the people meet, and end up with documents/recommendations that are then sent through RSSAC to the different communities.

The Chairs of RSSAC are Brad and Tripti. Here they are. The different liaisons RSSAC is involved with are the people coming, attending RSSAC, from other bodies. You have the IANA function

operators, the root zone maintainers, the Internet Architecture Board, and the Security and Stability Advisory Committee. So that's the people coming to RSSAC.

RSSAC is also sending some people to other bodies, such as the ICANN Board, the ICANN Nominating Committee, the Customer Standing Committee, and the Root Zone Evolution Review Committee.

So all the liaisons are here to make the links between RSSAC and the whole community.

The Caucus is, as I mentioned, the technical community for RSSAC. It's very open. That's what I want to say. If you want to apply, you just have to mention your interest in applying and provide some explanation as to why you're applying and whether you have some knowledge on the DNS. But that's basically all that's requested.

Most of the members of the Caucus are not root operators. They're just coming from every part of the community. The idea is to have some diversity and that the advisory provided to the community is made in a transparent way and considering all the spectrums.

If you want to apply, send an e-mail to this e-mail address. You're more than welcome to apply.

What about the work RSSAC is doing? We had recently a workshop. It's the second workshop. The workshop was focused on architecture, evolution, and [reinventing] RSSAC. We produced a few statements, as well as we worked on some publications.

One of the statements was the Client-Side Reliability of the Root DNS Data. In this statement, we intended to clarify that RSSAC has no interactions with the data of the root zone. It's only serving the [root] zone. Whatever instance your request is reaching, the same answer is going to be provided just by the root system.

There is absolutely no interest in changing the root zone or modifying that because, as a resolver, you're able to check that the packet has been changed or not with DNSSEC.

The other statement is about if you have one root server that in unavailable. We evaluated the impact on the end users. The impact is that there is no impact. You don't even notice that.

Another response was on the GNSO PDP on the new gTLDs. Basically, in one sentence, it says that adding new gTLDs does not have much impact on the root service. If something is being noticed, RSSAC is going to notify the Board. That's basically what [ends up happening].

So that's for the statements. There are also ongoing works. One is about the history of the root server system. This document is going to be published very soon. I would really encourage people reading this document to understand why there's 13 letters and how we came to the [designations] and so on. That's a really, really interesting document.

There is a root server naming scheme document. This document is basically considering that the zone file hosting the names of the root servers should be signed. If signed, should we still have the same designations for the root servers, which means a.rootservers.net, b.rootservers.net. Can we have different names? Is there any better way to designate these root servers? This document should be published soon also.

Another document was on the key technical elements of potential root operators. It lists what the requirements are and what is expected from an organization if it wants to become a root server operator.

Another document – well, it's more of a work party that is just being formed in the Caucus – is about distribution of the Anycast instances. This work party is just being started. The work is being done in the Caucus.

Here you have a description of the different technical elements.

So in a word, you have everything on the main webpage. You can also reach the different documents RSSAC is producing. You also have a special page for the Caucus, and feel free to apply.

I don't know if there are any questions.

UNIDENTIFIED MALE: We have about ten minutes left. If there's any questions, this is the time to ask them. We've hit [inaudible]. [Paulo's] got another question.

[PAULO JORGE]: My question is on the Caucus. Is there a limit to the number of members you can have?

[BRIAN REID]: 13.

DANIEL MIGAULT: No. No limit. The sky is the limit.

UNIDENTIFIED MALE: Any other questions? Thank you.

UNIDENTIFIED MALE: I don't know if my question is valid or not. Why don't we introduce a concept of every country having their own registry operators? Some countries who are not capable ask Afilias and Verisign to run their technical operations. Can we do the same thing with the root servers? So every country's operators can install root servers in universities so that students can study more deep research. Then the objective Caucus can be achieved in that way, I think. So more students and more researchers will be involved.

DANIEL MIGAULT: Yeah, of course. There is no constraint on that to apply to the Caucus, but for research? I think, even though it can be handled in a Caucus, I think DNS-OARC is a more appropriate place. But it does not prevent you from applying or the students from applying to the Caucus and DNS-OARC.

BRIAN REID: I might get fired for saying this, but you don't need anyone's permission to install a root server. You can go buy a computer, put the right software on it, tell it that it's serving the root zone, and then put root hints files in your client saying, "Hey, that guy over there is the root server," and guess what? It just became the root server.

It's your problem getting root data into it. It's your problem keeping the root data up to date. But there is nothing magic about a root server.

All this stuff we've been talking about here is a process that's designed to make sure that the right information is served by servers that work most of the time and are run by professional people, etc. But I could put a root server in my basement that lets my television set find the root if I wanted to. There's just no point in it.

LARS-JOHAN LIMAN: Since I got flack for saying the following thing yesterday, I might as well continue today. The thing that is inappropriate to do is to put a public root server using someone else's IP address; for instance, one of the existing root name server addresses.

To run a private root name server in your basement, in your lab at the university, which is disconnected from the Internet? Go ahead. I do it when I teach courses on DNS. In some cases, you have to do it to make things work.

On the public Internet, where everyone else can see it, at least don't do it using the existing IP addresses for the current root name servers.

BRIAN REID: On the other hand, if you were to learn the IP address of, say, I-Root, and you were using a NAT box in your house, and the inside of your network was using an RFC 1918 address, you could actually, if you wanted to, run a name server process using the address of I-Root, and it would be I-Root.

The chances are it wouldn't work as well as the real I-Root, and you probably shouldn't do it. But nothing is going to stop you, and no police are going to come knocking at your door if you do that.

UNIDENTIFIED MALE: [Kathy], we have a question online?

[KATHY]: I have a question from [Afifa Abbas] from Bangladesh. "We know the DNS works on the UPD 53 Port. Can you explain when DNS works on the TCP 53 Port?

BRIAN REID: When DNS was first invented, there wasn't very much information that needed to be send, so UDPs were fine. When DNSSEC came along, all of a sudden, the response was usually bigger than would fit into a single packet.

Rather than invent yet another scheme for using multiple packets to convey information, the protocol designers decided that the right thing to do here was to use an existing protocol, namely TCP, as a way of sending more information than would fit into a given packet.

The specification for how to do that is called EDNS0. It's simply way of encoding more than 512 bytes of information in a response, and it was put there to make DNSSEC feasible.

There's a big mess having to do with lack of standards in the implementation of EDNS0, but we're not going to go there.

UNIDENTIFIED MALE: [Shiva], I'm trying to balance the people who are asking questions, so I will get back to you.

UNIDENTIFIED MALE: Mirror root servers are local and global. Can the local prevent more DDoS attacks? Or the can the global better for DDoS attacks? So which instance is best – either local or global – to prevent DDoS attacks?

BRIAN REID: The difference between a local and a global server has entirely to do with BGP routing. What it means to have a global server is

that you advertise a BGP route that will propagate everywhere. What it means to have a local server is that you advertise a BGP route that is supposed to not go very far and you're telling your neighbors to please not send it any further. It is not a property of the server. It is a property of the BGP fabric in which the server is placed.

The question you asked is, "How does a DDoS attack figure out what server to attack?" The answer is, "That's up to the routers." A DDoSer will send a packet that is aimed at a certain address. Exactly where that packet ends up is not a property of the server. It's a property of the routers that lead to the server.

The ways of keeping global and local are too BGP-technical to waste time on right now.

UNIDENTIFIED MALE: I don't see any other hands, so we'll give the second-to-last question to [Shiva], and then we'll wrap it up.

[SHIVA]: As you explained, operating a root server is not a very big deal. Anyone can create to study at the university on how it operates. Right? You just have to install that.

I'm not against the current [process], but there's a lot of confusion. Even when I was studying engineering, I had no idea what a root server was. So nobody really wants to [study in this field] in our country.

But there's a huge amount of technical people in India, so can we allow countries to install root servers, by signing a proper contract with, like, Verisign as a maintainer and a distributor? I know, I'm not against it, but is there any possibility of doing such things like that in the current system?

BRIAN REID: Any possibility of what?

[SHIVA]: So that any country can become a root server operator. Because even if we install instances in different countries – I guess I'm not much clear. Who actually operates that mirror in that particular country?

BRIAN REID: It's not a mirror. It's an instance, and that's a very important distinction. A mirror is a copy of something else. An instance is a full-fledged entity in its own right that does not differ in any way from any other instance.

What? Yeah, but I really wish that, instead of being called root servers, they had originally be called coleslaw servers or something because, if it had a less glamorous name, people wouldn't want one so badly, without realizing that it doesn't do you any good at all to have. The difference between a root server operator and a root server host is completely irrelevant to good Internet access. People seem to want to be root server operators.

Given that I am one, I can't imagine why anybody actually wants to do it. I understand why everybody should have an instance of some root servers in their regional and national networks. That's obvious, but deciding which organization is going to get it and do all the maintenance work and the political work and flying-to-Hyderabad work – it took me 33 hours of traveling to get from my home to this meeting, and I did it because I'm a root server operators and that's what we're supposed to do.

F-Root has about 56 instances, and none of the hosts of those instances needed to get out of bed and come to Hyderabad. It's a different kind of thing. But the instance root servers are 100% the same as the one that's in my building. They don't do anything that's different. They don't have any access to any data that's different. They're just another one.

BRAD VERD: Brad Verd, Co-Chair of RSSAC. Just to try to address your question, stuff was said earlier about how it's easy to set up a copy of the root zone and serve the root. It is by no means easy to be a root server operator. We have 600 instances worldwide, running a platform in sync, where we provide cryptographically-signed answers across the world to the Internet as a whole.

So I want to make sure that we don't conflate the two things of being one of the 13 letters' root server operators and somebody running a copy of the root zone in a lab somewhere. Entirely different.

BRIAN REID: Just because it's easy doesn't mean you should do it. It's a bad thing.

UNIDENTIFIED MALE: I know we're at time, but we do have one more question and I want to maximize the opportunity for people in the congress here in India to ask these questions. So if it's okay – I know we're at time – maybe one or two minutes more? Is that okay?

Okay. Thank you.

ABDALMONEM GALILA: Last question. I am Abdalmonem Galila. I agree that UDP is limited to 512. TCP is limited to 4,096. So if I have my zone and DNSSEC signed my zone with two keys, maybe there is one size of [dig]. Or if my query exceeds the size of [METU], of network element – so the response would be dropped by [the firewall]. Is there any solution for that?

BRIAN REID: I'm not completely sure I understood the question. TCP does not have a limit of 4,096. TCP is a stream and you can send as much as you want over it. The TCP protocol has some fairly primitive and a not-all-that-strong way of checking for errors and turning errors into delays and that kind of thing.

If a name server, root or otherwise, is answering with TCP, that consumes a lot more resources on the server because the server has to maintain state about each about each TCP connection and keep that state active until the data is finished being transmitted. So, realistically, the capacity of a server, root or otherwise, is dramatically reduced if everybody is using TCP.

There is one country in which there is an F-Root server where 100% of the requests are TCP. That server is being hammered all the time. But most of the servers that F-Root runs are in the 5% TCP range.

We would have to get bigger computers and bigger networks if everybody used TCP all the time. Yes, the Internet is big and fast, but one of the reasons it's big and fast is that experts work very hard not to send any more data that you have to.

DNSSEC, with a larger key, is going to take more data. DNSSEC with two keys existing at the same time is going to have more data. That's just the way it is, and that's why caching is good.

ABDALMONEM GALILA: So TCP doesn't help if the packet size exceeds the maximum [transmission] unit of any network element? So this packet will –

BRIAN REID: TCP is capable of succeeding in the presence of fragmentation. It's not a good idea, but the Jacobson algorithm knows how to slow start and get the thing working and get past some really pathological cases in fragmentation. It does work pretty well most of the time, amazingly enough. Fragmentation was a bad idea, but it does work.

UNIDENTIFIED MALE: Alright. I appreciate everyone here. I hope you guys a lot from this presentation. Like I said, they don't disappoint when RSSAC comes. It's always worth coming to watch and listen to.

Brian, thank you very much. Daniel, thank you very much. Please give them a hand. And thank you all for coming.

[END OF TRANSCRIPTION]