SAN JUAN – Fellowship Daily Session
Monday, March 12, 2018 – 12:00 to 13:30 AST
ICANN61 | San Juan, Puerto Rico

UNIDENTIFIED MALE: Good morning. ICANN 61, March 12, Fellowship Daily Session.

SIRANUSH VARDANYAN: Today, we have a special session which our tech gurus agreed to run for you. I have a great company here, and I would like to give the floor to Rachel first and then you guys to introduce yourself for Fellows to know who is doing presentations.

Please take your lunch, come and take a seat and be attentive. This is really a very interesting session, which I'm sure you'd like. Rachel?

RACHEL REYES: Hi. Good afternoon, everyone. Welcome to the DNS Fundamentals session. This is a one and a half session. We'll take 15-30 minutes for your question and answer at the end of the presentation.

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

I'm Rachel Reyes. I'm a technical support for ICANN org. John Crain here who is sitting on my right will help me with the question and answer session later.

JOHN CRAIN: I'm John Crain. I am ICANN's Chief Security, Stability, and Resiliency Officer. I've also been involved in operating root servers and other DNS services over the last 20 or 30 years.

Hiding in the corner over there on the telephone is Mr. Matt Larson, also known as Mr. DNS. He's also been in the DNS industry probably for longer than I have.

With that, I'll pass it back to Rachel.

RACHEL REYES: Okay, so I hope I will get your attention while you're eating your lunch. So let's begin.

IP addresses are easy for a machine to use, but it's also true it's easy for us to remember names but it's very difficult for us to remember numbers.

Let's take, for example, some of us – at least for myself – it's very difficult to remember my family's phone numbers or even my friends, but it's easy to remember them by name, which is the same thing for remembering names and numbers in the DNS system.

In the early days of the Internet, names were very simple. No domain names yet. These are single-label names with 24 characters which are referred to as host names.

The name resolution is mapping off IP address to names. In the early days of the Internet host file names HOST.TXT, this is what we are updating and this is centrally maintained by Network Information Center or what we call NIC at the Stanford Research Institute. They are manually updating the file via e-mail, and this is released once per week and downloadable by FTP. That was in the early days of the Internet.

The problem with that system was everything is edited manually, so it's very error prone and very inefficient because you have to e-mail first before they can actually update it. That also requires significant bandwidth when you're trying to upload or download the file. So that didn't scale for long.

That's why people started a conversation in the 1980s on how to replace the current system. They came out with this DNS or domain name system concept where it would scale the current HOST.TXT system issue and also simplify e-mail routing. You can find more documents on the requirements in the RFCs provided here. RFC 799 and RFC 819 will tell you more about the requirements or that discussion regarding the DNS concept.

DNS in a nutshell. Here we're going to discuss first the terminology that we are using in the DNS system. We have DNS, DNS data, resolvers, name servers, caching, and replication.

I will discuss this in detail using the graph here. We have the stub resolver as one of the terminologies that we're going to discuss. And the recursive name server, this is the one sending queries to our name servers. So name servers are the ones on my left, which is on your right. And then these name servers are the ones giving definitive answers to the queries being thrown by the recursive server. We have a very little bubble here that says "cache." Cache is being used in the DNS system to make it more efficient and scalable. We will do a deep dive into this at a later part of the presentation.

The name space is the DNS database's structure in an inverted tree. Normally, how we read a tree data structure is from top to bottom, but in our DNS world we read it from bottom to top.

Let's take this graph for example. We don't read it as ".com.example.www." But instead, we read it as "www.example.com..," which is what call a fully qualified domain name.

In the name space, the first one is the root. The second here is what we call the top-level nodes, followed by the second-level nodes, and then the third-level nodes.

Each of these nodes has a label. The labels consist or the legal characters that you can use for labels are only letters, digits, and hyphens or what we call LDH.

The maximum length that we can use for a label is 63 characters. It's not really case sensitive, so basically you can write .com in a way like "com" or "cOm." It doesn't really matter.

Every node has a domain name. In our example, we're going to use here this tree. The domain name highlighted is "www.example.com.." These are all separated by dots.

As I mentioned earlier, the fully qualified domain name or FQDN ends with a dot. Most of the time when we are searching for a domain name, we don't really use dot at the end but instead we just type "example.com" or just "www.example.com."

A domain is a node and everything below it. In our example here, .com is the top node of our domain and anything below it is the .com territory or .com domain name.

Zones are the administrative administration, and each DNS zone is based on the boundary of authority and it's being delegated to an entity. A DNS zone can have one domain or many domains or subdomains. Delegation creates zones. The delegating zone is what we call the parent and the created zone is what we call the child.

Here the parent, which is the root zone, delegated the information to the child .com, .uk, .coffee. And .com has delegated it to this child which is the .foo, .bar, and example.

Name servers, as I mentioned earlier, are the ones answering queries being thrown by the recursive servers. The name servers authoritative for a zone has a complete knowledge of that zone. Basically, when you are sending a query, if it's an empty recursive name server, it will go directly to the zone because that's where you have to go because it has the definitive to your queries. Zones have multiple authoritative servers. That is because it wants to for efficiency and redundancy use.

How do you keep a zone's data in sync across multiple authoritative servers? We do have this DNS protocol which is built into the server that does the zone replication. It happens using the primary and the secondary servers.

The primary server has the definitive zone data. If you want to make changes to a zone, it has to be on the primary server. On the other hand, the secondary server or what we call the slave server is the one that is retrieving the zone data from another authoritative server. The process is what we call zone transfer.

Zone transfer is actually the communication between a DNS server to another authoritative server.

Another server that we have to discuss here is the master server where the zone file is originated from. But bear in mind that a master server doesn't need to be your primary server. Your secondary serve can serve as your [master] server as well.

Zone transfer is initiated by your secondary server. You can find under this RFC 1996 it has the details of how this zone transfer is in process or how you make changes to the zone file.

Now we go to the DNS resource records. DNS resource records are what we call to a lot of people as RRs. If you can recall as I mentioned earlier, every node has a domain name. A domain name has different kinds of data associated with it, and this data in the domain name are actually stored in the RRs.

We have different kinds of resource records, but we are going to discuss only a few of them. Let's go to the format of resource records. Resource records have five fields: the owner, time to live or TTL, class, type, and then the RDATA. The owner is the domain name the resource record is associated with. Time to live is the time the record can be cached into your server. Class is a mechanism for extensibility that is largely unused, which most of the time we always see IN in this for class. Type is the type of data the record stores. And RDATA is the data that the record carries.

I think you are more familiar if you look into this, or I can show it to you. This information is what we call the resource records. I think most of you if you are familiar with networking, you will be able to understand what I was saying earlier.

[As mentioned here], the type and the RDATA always appear which are necessary. These are the most commonly used resource record types: A data stands for the IPv4 address. The AAAA stands for the IPv6. NS is the authoritative name server. SOA or the start of authority always appears at the zone apex. In our example earlier, you can find SOA information at the .com. CNAME or the canonical name is the alias to another domain. MX stands for mail exchange server. And PTR is the pointer or used for reverse mapping.

Like I said, there are many other resource record types out there. As of December 2017, there are 84 already. You can go to this website to find more of these resource records types. If you go to that page, this is what you will get.

Let’s go with that A and AAAA records. As I mentioned earlier, this is how the A record looks like. It will give you the IPv4 address, and then AAAA will give you the IPv6 address.

Name server (NS) specifies the authoritative name server for a zone. It appears in two places, the parent and the child, so on. In this example, the left side is the name zone and then the right side is the name server, not the IP address.

This is how the name server record marks the delegation from the parent to the child. The .com has 13 name servers. Basically, these are the 13 root zones. It appears here from the root all the way to the .com.

During delegation, we're also including a glue record. What is a glue record? A glue record is either an IPv4 or IPv6 resource record. It's included in the parent record as part of the delegation. The reason why we need to have a glue record is because if let's say you are querying for what is the IP address of www.example.com, you will go straight to the root zone. The root will give you the IP address of a name server. Then you will ask again the name server, "What is the IP address of www.example.com?" And it will go into a loop until you will not be able to find your answer because they don't have the IP address yet. That's the reason why we need to have a glue record.

Start of authority are located at the zone apex. Here this is an example of how an SOA looks like. It has your domain, your name server. Hostmaster.example.com is the administrator of the zone. Then the serial number is the current version of the file.

Refresh stands for the number of seconds the secondary name server should wait before checking for updates. Then retry stands for the number of seconds a secondary name server should wait before retrying a failed zone transfer. Then expire stands for the maximum number of seconds that a secondary name server can use data before it must either refresh or expire. And then minimum is the TTL.

CNAME or the canonical name record creates an alias from one domain name to another. On my right side, probably on your left side, is the CNAME and then the right side is the canonical name and target of the alias. Remember that a CNAME creates an alias and points to a canonical name, but please bear in mind don't overuse it. Don't create chains or loops. It doesn't look good on your data as well.

Mail exchange record type. MX specifies a mail server and a preference for a mail destination. Our example here says example.com MX 10 mail.example.com. The correspondent numbers, the 10 and 20, are how the e-mail or the prioritization will happen. The lower the number, the better. This is our preferred way of [inaudible] of mail routing.

Reverse mapping. Most of the time, we are always looking for the IP address of a domain name, but there are times that we are looking for a namespace instead of the IP address. This is where the PTR resource record is very helpful. Not all the time we are using it, but on the networking side it is always there in the data. This is what it looks like.

Let me ask John Crain why we have in-addr.arpa.

JOHN CRAIN: in-addr.arpa, as we refer to it, is the reversing of the addresses. Some protocols actually check that to make sure that the name and number map both ways.

RACHEL REYES: Okay, so these are other resource record types that we have, but I seldom see them really, except for the CDS and CDNSKEY which is part of the DNSSEC.

This is an example of a zone file for example.com. It has the SOA, the name server, the IPv6, IPv4, and he MX record and CNAME as well. Then on the last part is the glue record. So it states already the IP address, which I mentioned earlier. Without this, it would just go on a loop.

Now we go into the resolution process. As I mentioned earlier, we have the stub resolvers, recursive name servers, and authoritative name servers. These servers cooperate to look up the DNS data in the namespace.

Stub resolvers are very local to your client. It could be in your phone or it could be on your laptop. Then the recursive name servers, again, these are the ones sending queries to the authoritative name server. The [authoritative] name servers are the ones sending answers to that query.

A DNS query always comprises three parameters, namely domain name, class, and type, which is what we have here in our example.

Two kinds of queries. Stub resolvers send recursive queries, and then recursive name servers send non-recursive or iterative queries or what we call referral. We'll discuss this later on.

I think I'll skip this.

This one I have to discuss because let's say you are starting a resolution process where the recursive server is empty or it just turned on. You don't have any option here but to go straight to the root name servers because the root zone files are located in that.

How does a name server find the root name servers? They must be configured. This is being configured by the server administrator, and there's no way to discover them.

This is the list of root name servers and root hints file. NS are the name servers. A are the IPv4 IP address. The AAAA are the IPv6.

Root zone administration is very complicated, so we should not discuss that. We should keep it simple. If you want to know more about it, maybe you can get Matt Larson M&M Peanuts and he will have more time for you to discuss on that. But we are not going to talk about that here.

There are two organizations that cooperate to administer the zone contents, basically ICANN and then Verisign. There are 12 organizations operating the authoritative name servers. Probably you are wondering why 12 where in fact we have 13 root servers. That is because Verisign has two: the A server and the J server. Why do they have two? Again, John Crain and Matt Larson will have answers for you, but probably they're not going to discuss it here. Probably outside unless John has time to tell the story?

JOHN CRAIN: No. It's just a factor of history. The last time name servers were distributed in the 90s when new name servers were added, not all of them were placed in new organizations and two of them didn't have homes. One of them went over to the east coast to Verisign who had a strong relationship with Jon Postel and ISA and the other one stayed in ISA which was the letter L. When ICANN was formed, that letter came to ICANN and, of course, J stayed with Verisign. So it's just a factor of the history of how they were distributed.

RACHEL REYES: There you go. If you would like to know what are the root servers in a country, you can go to this website: root-servers.org. I can show it to you actually. Let's say you are looking for what are the available root servers in Puerto Rico. The L and J are available in Puerto Rico currently. So you can come to this website if you would like to know that information.

Also, we have this Anycast that are now using instances of these root servers that will help you to look for the closest DNS or the root servers in your location. It also helps when you are doing a search. It's more efficient if you have instances in your location.

Root zone change process. As mentioned in this slide, this is a simplified version. There are more processes behind it actually. We're not going to discuss, but we're just giving you an overview on how a root zone file is being changed.

Basically, it starts with a TLD manager submitting a change to IANA. Then IANA will implement that request through, first, updating the root zone database and then creating a root zone file and publishing the root zone to all the root servers.

Now we go to the resolution process. This is actually what is happening if you are making a query either on your phone. It doesn't have to be just your phone. It can be your laptop or it could be another client. Each of your clients – laptop, phone – have this stub resolver which is local to your client.

Then it will ask a question, "What is the IP address of www.example.com?" That question will go to your recursive server with an IP address of 4.2.2.2, and it will ask like this, "What is the IP address of www.example.com?" Your recursive name server will answer like this, "I don't know but probably the root server has that information."

Why your recursive name server doesn't have that information yet is because it's a brand new recursive name server. As mentioned earlier an empty or a brand new recursive server doesn't have all the cache information yet, so it will go directly to the root server to ask the IP address to go to the root server because the root server has the root zone file.

Then your root server will return a referral, "I don't know the address, but I know the address of .com." So your recursive name server will go to the .com servers and ask, "What is the IP address of www.example.com?" Then your .com name server will answer, "I don't know, but I know the IP address of the name server and ns1.example.com."

So your recursive server will now come to this name server ns1.example.com and then this name server will give out the IP address or return the definitive answer to your query. Then your recursive name server will now return the IP address to your stub resolver.

This happens in seconds. It doesn't happen in minutes. This is the same like if you just launch an application from your phone or from your laptop. Sometimes it takes a while to load the page or to open the application. But if you are trying to reload it again, it's way faster. Why? It's because the information is already cached to your client.

Let's take it again. Caching speeds up the resolution process because it now knows the name and IP address of your root zone and your name servers. If you try to access or if you try to request, "What is the IP address of the ftp.example.com?" a while ago we asked for the IP address of www.example.com.

Now we're asking what is the IP address of ftp.example.com. So your Safari or the stub resolver will go to the recursive name server again, but this time it will not go back to the root zone, but it will go directly to the name server because it has the cache information already. Using cache makes the entire process efficient and faster. This is how the resolution process works.

We have a one-page slide here about DNSSEC. If you want to go to a deep dive discussion about DNSSEC, there are a few sessions available. Do we have a session available for this week for the DNSSEC that they can attend?

JOHN CRAIN: I'll look it up, but we actually have a DNSSEC session coming up. I think it's on Wednesday, but I'll look it up before the end of the session.

UNIDENTIFIED MALE: There was also a tutorial yesterday.

RACHEL REYES: Okay, great. So basically, this is just the basics about DNSSEC. I'll just read it out to you. Through DNSSEC DNS data can be digitally signed for authentication. Each zone has a public or a private key to pair and to work the DNSSEC.

Several records in the DNSSEC are DNSKEY which is the public key for a zone, RRSIG or the digital signature. NSEC or NSEC3 is the pointer to the next name in a zone, and DS is the delegation signer.

Again, if you would like to know more about DNSSEC, you can attend one of the DNSSEC sessions that we have here.

The domain name ecosystem is like this. We have the registry which has the database for the domain names and registrants, followed by the registrar which is the primary agent between the registrant and the registry, and the registrant is the holder of the domain name registration.

This is how the domain name registry process, but we're not going to discuss the entire process. What I'm just trying to tell you is that what we have discussed is part of the entire domain name registry. What we have discussed is under here under the authoritative name servers, recursive name server, and the Internet user.

That's all I have for today's session if anyone has a question.

JOHN CRAIN: Before we go to questions, I'll just give the DNSSEC data. It's Wednesday from 9:00 AM until 3:00 PM, and it's a whole day on DNSSEC. More than anybody needs.

RACHEL REYES: Okay.

SIRANUSH VARDANYAN: Yes, we can now start the Q&A part. Yes please.

NICOLAS FIUMARELLI: Hello. Nicolas Fiumarelli from Uruguay. You mentioned that DNS is not case sensitive, but what happens in the case of internationalized domain names?

RACHEL REYES: Matt can answer that question.

MATT LARSON: DNS itself is definitely not case sensitive. Internationalized domain names are like a layer on top of DNS. Could I ask you, Rachel, to go way, way back to one of those namespace slides at the beginning. Keep going please. That's good. Thank you.

If you look at the slide, I should say the node on the upper left, the one that starts with xn--, the way we decided to do internationalized domain names was as I said to implement them as a layer on top.

From the user's perspective, if an application is IDN enabled, the use interacts with it and sees domain names in internationalized characters. But that application then has to convert them into this LDH – letters, digits, hyphen – format that DNS knows about.

So from DNS' perspective, they just look like regular labels, albeit kind of funny. You can see xn-- is a code that means the rest of this label is an encoded IDN. There's actually a special encoding called Punycode which is sort of a pun on Unicode that was designed specifically to encode Unicode characters for labels in DNS.

NICOLAS FIUMARELLI: Thank you.

MATT LARSON: Yeah. Just to give a little more background on that, when we were doing this there were some people who said, "Well, why do we need this layer on top of DNS? Why don't we just put, say, UTF-8 right into DNS? Let's have UTF-8 labels in DNS."

There was concern understandably that the DNS system wouldn't expect that because that's not what it was designed to do. So we'd have to upgrade the entire DNS infrastructure, and we would still have to upgrade all the clients to put UTF-8 into labels.

So the thinking behind the way we did do IDNs was, "Well, we still have to upgrade all the clients, but at least we don't have to touch the rest of the DNS infrastructure." So it's a question of how much did you want to touch. Did you want to touch everything, applications and DNS infrastructure, or just applications?

SIRANUSH VARDANYAN: Good. We have a question over here. Please.

ABDULKARIM OLOYEDE: Thank you. I want to first of all ask a follow up question. You said there's a layer on top of DNS. At the same time from my own understanding it means if you send a query to the DNS server, you send the entire thing on to the DNS server. So how do you now have this layer sitting on top of DNS? That's just like a follow up question before I ask mine.

MATT LARSON: Sure. What I mean by a layer on top of DNS is that's what it is conceptually, but it's actually inside any application that understands IDNs.

For example, in a modern web browser that understands IDNs, you would type in some non-Latin characters and it would convert that into something that looks like the xn-- label. It might be xn--., xn--something. It's on a label by label basis that it's internationalized. So then that web browser send the query, calls the stub resolver, and the stub resolver sends the query to the name server and that query just has xn-- in the labels.

So what I meant by layer on top of DNS is that it's done in the application, not in DNS servers and resolvers.

ABDULKARIM OLOYEDE: Thanks. Now I'll go into my question. I have two questions. The first one is, when you were giving the presentation, I don't know. Probably I missed some part when I was eating or you were a bit fast. It's about the zone servers. You said it is a bit complicated. I was a little bit confused about zone servers. What do you mean by zone servers? Especially when you are talking about primary zone servers, secondary zone servers, then you were talking about some of these zone servers are like slaves and masters. Can you please just explain that part again?

Then the other part of my question is 4.2.2.2 is it meant for if you send any query, is it like the default recursive server for all queries?

JOHN CRAIN: If you're talking about the different types of name servers, we don't generally refer to them as zone servers, then there are basically three types of servers. There's the stub which sits on your laptop, for example, or on your phone and may sit in the operating system or it may sit in the application itself like in your browser. Those only answer questions.

Then you have the recursives which are normally either at your ISP or they may be on your home router at home. Those pass on the questions, if you like. Those are the ones that go out to the authoritative ones. Those are the ones that actually have the answers. That's why we call them authoritative. They have the authority to give the answer. That's what the zone files actually sit on is on the authoritative.

On the recursive, you have a querying engine that goes and queries. And the only data that's really stored on that is in the cache. So it has memory that it remembers its answers. And your stub resolver may also have. So you have a path from your device forever up.

Now there was another comment I think that Rachel made which was about the complexity of how the root zone is provisioned. I don't know if you were referring to that. That's a whole different thing. That's a whole provisioning system rather than a name server system. I'll let Matt talk to that if he wants to because he worked on some of that stuff from the other side.

As to which recursive server you use, typically that's defined when you set up your laptop or your network. When you connect through a network, say here, we use something called dynamic host configuration protocol or DHCP. That's what sends you that IP address you use, but it also may send you you're domain names, your recursive servers. You can have two of those. You can have one of those. You can have four of those. All your queries should then go to the ones that have been configured.

SIRANUSH VARDANYAN: There is a question from a remote participant. Okay, please finish this one.

MATT LARSON: Specifically you asked about 4.2.2.2. That's a server that's operated by Level 3 Communications, an ISP. It's what's called an open recursive server. Usually, anywhere you have a bunch of clients – so that's any network, either an ISP for their broadband customers or in this building on the ICANN network – wherever there are a bunch of clients with stub resolvers (so the lower left) you need to have a recursive server at the top.

As John said, the network operator is responsible for providing that recursive server, and when you connect to the network your device gets the IP address of that recursive server to configure itself.

However, you don't have to use that recursive server. You can use others. There are some very popular open recursive servers, which means they accept queries from anyone. You could call these third-party recursive servers, public recursive servers. Probably the most popular because it has such an easy address to remember is Google Public DNS which is 8.8.8.8. So you could configure your stub resolver on your phone if you wanted. You could change it's configuration to go to 8.8.8.8 instead of what you're ISP gives you.

Other popular ones, Open DNS has been around for a long time. They were some of the first people to say let's put recursive servers outside of your network, and we'll offer this service. Verisign has them. PCH has something called Qaud9, 9.9.9.9. So there are several public ones, and 4.2.2.2 is one from Level 3 that has been around a long, long time.

SIRANUSH VARDANYAN: Thank you. There is a question from a remote participant, [inaudible] from Africa. "Africa seems to be developing gradually and there is the need to increase its capacity. Apart from the

Fellowship program that is filling in the gap in the developing countries, what is ICANN or for that matter DNSSEC doing to increase its capacity? And what do we have to do in Africa in terms of policy?”

JOHN CRAIN: Capacity building, we’re talking about DNS, so I’ll talk about capacity around DNS. We generally work with the community. My group specifically, we do a lot of capacity building, a lot of training. But in Africa specifically, you’re more likely to see organizations like AfriNIC or if you go to AFNOG or AfTLD (as in AF top-level domain), you’re more likely to see them actually providing these kinds of trainings. And we work with them and support them.

There’s another organization called the Network Startup Resource Center that’s also heavily active in Africa teaching about DNS and other infrastructure issues.

On DNSSEC specifically, we’ve done multiple trainings in the African region both with AFNOG and basically all the AF, the African organizations down there. I’m not sure when the next one is scheduled, but I think there is a couple of DNSSEC hands-on trainings scheduled for Africa in the May/June time.

So we are actually quite active down there, but we rely on local expertise. If you're trying to train the world, and it's a big place, then that's not ICANN's job. ICANN is, of course, a small organization. Some people think it's too big still, but it's still a small organization. So we really reach out to the local technical communities and help them by either giving them materials or working on their materials with them. That scales a lot better than many others.

We're also working on some online learning platforms which will enable us to provide e-learning capabilities and will also enable us to translate those more easily into different languages.

So although we're not the world's university here, we do actually spend a lot of time trying to educate people. One of the things is if you see in my title I have the terms "security, stability, and resiliency," worrying about the ecosystem. One of the things you do to improve that is to make sure that people have better access to knowledge and the ability to build better systems.

SIRANUSH VARDANYAN: Thank you. Lendon?

LENDON TELESFORD: Hi. I'm Lendon from Grenada. I'm not sure it's a question on DNS itself or if it's on the entire system. I'm not sure if this is possible, but I'm going to ask anyway. In the presentation it was highlighted with Anycast and the different instances of the root servers. My question is within the client server and Anycast scheme, what mechanisms are there in place to protect against a type of DDoS attack that affects the perception of proximity so that clients become confused as to which server to visit for a response?

JOHN CRAIN: I'm trying to think the question through. Do you want to take it?

MATT LARSON: Well, I'm not quite sure what you mean by clients becoming confused about which server. I'm going to use a layer concept again in my answer. Anycast is really a layer below DNS. We have DNS and then Anycast is part of the Internet's routing system.

Let's take the root server system where I believe at this point now every single one of the IP addresses is Anycast. Let's say we have a recursive name server that's going to send a query to L-root. From the DNS layer, it just says, "I'm sending a query to this IP address."

But when that gets down to the routing layer, when the network itself has to actually transport that packet, the routers on the network will see that there are actually multiple instances, multiple places on the network where that IP address is available. In terms of how we would say that with BGP, the routing protocol, we would say that different networks announce the route to that particular network. So you have multiple networks all over the Internet saying, "I can get to L-root." "I can get to L-root." "I can get to L-root."

Individual routers then using information from BGP make their decision. "From my perspective, what's the best way to get to L-root?" All the routers do this. What that means is when you say, "I want to send a packet to L-root," based on where you are in the network you go to the "closest" instance. It's not just geographic distance. It's not just latency. There are all kinds of factors that affect BGP routing policy.

I'm not sure how we factor confusion into that. What can happen is if there were an attack on one instance and it became overloaded, the way operators usually configure name servers in an Anycast configuration is when the name server is a live, it's telling the network, "I'm here. You can advertise my route, that I exist." But if it becomes so congested that it collapse – depending on the failure – if it fails gracefully, then it would tell the network, "Oh, I'm not alive anymore. I can't accept DNS queries." Then the network recalculates, and it would send traffic to somewhere else.

LENDON TELESFORD: So on the router side, the decisions about where to route it is made based on prefilled BGP information?

MATT LARSON: I wouldn't say prefilled. It's changing all the time. Every network, every autonomous system number we call it which is a network from BGP's perspective, and autonomous system is a network that has a collection of routes, of networks that it advertises, that it says, "I have these IP addresses."

So every router that does BGP is constantly saying, "These are the networks I know about" and other router hear and decide on a real-time basis where everything is. That's a very, very simplified version of how it works, but BGP happens in real time constantly.

JOHN CRAIN: To add to that, this isn't something that's caused necessarily by Anycast. This was around before Anycast because you see multiple paths to a node. If somebody was advertising a website, if I'm really far away I'm probably only going to see one way out to send it, but if I'm fairly close I might see multiple paths to that. And they did the same thing back in the day. So it's a routing trick that was recognized and then used to add more servers. There was no technology change for Anycast. It was just a trick of the routing.

LENDON TELESFORD: I guess I was just wondering if there's a way to trick the routing trick.

JOHN CRAIN: I don't think that part of the trick. Now routing has its own security issues, but they're related to routing not to Anycast. Routing does have some security issues, or lack of security issues is probably a better way of putting it. But they're not specific to Anycast.

SIRANUSH VARDANYAN: Thank you. Over there please.

SHABNIL ANAL SAMI: Hi. This is Shabnil from Fiji. My question is if a country wants to host a root server, what is the best practice when they are deciding which one to host? Like from the 13, A, L, F, etc.? Or is it based on region or anyone can host?

JOHN CRAIN: It's based on who you decide to go to talk to. The first thing I will say is it's not a country. It's a network or a network operator that wants to host an instance. They can go to root-servers.org and see the list of operators. We're one of the operators. I think we actually have one in Fiji, and I think a couple of other people do.

You can just reach out to those operators and ask what are their conditions for doing this. They're slightly different by each operator, but it's not hard. There are 990 instances or locations today. So adding some more, it's just a matter of reaching out. You go to the root-servers.org website and there are actually links on there to each of the operators and you can see their documentation of how you get them. Happy to talk offline and help you through that.

SHABNIL ANAL SAMI: I was just concerned about which one to host like L, F. So basically just go? Because I saw Puerto Rico is hosting L, so Fiji was hosting L, so I was confused about the regional stuff like that. Thank you.

JOHN CRAIN: Yeah, it's not a regional issue. It's often a relationship issue. The reason that Fiji has L, which was the second one to go there, is because we have a staff member in Fiji, Save Vocea. So a lot of the time it’s about who you have relationships with or are building relationships with by going on the website and saying, “Oh, this one looks good.”

Also, the criteria, mostly the financial criteria of how you do this, may differ from one to the other. We, for example, at ICANN have a solution where you pay for the server and you put it online and then we do all the work. Others, you give them the money and then they send you the server. It’s just a slightly different model, so you need to figure out which one suits you.

All root servers from a DNS perspective are equal. They all give you the same answers. They all work from a query perspective exactly the same. So it’s really about the business relationships and maybe who you know.

SHABNIL ANAL SAMI: Okay, thank you.

SIRANUSH VARDANYAN: Any other questions? Yes, please.

UNIDENTIFIED MALE: I have two questions. How can to ensure [the duplication] of the areas from the registries of countries are the correct and not [pulling] incorrect [inaudible] like happened years ago? The second one is the copies of the root servers that are hosted in countries for example are just a part of the [inaudible] around the world, just a part of the [inaudible] in this root. For example the root L has just a part of all the [inaudible] or have all the records there?

MATT LARSON: No, all the root servers have the same information. There's only one root zone with its information, and then every root server has that same information.

Could you please repeat your first question?

UNIDENTIFIED MALE: For example, when a registry [inaudible].

SIRANUSH VARDANYAN: Just a question: there is a translation. You can do it in your own language.

UNIDENTIFIED MALE: Okay.

MATT LARSON: Perfect. Perfect.

SIRANUSH VARDANYAN: So use this opportunity.

UNIDENTIFIED MALE: Well, okay. One moment please.

SIRANUSH VARDANYAN: Yeah. Just a second for people to take their headsets.

UNIDENTIFIED FEMALE: Check. Can you get the translation in English? Are you getting the English? Okay, great. Thank you.

UNIDENTIFIED MALE: [through translator] When the records in each country have a record in the zone like a ccTLD, when the registry has a record in the zone and that is published and distributed throughout the DNS zones, how can you make sure that those publications are the correct ones and that they are no mistakes in the publishing of those zones or in the distribution? Because I understand that this has already happened in the past.

JOHN CRAIN: There were a few questions in there. I'll make sure of terminology and technology. So irrespective of ccTLD or gTLD, when they publish the zone that zone is made up of data that comes out of their databases and they are responsible for ensuring the correctness of that data. Once it's actually published, if you DNSSEC sign the zone and you are authenticating the DNS queries, then you can ensure that you've got what they published.

Now that's the end of it. That's the publishing and the DNS side. All the other side of it is really around network security and database security and integrity of your data. There have been problems in the past where people have hacked into ccTLD systems, which is what I think you're referring to.

UNIDENTIFIED MALE: Yes.

JOHN CRAIN: There is no such thing as a truly secure network. So I think people will always be susceptible at some level to this. What we do as ICANN and with our friends in those registries – specifically my group, the security group – when they've had issues they will reach out to us and we will help them with recovery. We will often help them also with finding experts to help them redesign their systems.

We've had a couple, and I think I know which one you're referring to, but I'm not going to name them. We've had a couple of cases where they had SQL injection attacks. It's a specific type of attack against their system.

That allowed the miscreant, the bad person, to change records for a large organization and point the web server address somewhere else. So it looked like – so I am talking about the correct one I think I'm talking about – what that then looks like to the outside world is that web server has been hacked. But in fact it's not. It was the registry systems.

So in that case, we spent a lot of time working with the registry operator, and they now have a completely new system that has had much more auditing, etc., against it. They've learned from their mistakes and moved forward, and that's what you do in the case of a compromise. You understand how you got compromised, you fix your systems, and you learn from that and improve your processes.

But that doesn't mean that some other ccTLD somewhere or even a gTLD will never, ever get hacked because that's just not the reality. We see major corporations with millions of dollars in funds also being attacked.

UNIDENTIFIED MALE: Thank you very much.

SIRANUSH VARDANYAN: Yes, please.

JASON HYNDS: So, John, a follow up to that.

SIRANUSH VARDANYAN: Your name please?

JASON HYNDS: Oh, sorry. Yes. I'm Jason Hynds from Barbados. John, my follow up question to that is, are there any publications that you make to help registry operators prevent these common compromises before they happen?

JOHN CRAIN: Working with some of the partners that I mentioned earlier around the capacity building, we've done thousands of hours of training with operators around things like, how do you secure a network? How do you monitor a network? Of course, the community that you see here of operators also have their own groupings and share a lot of best practices and help, including engineering help.

So in Jamaica as part of the LAC region, there's an organization called LACTLD which includes all of the registries – not all of them but most of the registries – from the region. They have regular meetings, and they have technical sessions. When one of them has issues, the others come to their aid.

So it's not an issue that they face on their own. It's an issue that affects the whole community. Through training and mutual aid, there is a lot going on there.

SIRANUSH VARDANYAN: Questions? No more questions.

I would encourage you to participate for sure on Wednesday for this DNSSEC workshop which will be from 9:00 AM until noon.

UNIDENTIFIED MALE: Until 3:00.

SIRANUSH VARDANYAN: Until 3:00.

UNIDENTIFIED MALE: It's long.

SIRANUSH VARDANYAN: Okay, then in the middle we have a lunchtime Fellowship session. But anyway, please join that workshop. It's really very important.

Any final words from our presenters?

RACHEL REYES: Thanks, everyone, for staying with us. I appreciate your presence. Thanks, John and Matt, for helping me answer the questions.

SIRANUSH VARDANYAN: I would like to thank our interpreters and our tech team. My huge appreciation to Matt, John, and Rachel for your time. I know you are very busy during this meeting, but thank you for coming and doing this presentation for Fellows. Your applause to our presenters.

With that, our meeting is adjourned. Thank you.

[END OF TRANSCRIPTION]