The
Internet Protocol Journal

June 2010 Volume 13, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Protocol changes are never easy, especially when they involve something as fundamental as the *Internet Protocol* (IP). This journal has published numerous articles about the depletion of IPv4 addresses and several articles about IPv6, including methods for a gradual transition from v4 to v6. A lot of energy has gone into the development, promotion, and deployment of IPv6, but in reality only a small fraction of the global Internet currently supports IPv6. Meanwhile, the *Internet Assigned Numbers Authority* (IANA) and the *Regional Internet Registries* (RIRs) will "soon" (12 to 24 months from now is predicted) run out of IPv4 addresses to allocate. Although this situation has some serious implications for new entrants to the *Internet Service Provider* (ISP) market, it does not spell the end of the Internet as we know it. Numerous *Network Address Translation* (NAT) solutions are already widely deployed, and the IETF is discussing other solutions. One example is *Address Sharing* as explained by Geoff Huston in our first article.

Changes to the *Domain Name System* (DNS) are also underway. The *Domain Name System Security Extensions* (DNSSEC) are being gradually deployed in the global Internet. As with any complex technology, implementation of DNSSEC is not without problems. Our second article, by Torbjörn Eklöv and Stephan Lagerholm, is a step-by-step guide for those considering implementing DNSSEC in their network.

By now you will be aware that we have implemented a renewal system for subscribers and will not be automatically extending your subscription unless you contact us via e-mail or use the online tool to renew your subscription. You can find your subscription ID and expiration date either on the back page of your copy or on the envelope that it came in. In order to access your record, click the "Subscriber Services" link on our webpage at **www.cisco.com/ipj**, and enter your e-mail address and the subscription ID. The system will send you a link that allows direct access to your record, and you will be able to update your address and renew your subscription. If you no longer have access to the e-mail you used when you subscribed, or have forgotten your subscription ID, just send a message to **ipj@cisco.com** and we will make the necessary changes for you.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

NAT++: Address Sharing in IPv4

by Geoff Huston, APNIC

In this article I examine the topic that was discussed in a session at the 74th meeting of the *Internet Engineering Task Force* (IETF) in March 2009, about *Address Sharing* (the SHARA BOF)[0], and look at the evolution of *Network Address Translation* (NAT) architectures in the face of the forthcoming depletion of the unallocated IPv4 address pool.

Within the next couple of years we will run out of the current supply of IPv4 addresses. As of the time of writing this article, the projected date when the *Internet Assigned Numbers Authority* (IANA) pool will be depleted is August 3, 2011, and the first *Regional Internet Registry* (RIR) will deplete its address pool about March 20, 2012.

Irrespective of the precise date of depletion, the current prediction is that the consumption rate of addresses at the time when the free pool of addresses is exhausted will probably be running at some 220 million addresses per year, indicating a deployment rate of some 170–200 million new services per year using IPv4. The implication is that the Internet will exhaust its address pool while operating its growth engines at full speed.

How quickly will IPv6 come to the rescue? Even the most optimistic forecast of IPv6 uptake for the global Internet is measured in years rather than months following exhaustion, and the more pessimistic forecasts extend into multiple decades.

For one such analysis using mathematical modelling techniques, refer to Jean Camp's work[1]. One of the conclusions from that 2008 study follows: "There is no feasible path which results in less than years of IPv4/IPv6 co-existence. Decades is not unreasonable."

The implication of this conclusion is that we will need to operate a dual-stack Internet for many years to come, and the associated implication is that we will have to make the existing IPv4 Internet span a billion or more new deployed services—and do so with no additional address space.

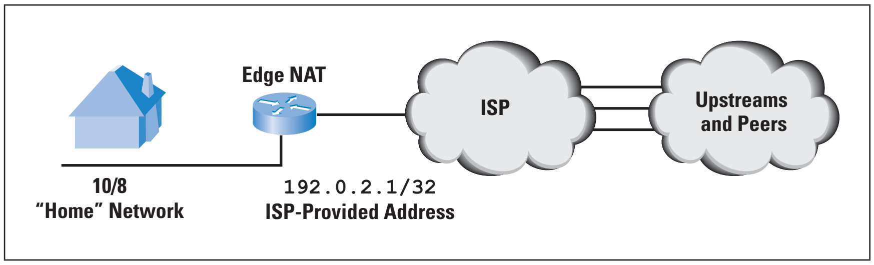
So how are we going to make the IPv4 address pool stretch across an ever-larger Internet?

Given that the tool chest we have today is the only one available, there appears to be only one answer to this question: Use *Network Address Translators,* or NATs.

For a description of how NATs work and some of the terminology used to describe NAT behavior, refer to the article "Anatomy: A Look Inside Network Address Translators," published in this journal[2].

Today NATs are predominately edge devices that are bundled with DSL modems for residential access, or bundled with routing and security firewall equipment for small to midsize enterprise use as an edge device. The generic model of NAT deployment currently is a small-scale edge device that generally has a single external-side public IP address and an internal-side private IP network address (often network 10). The NAT performs address and port translation to map all currently active sessions from the internal addresses to ports on the public IP address. This NAT deployment assumes that each edge customer has the unique use of a public IP address (refer to Figure 1).

Figure 1: Conventional NAT Deployment



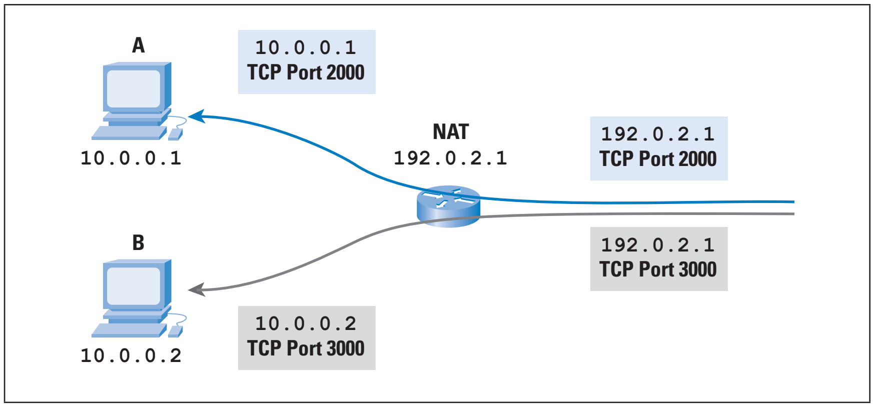
The question provoked by IPv4 address exhaustion is what happens when there are no longer sufficient IPv4 addresses to provide this 1:1 mapping between customers and public IPv4 addresses? In other words, what happens when there are simply not enough IPv4 addresses to allow all customers to have exclusive use of their own unique IPv4 address?

This question has only two possible answers. One is for no one to use IPv4 addresses at all, on the basis that the entire Internet has migrated to use IPv6. But this answer appears to be an uncomfortable number of decades away, so we need to examine the other answer: If there are not enough addresses to go around, then we will have to *share* them.

But isn't sharing IP addresses impossible in the Internet architecture? The IP address in a packet header determines the destination of the packet. If two or more endpoints share the same address, then how will the network figure out which packets go to which endpoint? It is here that NATs and the transport layer protocols, the *Transmission Control Protocol* (TCP) and the *User Datagram Protocol* (UDP), come together. The approach is to use the *port address* in the TCP and UDP header as the distinguishing element.

For example, in Figure 2, incoming TCP packets with TCP port address 2000 may need to be directed to endpoint A, while incoming TCP packets with TCP port address 3000 need to be directed to endpoint B. The incoming TCP packets with a port address of 2000 are translated to have the private IP address of endpoint A, and incoming TCP packets with a port address of 3000 are translated to have the private address of endpoint B.

Figure 2: Address Sharing with NATs



As long as you restrict yourself to applications that use TCP or UDP, you don't rely on receiving *Internet Control Message Protocol* (ICMP) packets, and you don't use applications that contain IP addresses in their payload, then you might expect this arrangement to function.

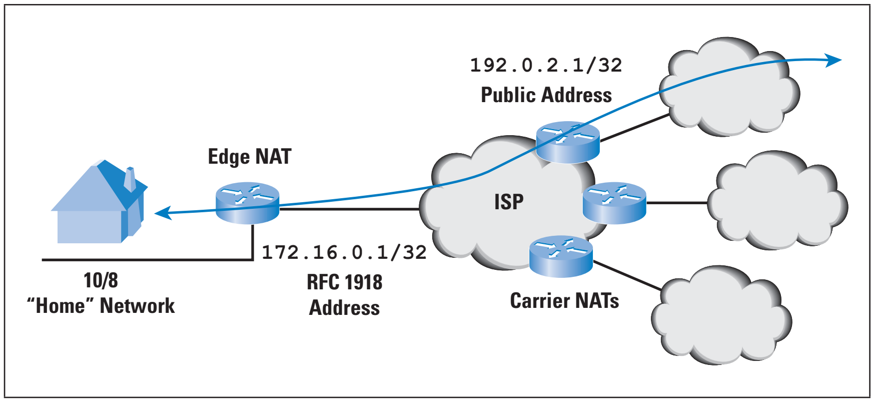
ICMP is a problem because the ICMP packet does not contain a TCP or UDP transport layer. All that a NAT sees in the ICMP packet is its own external address as the destination IP address. To successfully deliver an ICMP packet through a NAT, the NAT needs to perform a more complex function that uses the ICMP-encapsulated IP header to select the original outbound combined IP + TCP header or IP + UDP header in the ICMP payload. The source IP address and transport protocol port address in the ICMP payload are then used to perform a lookup into the NAT binding table and then perform two mappings: one on the ICMP header to map the destination IP address to the internal IP address, and the second on the payload header where the source IP address and port number are changed to the interior-side values, and the checksums altered as appropriate. Now in most cases ICMP really is not critical, and a conservative NAT implementation may elect to avoid all that packet inspection and simply discard all incoming ICMP messages, but one message that is important is the ICMP *packet-too-large-and-fragmentation-disabled* message used in IPv4 *Path MTU Discovery*[3].

Sharing IP addresses is fine in theory, but how can we achieve it in practice? How can many customers, already using NATs, share a single public IP address?

Carrier-Grade NATs

One possible response is to add a further NAT into the path. In theory the *Internet Service Provider* (ISP) could add NATs on all upstream and peer connections, and perform an additional NAT operation as traffic enters and leaves the ISP's network. Variations of this approach are possible, placing the ISP NATs at customer aggregation points within the ISP's network, but the principle of operation of the ISP NAT is much the same.

Figure 3: Carrier NATs



The edge NATs translate between private address pools at each customer's site and an external address provided by the ISP, so nothing has changed there. The change in this model is that the ISP places a further NAT in the path within the ISP network, so that a set of customers is then sitting behind a larger NAT inside the ISP's network, as shown in Figure 3.

This scenario implies that the external address that the ISP provides to the customer is actually yet another private address, and the ISP's NAT performs yet another transform to a public address in this second NAT. In theory this NAT is just a larger version of an existing NAT with larger NAT binding space, higher packet-processing throughputs, and a comprehensive specification of NAT binding behavior. In practice it may be a little more complicated because at the network edge the packet rates are well within the processing capability of commodity processors, whereas in the core of the network there is an expectation of higher levels of robust performance from such units. Because it is intended that such a NAT handle thousands of customers and large numbers of simultaneous data flows and peak packet rates, it requires a performance level well beyond what is seen at the customer edge and, accordingly, such a NAT has been termed a *Carrier-Grade NAT* (CGN), or a *Large-Scale NAT* (LSN).

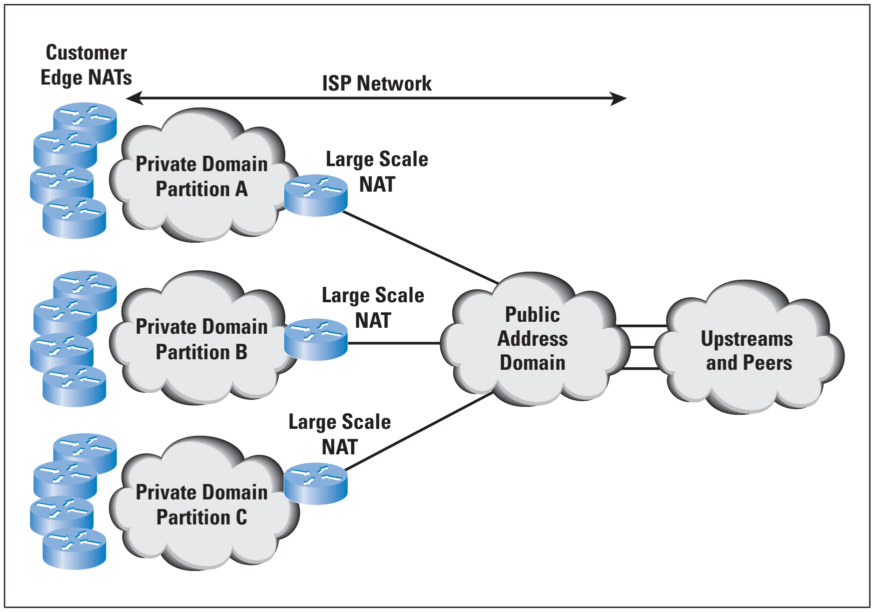
From the inside of the two NATs, not much has changed with the addition of the CGN in terms of application behavior. It still requires an outbound packet to trigger a binding that allows a return packet through to the internal destination, so nothing has changed there. Other aspects of NAT behavior, notably the NAT binding lifetime and the form of *Cone Behavior* for UDP, take on the more restrictive of the two NATs in sequence. The binding times are potentially problematic in that the two NATs are not synchronized in terms of binding behavior. If the CGN has a shorter binding time, it is possible for the CGN to misdirect packets and cause application-level problems. However, this situation is not overly different from a single-level NAT environment where aggressively short NAT binding times also run the risk of causing application-level problems when the NAT drops the binding for an active session that has been quiet for an extended period of time.

However, one major assumption is broken in this structure, namely that an IP address is associated with a single customer. In this model a single public IP address may be used simultaneously by many customers at once, albeit on different port numbers. This scenario has obvious implications in terms of some current practices in filters, firewalls, "black" and "white" lists, and some forms of application-level security and credentials where the application makes an inference about the identity and associate level of trust in the remote party based on the remote party's IP address.

This approach is not without its potential operational problems as well. For the ISP, service resiliency becomes a critical concern in so far as moving traffic from one NAT-connected external service to another will cause all the current sessions to be dropped, unless the internal ISP network architecture uses a transit access network between the CGNs and the external transit providers. Another concern is one of resource management in the face of potentially hostile applications. For example, an end host infected with a virus may generate a large amount of probe packets to a large range of addresses. In the case of a single edge NAT, the large volumes of bindings generated by this behavior become a local resource management problem because the customer's network is the only affected site. In the case where a CGN is deployed, the same behavior starts to consume binding space on the CGN and, potentially, can starve the CGN of external address bindings. If this problem is seen to be significant, the CGN would need to have some form of external address rationing per internal client in order to ensure that the entire external address pool is not consumed by a single errant customer application. This "rationing" would have the unwanted effect of forcing the ISP to deny access to its customers.

The other concern here is one of scalability. Although the greatest leverage of the CGN in terms of efficiency of usage of external addresses occurs when the greatest numbers of internal edge-NAT-translated clients are connected, there are some real limitations in terms of NAT performance and address availability when an ISP wants to apply this approach to networks where the customer population is in the millions or larger. In this case the ISP is required to use an IPv4 private address pool to number every client. But if all customers use network 10 as their "internal" network, then what address pool can the ISP use for its private address space? One of the few answers that come to mind is to deliberately partition the network into numerous discrete networks, each of which can be privately numbered from the smaller private address pool of `172.16.0.0/12`, allowing for some 600,000 or so customers per network partition, and then use a transit network to "glue" together the partitioned elements, as shown in Figure 4.

Figure 4: Multiple Carrier NAT Deployment Using Network Partitioning



The advantage of the CGN approach is that for the customer nothing changes. Customers do not need to upgrade their NAT equipment or change them in any way, and for many service providers this motivation is probably sufficient to choose this path. The disadvantages of this approach lie in the scaling properties when looking at very large deployments, and the problems of application-level translation, where the NAT attempts to be "helpful" by performing deep packet inspection and rewriting what it thinks are IP addresses found in packet payloads. Having one NAT do this rewriting is bad enough, but loading them up in sequence is a recipe for trouble!

Are there alternatives?

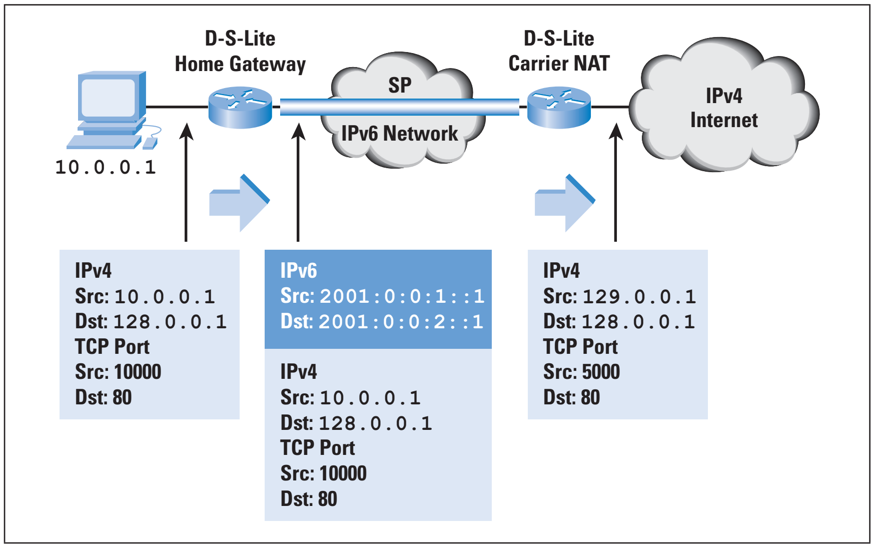
Dual-Stack Lite and Carrier-Grade NATs

One rather elegant alternative is described by Alain Durand and others in an Internet Draft "Dual-stack lite broadband deployments post IPv4 exhaustion"[4]. The assumption behind this approach is that the ISP's network infrastructure needs to support IPv6 running in native mode in any case, so is there a way in which the ISP can continue to support IPv4 customers without running IPv4 internally?

Here the customer NAT is effectively replaced by a tunnel ingress/egress function in the *Dual-Stack Lite Home Gateway*. Outgoing IPv4 packets are not translated, but are encapsulated in an IPv6 packet header, where the IPv6 packet header contains a source address of the carrier side of the home gateway unit and a destination address of the ISP's gateway unit. From the ISP's perspective, each customer is no longer uniquely addressed with an IPv4 address, but instead is addressed with a unique IPv6 address. The customer's interface to the ISP network, the Home Gateway, is configured with this IPv6 address as the customer end of the IPv4-in-IPv6 tunnel, where the other end of the tunnel is the IPv6 address of the ISP's Dual-Stack Lite Gateway unit.

The service provider's Dual-Stack Lite gateway unit performs the IPv6 tunnel termination and a NAT translation using an extended local binding table. The "interior" NAT address is now a 4-tuple of the IPv4 source address, protocol ID, and port, plus the IPv6 address of the home gateway unit, while the external address remains the triplet of the public IPv4 address, protocol ID, and port. In this way the NAT binding table contains a mapping between interior "addresses" that consist of IPv4 address and port plus a tunnel identifier and public IPv4 exterior addresses. This way the NAT can handle a multitude of network 10 addresses, because the addresses can be distinguished by different tunnel identifiers. The resultant output packet following the stripping of the IPv6 encapsulation and the application of the NAT function is an IPv4 packet with public source and destination addresses. Incoming IPv4 packets are similarly transformed, where the IPv4 packet header is used to perform a lookup in the Dual-Stack Lite gateway unit, and the resultant 4-tuple is used to create the NAT-translated IPv4 packet header plus the destination address of the IPv6 encapsulation header (refer to Figure 5).

Figure 5: Dual-Stack Lite



IPv4
Src: 10.0.0.1
Dst: 128.0.0.1
TCP Port
Src: 10000
Dst: 80

IPv6
Src: 2001:0:0:1::1
Dst: 2001:0:0:2::1

IPv4
Src: 10.0.0.1
Dst: 128.0.0.1
TCP Port
Src: 10000
Dst: 80

IPv4
Src: 129.0.0.1
Dst: 128.0.0.1
TCP Port
Src: 5000
Dst: 80

The advantage of this approach is that now only a single NAT is needed in the end-to-end path because the functions of the customer NAT are now subsumed by the carrier NAT. This scenario has some advantages in terms of those messy "value-added" NAT functions that attempt to perform deep packet inspection and rewrite IP addresses found in data payloads. There is also no need to provide each customer with a unique IPv4 address, public or private, so the scaling limitations of the dual-NAT approach are also eliminated. The disadvantages of this approach lie in the need to use a different *Customer Premises Equipment* (CPE) device, or at least one that is reprogrammed. The device now requires an external IPv6 interface and at a minimum an IPv4 or IPv6 tunnel gateway function. The device can also include a NAT if desired, but it is not required in terms of the basic Dual-Stack Lite architecture.

This approach pushes the translation into the middle of the network, where the greatest benefit can be derived from port multiplexing, but it also creates a critical hotspot for the service itself. If the carrier NAT fails in any way, the entire customer base is disrupted. It seems somewhat counter intuitive to create a resilient network with stateless switching environments and then place a critical stateful unit in the middle! So is there an approach that can push this translation back to the edges while avoiding a second NAT in the carrier's network?

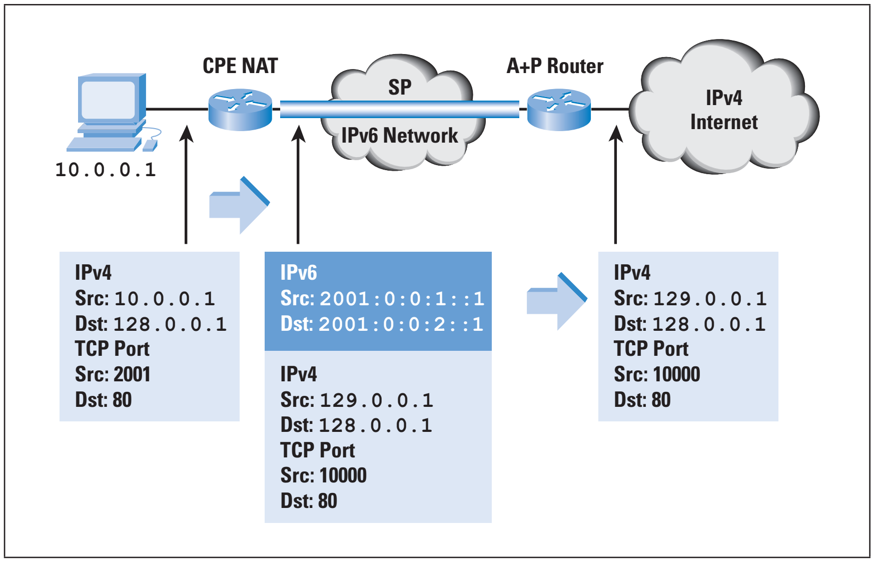
The Address Plus Port Approach

The observation here is that CPE NATs currently map connections into the 16-bit port field of the single external address. If the CPE NAT could be coerced into performing this mapping into 15 bits of the port field, then the external address could be shared between two edge CPE devices, with the leading bit of the port field denoting which CPE device. Obviously, moving the bit marker across the port field would allow more CPE devices to share the one address, but it would reduce the number of available ports for each CPE device in the process.

The theory is again quite simple. The CPE NAT is dynamically configured with an external address, as happens today, and a port range, which is the additional constraint. The CPE NAT performs the same function as before, but it is now limited in terms of the external ports it can use in its NAT bindings to those that lie within the provided port range, because some other CPE may be concurrently using the same external IP address with a different port range.

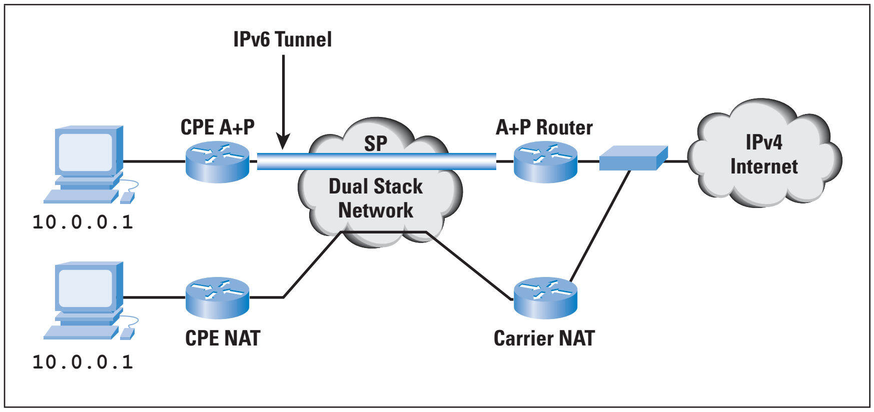
For outgoing packets this limitation implies only a minor change to the network architecture, in that the RADIUS[9] exchange to configure the CPE now must also provide a port range to the CPE device. However, the case of incoming packets is more challenging. Here the ISP must forward the packet based not only on the destination IP address, but also on the port value in the TCP or UDP header. A convenient way to forward the packet is to take the Dual-Stack Lite approach and use an IPv4-in-IPv6 tunnel between the CPE and the external gateway (Figure 6). This gateway, or *Address Plus Port* (A + P) router, needs to be able to associate each address and port range with the IPv6 address of a CPE device, which it can learn dynamically as it decapsulates outgoing packets. Corresponding incoming packets are encapsulated in IPv6 using the IPv6 destination address that it has learned previously. In this manner the NAT function is performed at the edge, much as it is today, and the interior device is a more conventional form of tunnel server.

Figure 6: Address Plus Port Framework



This approach relies on every CPE device being able to operate using a restricted port range, to perform IPv4-in-IPv6 tunnel ingress/egress functions, and to act as an IPv6 provisioned endpoint for the ISP network, which is perhaps an unrealistic hope. Further modifications to this model (Figure 7) propose the use of an accompanying CGN operated by the ISP to handle those CPE devices that cannot support these Address Plus Port functions.

Figure 7: Combined Address Plus Port and Carrier Grade NAT



If the port range assigned to the CPE is from a contiguous range of port values, then this approach could exacerbate some known problems with infrastructure protocols. There are *Domain Name System* (DNS) problems with guessable responses. The so-called "Kaminsky Attack" on the DNS[5, 6] is one such example where the attack can be deflected, to some extent, by using a randomly selected port number for each DNS query. Restricting the port range could mitigate the efficacy of such measures under certain conditions.

However, despite such concerns, the approach has some positive aspects. Pushing the NAT function to the edge has some considerable advantage over the approach of moving the NAT to the interior of the network.

The packet rates are lower at the edge, allowing for commodity computing to process the NAT functions across the offered packet load without undue stress. The ability for an end-user's application to request a particular NAT binding behavior by speaking directly with the local NAT using the *Internet Gateway Device Protocol,* as part of the *Universal Plug and Play* (UPnP)[7] framework, will still function in an environment of edge NATs operating with restricted port ranges. Aside from the initial provisioning process to equip the CPE NAT with a port range, the CPE, and the edge environment is largely the same as in today's CPE NAT model.

That is not to say that this approach is without its negative aspects, and it is unclear as to whether the perceived benefits of a "local" NAT function outweigh the problems associated with this model of address sharing. The concept of port "rationing" is a very suboptimal means of address sharing, given that after a CPE device has been assigned a port range those port addresses are unusable by any other CPE. The prudent ISP would assign to each CPE device a port address pool equal to some estimate of peak demand, so that, for example, each CPE device would be assigned 1,000 ports, allowing a single external IP address to be shared across only 60 such CPE clients. Neither the Carrier-Grade NAT approach nor the Dual-Stack Lite approach attempts this form of rationed allocation, allowing the port address pool to be treated as a common resource, with far higher levels of usage efficiency through dynamic management of the port pool.

The difference here is that in the dynamically managed approach any client can use the currently unused port addresses, whereas in the rationed approach each client has access to a fixed pool of port addresses that cannot be shared with any other client—even when the client does not need them. The difference here parallels the difference in network efficiency between time-division multiplexed synchronous circuits and asynchronous packets at Layer 2 in the network model. In the Address Plus Port framework the leverage obtained in terms of making efficient use of coopting these additional 16 bits of port address into the role of additional bits of client identifier address space is reduced by the imposition of a fixed boundary between customer and ISP use in the port address plan. The central NAT model of a CGN effectively pools the port address range and facilitates far more efficient sharing of this common port address pool across a larger client base.

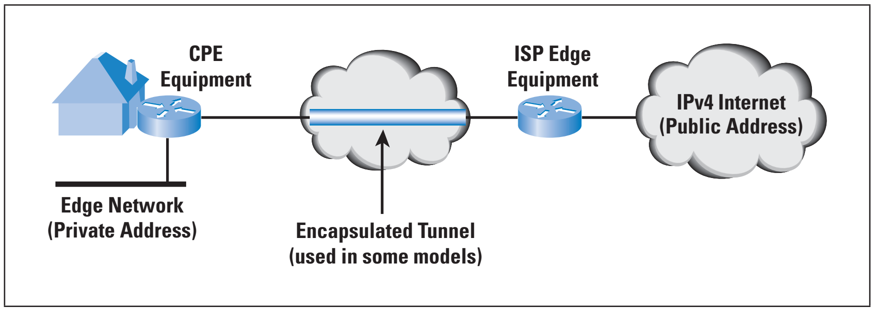
Alain Durand reported to IETF 74 on a data-collection experiment using a *Cable Modem Termination System* (CMTS) with 8,000 subscribers where the peak port consumption level was 40,000 ports, or a maximum average port consumption of 5 ports per subscriber in each direction. As Alain noted, this average value needs to be compared with the hundreds of ports consumed by a single client browsing a Web 2.0 or *Asynchronous Java and XML* (AJAX) site, but he also noted that a central model of port sharing does yield far higher levels of address-sharing efficiency than the Address Plus Port advanced allocation model.[8]

The other consideration here is that this approach constitutes a higher overhead for the ISP, in that the ISP must support both "conventional" CPE and Address Plus Port equipment. In other words, the ISP must deploy a CGN and support customer CPE using a two-level NAT environment in addition to operating the Address Plus Port infrastructure. Unless customers would be willing to pay a significant price premium for such an Address Plus Port service, it is unlikely that this option would be attractive for the ISP as an additional cost after the CGN cost.

General Considerations with Address Sharing

The basic elements of any such approach to address sharing involve the CPE equipment at the edge, optionally some form of tunneling of traffic between the CPE and the carrier equipment, and carrier-provided equipment at the edge of the carrier's network (refer to Figure 8).

Figure 8: Generic Architecture for Address Sharing


CPE Equipment
ISP Edge Equipment
IPv4 Internet (Public Address)
Edge Network (Private Address)
Encapsulated Tunnel (used in some models)

A variety of technical solutions here involve these basic building blocks, so it is not true to say that this challenge is technically significant. But few ISPs have decided to proceed with large-scale deployment of any form of address-sharing technology for their IPv4 network infrastructure. So what is the problem here?

I suspect that the real concern is the consideration of the relevant business model that would guide this deployment. Today's Internet is large. It encompasses some 1.7 billion human users, a larger pool of devices, and hundreds of millions of individual points of control. If we want to change this deployed system, we will need copious quantities of money, time, and unity of purpose. So do we have money, time, and unity of purpose?

Money is missing: It could be argued that we have left the entire IPv6 transition effort to this late stage because of a lack of money. The main advantage of the Internet was that it was cheap. Packet sharing is intrinsically more efficient than circuit sharing, and the shift in functions of network service management from the network to the customer-owned and -operated endpoints implied further cost savings for the network operator. So the Internet model gained ascendency because for consumers it represented a cost-effective choice. It was cheap.

But what does IPv6 offer consumers? For existing Internet consumers it appears that IPv6 does not offer anything that they don't already have with IPv4—it offers mail, the web, various forms of voice services, and games. So consumers are not exactly motivated to pay more for the same services they already enjoy today.

In addition, it would appear that the ISP must carry this cost without incremental revenue from its customer base. But the ISP industry has managed to shave most of its revenue margins in a highly competitive industry, and at the same time lose control of services, their delivery, and their potentially lucrative revenue margins. Thus the ISP industry has been collectively idle in this area not because it cannot see the problem in terms of the imminent exhaustion of IPv4, but because it has little choice because of financial constraints that have prevented it from making the necessary longer-term investments in IPv6. So if the ISP industry has been unwilling to invest in IPv6 so far, then what incentive is there for it to invest in IPv6 and at the same time also invest in these IPv4 address-sharing measures? Is the lure of new, low-margin customers sufficient incentive to make such investments in this carrier-grade equipment? Or is the business case still insufficiently attractive?

Time is missing: The unallocated IPv4 address pool is already visibly waning. Without any form of last-minute rush, the pool will be around for the next 2 years, or until 2012 or so. But with any form of typical last-minute rush, this pool could be depleted in the coming months rather than in the coming years. Can we do what we need to do to get any of these approaches to a state of mass-market deployment in the next few months? All these approaches appear to be at the early stages of a timeline that starts with research and then moves on to development, prototyping, and trials; then to standards activity and industry engagement to orchestrate supply lines for end user equipment, ISP equipment, and definition of operational practices; then to product and service development; and finally, to deployment. For an industry that is the size of the Internet, "technical agility" is now an obsolete historical term. Even with money and unity of purpose this process will take some years, and without money—or even the lure of money—it becomes a far more protracted process, as we have seen already with IPv6 deployment.

And do we have *unity of purpose* here? Do we agree on an approach to address sharing that will allow players to perform their tasks? That will allow consumer product vendors to develop the appropriate product? That will allow application developers to develop applications that will operate successfully in this environment? That will allow the end user platform vendors to incorporate the appropriate functions in the operating system stacks? That will allow ISPs to integrate vendors' productions into their operational environments? Right now it is pretty clear that what we have is a set of ideas, each of which has relative merits and disadvantages, and no real unity of purpose.

It is easy to be pessimistic at this stage, given that the real concerns here appear to be related more to the factors associated with a very large industry attempting to respond to a very challenging change in the environment in which it operates. The question here is not really whether Address Plus Port routing is technically inferior to Dual-Stack Lite, or whether Carrier-Grade NATs are technically better or worse than either of these approaches. The question here is whether this industry as a whole will be able to sustain its momentum and growth across this hiatus. And, from this perspective, I believe that such pessimism about the future of the Internet is unwarranted.

The communications industry has undergone significant technological changes over the years, and this change is one more in the sequence. Some of these transformations have been radical in their effect, such as the introduction of the telephone in the late nineteenth century, whereas others have been more subtle, such as in the introduction of digital technology to telephony in the latter part of twentieth century, replacing the earlier analogue circuit model of telephony carriage. Some changes have been associated with high levels of risk, and we have seen a myriad of smaller, more agile players enter the market to lead the change while the more risk-averse enterprises stand back. On the other hand, other changes require the leverage of economies of scale, and we have seen market consolidation behind a smaller number of highly capitalized players.

My personal opinion is that the Dual-Stack Lite approach is the best one, because it appears to be technically elegant. I suspect, however, that the lowest-common-denominator fall-back position that this somewhat conservative industry will adopt will rely strongly on Carrier-Grade NATs, and the industry is likely to eschew the more complex support mechanisms required by the various permutations of Address Plus Port routing.

Further Reading

[0] The Address Sharing BOF was held at IETF 74 in March 2009. The presentations and a summary of the session can be found as part of the proceedings of that meeting: `http://www.ietf.org/proceedings/09mar/shara.html`

[1] `http://www.ripe.net/ripe/meetings/ripe-56/presentations/Camp-IPv6_Economics_Security.pdf`

[2] Geoff Huston, "Anatomy: A Look Inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

[3] Jeff Mogul and Steve Deering, "Path MTU Discovery," RFC 1191, November 1990.

[4] `draft-ietf-softwire-dual-stack-lite-00.txt`

[5] `http://www.doxpara.com/DMK_BO2K8.ppt`

[6] `http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html`

[7] **http://en.wikipedia.org/wiki/Universal_Plug_and_Play**

[8] **http://www.ietf.org/proceedings/09mar/slides/shara-8/shara-8.htm**

[9] C. Rigney, S. Willens, A. Rubens, W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865, June 2000.

[10] Egevang, K., and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994.

[11] Srisuresh, P., and D. Gan, "Load Sharing Using IP Network Address Translation (LSNAT)," RFC 2391, August 1998.

[12] Srisuresh, P., and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999.

[13] Tsirtsis, G., and P. Srisuresh, "Network Address Translation—Protocol Translation (NAT-PT)," RFC 2776, February 2000.

[14] Hain, T., "Architectural Implications of NAT," RFC 2993, November 2000.

[15] Srisuresh, P., and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, January 2001.

[16] Holdrege, M., and P. Srisuresh, "Protocol Complications with the IP Network Address Translator," RFC 3027, January 2001.

[17] D. Senie, "Network Address Translator (NAT)-Friendly Application Design Guidelines," RFC 3235, January 2002.

[18] Srisuresh, P., J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox Communication Architecture and Framework," RFC 3303, August 2002.

[19] Daigle, L., and IAB, "IAB Considerations for Unilateral Self-Address Fixing (UNSAF) Across Network Address Translation," RFC 3424, November 2002.

[20] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN—Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, March 2003.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC. He was a member of the Internet Architecture Board (IAB) from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001.
E-mail: **gih@apnic.net**

Operational Challenges When Implementing DNSSEC

by Torbjörn Eklöv, Interlan Gefle AB, and Stephan Lagerholm, Secure64 Software Corp.

As a reader of *The Internet Protocol Journal*, you are probably familiar with the *Domain Name System* (DNS) "cache poisoning" techniques discovered a few years ago. And you have most likely heard that *Domain Name System Security Extensions* (DNSSEC)[0, 13, 14, 15] is the long-term cure. But you might not know exactly what challenges are involved with DNSSEC and what experience the early adopters have gathered and documented. Perhaps you waited with your own rollout until you could gather more documentation about operational experiences when rolling out DNSSEC.

Stephan Lagerholm and Torbjörn Eklöv are DNS architects with significant DNSSEC experience. Torbjörn lives in Sweden and has helped several municipalities, as well as other organizations, sign their zones. Stephan Lagerholm lives in Dallas, Texas, and has been involved in implementing DNSSEC at several U.S. federal agencies. This article summarizes their experiences, including lessons learned from implementing the technology in production environments, and discusses associated operational concerns.

Background

A plethora of information about DNSSEC and cache poisoning attacks is available on the Internet[16], so we will not repeat it, but we think it is important to state where DNSSEC is today.

During the last few years the number of deployments, as well as the size and importance of the signed domains, has increased significantly. One of the main reasons for adoption of the DNSSEC during the past year was that the U.S. *Office of Management and Budget* (OMB) issued a mandate requiring the signing of the `.gov` domain in the beginning of the year. U.S. federal agencies were mandated to sign their domains by the end of 2009. Some agencies have already implemented the technology, whereas others are still working on it.[1]

Acceptance of DNSSEC technology is also reaching outside of the U.S. government. *Top Level Domains* (TLDs) around the globe have announced DNSSEC initiatives. To mention a few, Afilias signed .org and Neustar recently announced signing of `.us`. Several *County Code TLDs* (ccTLDs), including `.nl` and `.de`, announced that DNSSEC implementation is a work in progress. VeriSign has announced that it is working on signing the largest TLDs, namely `.com` and `.net`. Finally, the *Internet Corporation for Assigned Names and Numbers* (ICANN) along with VeriSign released a timeline for signing the root zone. And of course, the pioneer `.se` is on its fourth year as a signed TLD.

Several vendors have released software and products to support and make the signing of zones easier. A range of different products is now available on the market.

DNS professionals now have a broad choice of technology—from collections of open-source signing scripts to advanced systems with full automation and support for *Federal Information Processing Standard* (FIPS)-certified cryptography.

Operational Challenges

DNSSEC might significantly affect operations unless it is carefully implemented because it requires some changes to the underlying DNS protocol. Those changes are, in fact, the first significant changes that have been made to the DNS protocol since it was invented. Those changes might sometimes fool old systems into believing that the packets are illegal. DNSSEC also introduces new operational tasks such as rolling the keys and resigning the zone. Such tasks must be performed at regular intervals. Furthermore, as with any new technology, there are misconceptions about how to interpret the RFC standard.

The First Bug Reported

Late summer 2007, Torbjörn Eklöv convinced the municipality of Gävle in Sweden of the benefits of DNSSEC. He proudly signed what is believed to be the first municipality zone in the world, **`gavle.se`**. At first, everything worked fine. A week or so later, Gävle received reports from citizens who could not reach the municipality's websites. It turned out that a new version of *Berkeley Internet Name Domain* (BIND) was rolled out by a large service provider and that this version of BIND introduced a rather odd bug that affected DNSSEC. The result of the bug was that home users with some home routers and firewalls could not reach any signed domains.

Some people who heard about the problem at **`gavle.se`** wrongly believed that DNSSEC caused the problem and that DNSSEC is broken. However, this assumption is not true; DNSSEC worked as expected, but a bug in a particular version of BIND caused the problem. The problem triggered some research on how home routers handle DNSSEC. *Stiftelsen för Internetinfrastruktur,* the organization that runs the **`.se`** TLD, issued a report describing how commonly used home routers and firewalls handled the new protocol changes in DNS[2]. Later, Nominet, which administers the **`.uk`** TLD, issued a similar report[3]. In addition, DENIC, which administers the **`.de`** TLD, researched the same subject[4]. The results are all discouraging; only 9 out of 38 tested home gateways supported DNSSEC correctly in the most recent reports.

A *Birds of a Feather* (BoF) session was held at the 76th meeting of the *Internet Engineering Task Force* (IETF) in Hiroshima to discuss the problems involving home gateways[5]. We look forward to seeing progress in this area.

Preparing Your Firewall for DNSSEC

Most problems with DNSSEC are related to firewalls. Make sure to involve your security and networking administrators so that they can make the required changes before taking DNSSEC into production.

Two types of firewall problems are most common:

The first involves the *Transmission Control Protocol* (TCP). There is a misconception among firewall vendors and security administrators that DNS queries use the *User Datagram Protocol* (UDP) and that zone transfers use TCP. Unfortunately, this assumption is not entirely true. DNS queries first try UDP, but revert to TCP if no response is received for the initial UDP query or if the response lacks important information because it is truncated. The possibility of something in the path blocking the response to the initial query is much higher with DNSSEC because of the increased size of the responses.

> *For DNSSEC to work correctly, it is mandatory that you open your firewall for both TCP and UDP over port 53.*

The second problem is related to the *IP Buffer Reassembly* size. The authors of the DNSSEC standard realized that a potential problem might exist with TCP queries. TCP puts a higher burden on the DNS servers. (TCP is much more expensive to process than UDP.) To avoid too much TCP traffic, the authors made the EDNS0 extension mandatory for DNSSEC. EDNS0 is one of the *Extension Mechanisms for DNS* (EDNS), a standard that, among other things, allows a client to signal that it is capable of receiving DNS replies over UDP that are larger than the previous limit of 512 bytes. Some firewalls are not aware of the fact that the EDNS0 standard allows for larger packets and they either block any DNS packet using EDNS0, or block any DNS packet larger than the 512 bytes regardless of the EDNS0 signaling.

Other firewalls allow for the large packets by default, whereas a few vendors require the firewall to be manually configured to do so. Any device in the path that does packet inspection at the application layer must be aware of the EDNS0 standard to be able to make a correct decision about whether to forward the packet or not. ICANN has summarized the status of EDN0 support in some commonly used firewalls[6].

Note that it is not enough to test that your firewall allows large incoming DNS replies by sending DNS queries to the Internet[7]. You must also test that an external source can receive large DNS replies that your DNS server is sending. One way of doing so is to use an open DNSSEC-aware resolver[8, 9].

> *Test and configure your firewall to allow for use of EDNS0 and for DNS packets larger than 512 bytes over UDP.*

Preparing Your Slaves

Setting up DNSSEC involves substantial changes to the master name server so it can sign and serve the signed data. However, it is easy to foresee that the slaves must be upgraded, too. The slaves are much easier to upgrade and operate because they never produce signatures.

They are secondary systems that transfer data from the primary server and respond to DNS queries. But the slaves must understand how to respond to queries requesting signed data.

Slaves must be upgraded to BIND 9.3 or better to understand the *Next Secure* (NSEC)[14] standard. NSEC is a method to provide authenticated denial of existence for DNS resource records. The newer *Next Secure 3* (NSEC3)[10] standard introduces some additional requirements for the slaves. If you use NSEC3, you must upgrade the slaves to BIND 9.6 or later. Version 3 of *Name Server Daemon* (NSD)[17] and any version of *Secure64 DNS Authority/Signer*[18] can do both NSEC and NSEC3. Windows Server 2008 R2 for the x86-64 architecture supports DNSSEC as a master, slave, and validating resolver. However, we recommend limiting the use of the Windows platform to slaves and for domains using NSEC. Our opinion is that it is very hard to implement DNSSEC on Windows, and we suggest that you wait until Microsoft offers a sensible *Graphical User Interface* (GUI) and support for NSEC3. Note that the Itanium version of Windows 2008 R2 supports neither DNS nor DNSSEC.

> *Make sure your slaves can handle the version of DNSSEC you intend to use.*

If the slaves are administered by another party, contact the administrator before you begin DNSSEC implementation. Make sure the slaves are running a version capable of DNSSEC. Stephan helped a large U.S. federal agency sign its domains. The agency used one of the major federal contractors to run its slave servers. After multiple attempts to reach somebody that understood DNS and DNSSEC, Stephan finally learned that the slaves were running BIND 9.2.3 and that the contractor had no plans to upgrade. The only alternative for the agency was to in-source the slaves and run them itself.

> *If your slaves are administered by another party, make sure you know if and what version of DNSSEC that party supports before you start implementing.*

Communicate with Your Parent

TLDs allow you to communicate with them in two ways:

- *Registrant–Registrar–Registry Model:* In this, the most common model, the registrant (**`example.org`**) does not communicate directly with the registry (**`.org`**). Instead, a third-party registrar handles all communication related to DNS and DNSSEC. This model is, for example, used by the **`.se`** and **`.org`** TLDs.
- *Registrant–Registry Model:* This model is normally used by smaller TLDs such as **`.gov`**. It allows direct communication between the registrant (**`agency.gov`**) and the registry (**`.gov`**). The TLD acts as both a registrar and a registry in this model.

Most problems described in the following paragraphs apply to both models, but those involving multiple registries are obviously applicable only to the Registrant–Registrar–Registry model.

Establishing a *Chain of Trust* in DNSSEC involves uploading one or more public keys to the parent. Ultimately the parent publishes a *Delegation Signer* (DS) record, a smaller fingerprint that can be constructed from the DNSKEY record. To upload your keys, you must use a registrar that supports DNSSEC. If your registrar does not support DNSSEC, you need to move your domains to another registrar (or convince your current registrar to start supporting DNSSEC). It usually takes a few days or up to a week to move a domain from one registrar to another.

> *Make sure that your registrar supports DNSSEC. If it does not, move your domain to a registrar that supports DNSSEC before you begin signing your zone.*

Some registrars allow registration under multiple TLDs. However, just because a registrar handles DNSSEC for one TLD does not mean that it handles DNSSEC for all TLDs it serves. For example, several registrars in Sweden support DNSSEC for `.se` but not for `.org` or `.us`.

> *Make sure that your registrar handles DNSSEC under the TLD in question.*

Most registrars offer you the opportunity to use their name server instead of your own. The service is either offered for free or for an additional cost. The registrar typically provides a web interface where you can change your zone data. This service is a good and useful choice if your domains are uncomplicated and small. Larger and more complex domains are better operated on your own servers.

Some registrars that provide this type of service can handle DNSSEC only if you use their name servers and not your own name servers. These registrars can establish the chain of trust with the parent only if the zone is under their control. They lack a user interface for uploading a DS key that you generate on your own name servers.

> *If you intend to use your own name servers, make sure that your registrar supports this deployment model, and allows you to upload a DS record for further distribution to the registry.*

In theory, the child zone system should create the DS record fingerprint and upload it to the parent. In practice, some registrars require you to upload the DNSKEY record to them. They then create the DS record for you. (This practice is bad because the registrar must know the hash algorithm used to construct the DS record, which it might not know.) The DNSKEY record comes in several different formats, depending on the platform you used to create the keys (BIND, Microsoft, NSD, Secure64, etc.). The formats have minor differences, and you might have to convert the DNSKEY into a format that the registrar accepts.

Not everything works smoothly, even with the correct DNSKEY format. The logic at one registrar's website was to deny uploading of DNSKEYs unless the optional *Time To Live* (TTL) field existed. (The TTL value is useless in the DNSKEY context because the parent overrides this value with its own TTL). You may have to manually change your DNSKEY before uploading it to comply with the checks that the registrar performs.

> *If your registrar requires you to upload the DNSKEY, make sure that your solution can generate the requested format. If not, you need to manually change the fields with a text editor.*

As noted previously, some registrars are performing too many checks and irrelevant checks before accepting and creating the secure delegation. Other registrars do not check at all or have limited checks that do not work as expected. For example, some registrars assume that your key is created using a certain algorithm, and they do not double-check it prior to creating a DS record. One registrar created a bogus DS record if you uploaded a DNSKEY with upper-case characters in the domain name. The bogus DS record looked valid, and troubleshooting to find this error took hours.

Another example is keys created with *Webmin*[11], a graphical tool that you can use for signing zones. Webmin defaults to using the less-common *Digital Signature Algorithm* (DSA) for its DNSKEYs. The registrar did not complain when uploading the Webmin key, and it created a bogus DS record by assuming that it was an RSA key.

It is hard for a registrant to do anything about errors at the registrars. The best you can do is to make sure that you upload the correct key with the correct parameters such as algorithm, key length, key-id, etc. If something goes wrong, you might have to change the keys in production. Rolling the keys to the same algorithm and key length is relatively easy—but changing your keys to another algorithm adds extra complexity. It is an interesting exercise to change to another algorithm in production, but it is something we recommend avoiding if possible.

> *Double-check the DNSKEY/DS so that it is created with the correct parameters prior to uploading it.*

Communicate with Your Children

If you have sub-domains in your domain, you must make sure that you can accept and publish the DS records that your children upload to you. This situation is not a problem if you use zone files in text format—you can simply insert the DS record using your favorite editor. But it might be a problem if you are using an *Internet Protocol Address Management* (IPAM) system. In that case make sure that it can insert DS records into the zones that are managed by the system. Some IPAM systems do not support insertion of DS records correctly.

> *Make sure that your IPAM system can insert DS records into your zones.*

A common strategy among organizations with high-availability requirements for their critical servers is to use a global load balancer, which is basically a DNS server that responds differently depending on the status of the service in question. For example, assume a load balancer can respond to a question for `www.example.com` with `192.0.2.1` and `192.0.2.2` if both web servers are up. If `.1` becomes unavailable, the load balancer notices a failure and responds only with `.2`. In order to use a global load balancer, you must delegate `www` as a sub-domain to its own DNS process.

When DNSSEC is implemented, you must make sure that the load balancer can handle DNSSEC (and not that many do); otherwise it is impossible to sign the responses for those resources. Unfortunately, these resources are the most critical ones for your environment and would benefit the most from DNSSEC signing.

> *Make sure that your load balancers support DNSSEC. If they do not, have an alternative strategy.*

Rolling the Keys

You should change the DNSKEYs regularly and when you think the keys are compromised. The process of doing so is called *rolling the keys*. There are normally two different keys in DNSSEC, the *Key Signing Keys* (KSKs) and the *Zone Signing Keys* (ZSKs). Rolling the ZSK is an internal process and does not require communication with the parent. Rolling the KSK, on the other hand, requires the parent to publish a new DS record.[12]

There is no standard yet that describes how the communication between the parent and the child should occur when a key is rolled. Early DNSSEC-capable registrants used a web interface that allowed their registrants to upload and manipulate the DNSSEC information. With a web interface, each domain must be handled separately and there is no easy way to automate the interaction.

The web interface works for a handful of domains but becomes very cumbersome when you have many domains. For those types of organizations, it is important to make sure that there is some kind of *Application Programming Interface* (API) or script access to the registrar. This interface allows the organization to upload new DS records during the rollover in a convenient way.

> *Make sure that your registrar supports automation through an API if you have many domains.*

Scripting with an API as described previously is one way of communicating with the registrar. Another way of achieving the same type of automation is for the parent (or registrar) to monitor the child for any changes to the DNSKEY records.

Note that the chain of trust is still intact during a nonemergency rollover. The parent can securely poll the child and grab the new DNSKEY records and convert them into DS records. The polling from the parent to each signed child needs to occur regularly so that a rollover is picked up quickly. This regularity of polling makes the scheme best for domains with fewer delegations (in the order of thousands, not millions—consider how much bandwidth an hourly polling of 15 million children would require).

Automation is a good thing, but make sure you understand the implications when opting for automatic detection of key rollovers. The automation scripts are not fail-safe. It has been reported that early versions of such scripts under some circumstances wrongly assumed that a key rollover occurred and deleted the DS record, thus breaking the chain of trust.

> *Understand the implication when opting for automatic detection, addition, and deletion of DS records.*

Management of DNSSEC

Without DNSSEC, you are not bound to any particular registrar; you can switch to a new registrar fairly easily. With DNSSEC, this situation changes. First of all, if you let the registrar sign the zone on your behalf, the registrar will be in charge of the key used to sign your zone. Extracting your key so that it can be imported to another registrar is not always straightforward (also remember that there is really no incentive for your previous registrar to help you because you just discontinued its service). An alternative is to unsign the zone before you change registrars, but that option might not always be a viable one. The lack of standards makes it hard to change registrars on a signed domain that is in production.

You must tell your new registrar that you are using DNSSEC, and you must make sure that the registrar supports it. If not, the registrar might accept the transfer but be unable to publish the DNSKEY records. The result would be a DS record published by the registry but no corresponding DNSKEY records at the child, making the zone “security lame” and causing failed validation.

The same types of problems exist if you are running your own name servers. If you change your master server, make sure that you transfer the secret keys as well. Signing with new keys will not work unless you flush out the old keys with rollovers and upload a new DS record to your parent.

> *Have a plan ready for how to transfer your keys to a new master server.*

Timers

It is important to adjust your signature validity periods and the *Start of Authority* (SOA) timers so that they match your organizational requirements and operational practices. SOAs expire and signature validity periods all too often are too short.

Unless you are restricted by guidelines saying otherwise, you should strive to set the timers reasonably high. Set the timers so that your zones can cope with an outage as long as the longest period that the system might be unattended.

For example, if you know that your top DNS administrator usually has three weeks of vacation in July, you could consider setting the times so that the zone can survive four weeks of downtime. If you are confident in your signing solution and are monitoring your signatures carefully, you might set it a little bit lower.

Signature lifetime is a trade-off between security (low signature lifetimes) and convenience (high signature lifetimes). Setting a really high signature lifetime is convenient from an operational perspective but is less secure. Some organizations such as the IETF use an excessive signature lifetime of one year (**`dig ietf.org DNSKEY +dnssec | grep RRSIG`**). This lifetime is clearly not recommended, and they should know better.

> *Carefully set your signature lifetimes and SOA times to reflect your organization's operational requirements and practices.*

A Note on Validation

This article has focused on the authoritative part of DNSSEC. That part includes signing resource records and serving DNS data. The operational challenges with signing data are much greater than the challenges of validating data. To validate data, the only thing you need to do regularly is update your trust anchor file. Make sure you do so. Torbjörn reports several outages when the **`.se`** DNSKEY used in the **`.se`** trust anchor expired in January 2010. We look forward to the work being done in this area to automate the process.

Summary

DNSSEC has been deployed and taken in production for several large and critical domains. It is not hard to implement DNSSEC, but doing so introduces some operational challenges. Those challenges exist both during the implementation phase when the zone is being signed for the first time and during the operation of the zone. Make sure you understand the possible effects of implementation and plan ahead. The following checklist summarizes the most important pitfalls with DNSSEC:

- Open your firewall for EDNS0 signaling and allow large DNS packets using UDP and TCP over port 53.
- Check the DNSSEC capabilities of all your masters and slave servers.
- Check the DNSSEC capabilities of your registrar and understand their requirements for the public key you are uploading.
- Make sure your IPAM system can handle secure delegations.

- Plan how to handle load balancers.
- Develop an automation strategy if you have a lot of zones.
- Plan how you will transfer your keys to a new master server if a disaster occurs.
- Implement a policy for DNSSEC timer settings.

Happy signing!

For Further Reading

[0] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

[1] Carolyn Duffy Marsan, "80% of Government Web Sites Miss DNS Security Deadline," *Network World,* January 21, 2010, **`http://www.networkworld.com/news/2010/012010-dns-security-deadline-missed.html`**

[2] Jaokim Ålund and Patrik Wallström, "DNSSEC—Tests of Consumer Broadband Routers," **`http://www.iis.se/docs/Routertester_en.pdf`**

[3] Ray Bellis and Lisa Phifer, "Test Report: DNSSEC Impact on Broadband Routers and Firewalls," September 2008, **`http://download.nominet.org.uk/dnssec-cpe/DNSSEC-CPE-Report.pdf`**

[4] Thorsten Dietrich, "DNSSEC-Unterstützung durch Heimrouter," **`http://www.denic.de/fileadmin/Domains/DNSSEC/DNSSEC_20100126_Dietrich.pdf`**

[5] Broadband Home Gateway BoF, **`http://tools.ietf.org/agenda/76/homegate.html`**

[6] ICANN DNS Root Server System Advisory Committee (RSSAC) and Security and Stability Advisory (SSAC), "Testing Firewalls for IPv6 and EDNS0 Support," January 2007. **`http://www.icann.org/en/committees/security/sac016.htm`**

[7] Doman Name System Operations Analysis and Research Center (OARC)'s DNS Reply Size Test Server: **`https://www.dns-oarc.net/oarc/services/replysizetest`**

[8] OARC's Open DNSSEC Validating Resolver: **`https://www.dns-oarc.net/oarc/services/odvr`**

[9] Comcast DNSSEC Information Center, **`http://www.dnssec.comcast.net/`**

[10] Torbjörn Eklöv, "DNSSEC: Will Microsoft Have Enough Time?" *CircleID,* January 2010, `http://www.circleid.com/posts/dnssec_will_microsoft_have_enough_time/`

[11] `http://www.webmin.com/`

[12] George Michaelson, Patrik Wallström, Roy Arends, and Geoff Huston, "Rolling over DNSSEC Keys," *The Internet Protocol Journal,* Volume 13, No. 1, March 2010.

[13] Roy Arends, Rob Austein, Dan Massey, Matt Larson, and Scott Rose, "DNS Security Introduction and Requirements, RFC 4033, May 2005.

[14] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose, "Resource Records for the DNS Security Extensions," RFC 4034, May 2005.

[15] Roy Arends, Rob Austein, Dan Massey, Matt Larson, and Scott Rose, "Protocol Modifications for the DNS Security Extensions," RFC 4035, May 2005.

[16] United States Computer Emergency Readiness Team (US-CERT), "Multiple DNS Implementations Vulnerable to Cache Poisoning," July 2008, `http://www.kb.cert.org/vuls/id/800113`

[17] `http://nlnetlabs.nl/projects/nsd/`

[18] `http://www.secure64.com/secure-DNS`

STEPHAN LAGERHOLM is a Senior DNS Architect with Secure64 Software Corporation—a software company offering high-performance DNS server software that makes the DNS trustworthy and secure. Secure64 DNS applications include key management and zone-signing software that make it easy to deploy DNSSEC securely and correctly as well as DNS server software that is always available. Stephan is a DNS and security expert with more than 11 years of international experience in the field. His background includes leadership positions at the largest networking and security system integrator in Scandinavia, and responsibility for designing hundreds of complex IT networks. Stephan is one of the few persons in the United States to have integrated DNSSEC into production environments. Stephan is CISSP-certified and holds a Master of Science degree in Computer Science and Mathematics from Uppsala University in Sweden. E-mail: `Stephan.Lagerholm@secure64.com`

TORBJÖRN EKLÖV is the founder and partner of Interlan Gefle AB, an IT consulting company in Sweden with 20 employees. He is a DNSSEC and IPv6 pioneer. All internal and external services at Interlan use both IPv6 and IPv4, and the company hosts about 200 DNSSEC-signed domains. Torbjörn has worked with Internet communication and security for 15 years, and is the founder and manager of Secure End User Connection (SEC), or Säker KundAnslutning (SKA) in Swedish, an organization that certifies products and broadband networks to protect subscribers from spoofing and hijacking. His favorite homepage is `http://test.ipv6.tk`. You can reach him at `Torbjorn.Eklov@interlan.se`

Book Review

The Art of Scalability

The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise, by Martin L. Abbott and Michael T. Fisher, ISBN-13: 978-0-13-703042-2, Pearson Education, 2010.

It is often claimed that the primary lesson of the Internet is one of "scaling." So the title of this book bodes well for relevance to Internet designers. A reader would likely expect discussion of hashing algorithms, fast-path coding, protocol latencies and chattiness, distributed redundancy design, and similar guidance for handling a billion users. The reader would largely be wrong, although some of the book is dedicated to technical performance. What is easily missed in the title is the word "organizations." It does not mean organization of modules. It means organizations within a *company.*

This book is very much a holistic one. It takes the painfully realistic position that well-designed protocols and software modules matter only if the company structure or team operation is tuned to growing and running a large-scale service. The book is comprehensive and primarily tailored for highly formal management, with substantial, bureaucratic procedures designed to ensure thorough consideration of scalability needs and implications. It is loaded with discussion of many different organizational and technical management tools that assist in making diligent decisions. For most readers and most companies, attempting to apply this level of formality is dramatic overkill. However, knowing about it is not.

The book is 533 pages, with 33 chapters and 3 appendices. The writing style is reasonably clean, but pedantic. Don't expect the type of entertainment-oriented writing that is common these days. The authors' experiences include *eBay* and *PayPal,* so scaling matters have been within their direct work responsibilities. As holds for any book attempting this kind of breadth, from technology design to organization management, discussion frequently is superficial and will be obvious to some readers, while the specific detail will in places be irrelevant to many others. Although these characteristics might be taken as negatives, they actually serve to demonstrate the utility of the book as an introduction and basic reference to the topic of scaling. A quick scan of the book helps the reader see how many different aspects of an organization's activities can aid or hinder large-scale operations. Exploring specific chapters can explain concepts and topics and suggest particular tools to help in planning or analysis.

Organization

Part I, "Staffing a Scalable Organization," comprises six chapters. It provides a tutorial on classic problems in structuring and staffing an organization for growth. Little is taken for granted. So there is guidance about the characteristics needed in a CEO, CFO, or CTO for aiding leadership in working to scale the company and the company's products. It even has a chapter on "Leadership 101."

For the most part, this section is likely to be useful only for readers with no management background, because the material is extremely basic. What distinguishes it is only the constant consideration of the way its topics are relevant to scaling. The likely utility of the section is in helping employees "manage up" so they can interact with management better when seeking support for changes needed to implement or maintain scalable development or operations. On the other hand, an interesting discussion explored why some simple and entirely logical choices for organizing a company work against accountability and scaling.

Part II, "Building Processes for Scale," at nearly 200 pages is 40 percent of the book. Whereas the first part concerned the people, this one concerns what they do. The first half of this part strongly emphasizes processes for anticipating and responding to scaling problems and for judiciously allocating limited resources. Hence there is even a chapter that considers "build versus buy." Technical topics discussed here are conceptual rather than concrete. They concern risk, performance, capacity, and failure recovery. Each is treated as a planning and design concern, with estimates and procedures. A warning: The word "architecture" shows up in the title of several middle paragraphs in this section, but don't be confused. It refers to groups that do architecture, not to the technical details of architecture.

Part III is "Architecting Scalable Solutions." Now at last, techies will start to get their geek fix. But perhaps with more abstraction than they will expect? Again, this book is more about properly organizing things than about algorithms. The section introduces "technology-agnostic design," with consideration of fault isolation and various growth factors, including repeated attention to cost, risk, scalability, and availability. There are chapters on database scaling and the use of caching for performance. The authors are fond of asynchronous and state-free interaction, with the view that it is more robust. The precise reason for this conclusion was not entirely clear to me, but presumably it is because it is easier to recover and retarget an exchange after an outage occurs during an interaction.

Two chapters of this part of the book are devoted to the "AKF Scale Cube," and indeed the Index has a large number of citations to it. (AKF refers to the authors' company.) For this analytic tool, the x-axis "...represents cloning of services and data with absolutely no bias." In other words, these graphs are pure replications of equivalent, parallel components or activities, used to distribute load. The y-axis "... represents a separation of work responsibility by either the type of data, the type of work performed for a transaction, or a combination of both... We often refer to these as service or resource oriented splits." The nature of the z-axis is described as "...biased most often by the requestor or customer... focused on data and actions that are unique to the person or system performing the request." I took this as meaning that the axis divides work according to tailored attributes.

Part IV is the catchall for remaining topics, with some requisite discussion of clouds and grids, application monitoring, and data center planning.

Summary

The book will be useful for architects who need to understand how to scale their own work and how to support their organization for long-term growth. It will also be useful for technical, operations, and other managers who need to understand the technical and operations scaling problems, support their own architects, and work with the rest of their organization to anticipate and satisfy scaling requirements.

—*Dave Crocker, Brandenburg InternetWorking*
dcrocker@bbiw.net

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Call for Candidates for Itojun Service Award

The *Itojun Service Award* is presented every year to an individual or a group who has made outstanding contributions in service to the IPv6 community. The deadline for nominations for this year's award is July 12, 2010. The award will be presented at the 79th meeting of the *Internet Engineering Task Force* (IETF) to be held in November 2010 in Beijing, China.

The Itojun Service Award, established by the friends of Itojun and administered by the *Internet Society* (ISOC), recognizes and commemorates the extraordinary dedication exercised by Itojun over the course of IPv6 development. The award includes a presentation crystal, a US$3,000 honorarium, and a travel grant.

The award is focused on pragmatic technical contributions, especially through development or operation, with the spirit of servicing the Internet. With respect to the spirit, the selection committee seeks contributors to the Internet as a whole; open source developers are a common example of such contributors, although this is not a requirement for expected nominees. While the committee primarily considers practical contributions such as software development or network operation, higher-level efforts that help those direct contributions will also be appreciated in this regard. The contribution should be substantial, but could be immature or ongoing; this award aims to encourage the contributors to continue their efforts, rather than just recognizing well-established work. Finally, contributions of a group of individuals will be accepted as deployment work is often done by a large project, not just a single outstanding individual.

The award is named after Dr. Jun-ichiro "Itojun" Hagino, who passed away in 2007, aged just 37. Itojun worked as a Senior Researcher at *Internet Initiative Japan Inc.* (IIJ), was a member of the board of the *Widely Integrated Distributed Environment* (WIDE) project, and from 1998 to 2006 served on the groundbreaking KAME project in Japan as the "IPv6 Samurai." He was also a member of the *Internet Architecture Board* (IAB) from 2003 to 2005.

For additional information on the award, please visit:
`http://www.isoc.org/awards/itojun/`

Less than 10% of IPv4 Addresses Remain Unallocated, says NRO

The *Number Resource Organization* (NRO), the official representative of the five *Regional Internet Registries* (RIRs) that oversee the allocation of all Internet number resources, recently announced that less than 10 percent of available IPv4 addresses remain unallocated. This small pool of existing IP addresses marks a critical moment in IPv4 address exhaustion, ultimately impacting the future network operations of all businesses and organizations around the globe.

"This is a key milestone in the growth and development of the global Internet," noted Axel Pawlik, Chairman of the NRO. "With less than 10 percent of the entire IPv4 address range still available for allocation to RIRs, it is vital that the Internet community take considered and determined action to ensure the global adoption of IPv6. The limited IPv4 addresses will not allow us enough resources to achieve the ambitions we all hold for global Internet access. The deployment of IPv6 is a key infrastructure development that will enable the network to support the billions of people and devices that will connect in the coming years," added Pawlik.

The *Internet Protocol* (IP) is a set of technical rules that defines how devices communicate over a network. There are currently two versions of IP, IPv4 and IPv6. IPv6 includes a modern numbering system that provides a much larger address pool than IPv4. With so few IPv4 addresses remaining, the NRO is urging all Internet stakeholders to take immediate action by planning for the necessary investments required to deploy IPv6.

The NRO, alongside each individual RIR, has actively promoted IPv6 deployment for several years through grassroots outreach, speaking engagements, conferences and media outreach. To date, their combined efforts have yielded positive results in the call to action for the adoption of IPv6.

Given the less than 10 percent milestone, the NRO is continuing its call for Internet stakeholders, including governments, vendors, enterprises, telecoms operators, and end users, to fulfill their roles in IPv6 adoption, specifically encouraging the following actions:

- The business sector should provide IPv6-capable services and platforms, including web hosting and equipment, ensuring accessibility for IPv6 users.
- Software and hardware vendors should implement IPv6 support in their products to guarantee they are available at production standard when needed.
- Governments should lead the way by making their own content and services available over IPv6 and encouraging IPv6 deployment efforts in their countries. IPv6 requirements in government procurement policies are critical at this time.
- Civil society, including organizations and end users, should request that all services they receive from their ISPs and vendors are IPv6-ready, to build demand and ensure competitive availability of IPv6 services in coming years.

The NRO's campaign to promote the next generation of Internet Protocol continues to positively impact the Internet community. IPv6 allocations increased by nearly 30% in 2009, as community members continued to recognize the benefits of IPv6.

"Many decision makers don't realize how many devices require IP addresses—mobile phones, laptops, servers, routers, the list goes on," said Raul Echeberria, Secretary of the NRO. "The number of available IPv4 addresses is shrinking rapidly, and if the global Internet community fails to recognize this, it will face grave consequences in the very near future. As such, the NRO is working to educate everyone, from network operators to top executives and government representatives, about the importance of IPv6 adoption," added Echeberria.

IP addresses are allocated by the *Internet Assigned Numbers Authority* (IANA), a contract operated by the *Internet Corporation for Assigned Names and Numbers* (ICANN). IANA distributes IP addresses to RIRs, who in turn issue them to users in their respective regions. "This is the time for the Internet community to act," said Rod Beckstrom, ICANN's President and Chief Executive Officer.

"For the global Internet to grow and prosper without limitation, we need to encourage the rapid widespread adoption of the IPv6 protocol," he added.

The NRO is the coordinating mechanism for the five RIRs. The RIRs—AfriNIC, APNIC, ARIN, LACNIC, and the RIPE NCC—ensure the fair and equitable distribution of Internet number resources (IPv6 and IPv4 addresses and *Autonomous System* (AS) numbers) in their respective regions. The NRO exists to protect the unallocated Internet number resource pool, foster open and consensus-based policy development, and provide a single point of contact for communication with the RIRs.

Learn more about the NRO at **`www.nro.net/media`**

The five RIRs that make up the NRO are independent, not-for-profit membership organizations that support the infrastructure of the Internet through technical coordination. The IANA allocates blocks of IP addresses and ASNs, known collectively as *Internet number resources,* to the RIRs, who then distribute them to users within their own specific service regions. Organizations that receive resources directly from RIRs include *Internet Service Providers* (ISPs), telecommunications organizations, large corporations, governments, academic institutions, and industry stakeholders, including end users. The RIR model of open, transparent participation has proven successful at responding to the rapidly changing Internet environment. Each RIR holds one or two open meetings per year, as well as facilitating online discussion by the community, to allow the open exchange of ideas from the technical community, the business sector, civil society, and government regulators.

The five RIRs are:

- AfriNIC: `http://www.afrinic.net`
- APNIC: `http://www.apnic.net`
- ARIN: `http://www.arin.net`
- LACNIC: `http://www.lacnic.net`
- RIPE NCC: `http://www.ripe.net`

ISOC Funds Projects to Support Internet Access, Security, and Policy Development

The *Internet Society* (ISOC) recently announced it is funding community-based projects around the world addressing issues such as Internet leadership, education, core infrastructure, local governance, and policy development, with a strong focus on currently underserved communities.

"The diversity of projects awarded highlights the profound importance of the Internet in so many aspects of our lives, in all parts of the world," said Jon McNerney, Chief Operating Officer of the Internet Society. "The passion and creativity of those developing the projects within their communities drives the Internet Society's commitment to help bring the benefits of the Internet to people everywhere."

As part of the ISOC *Community Grants Program,* each project will receive up to US$10,000 for efforts that promote the open development, evolution, and use of the Internet for the benefit of all people throughout the world.

Projects funded in this round include:

- Training programs to build digital literacy within safe environments in India and Uganda
- Village-operated telecommunication services in East Timor
- Support for development of core Internet time infrastructure
- Policy and practical action in Kenya to improve online safety for women
- Online support for NGOs in Tunisia and more effective local governance in India
- Promotion of Internet leadership in Ecuador
- Development of important public policy resources in Georgia and Australia

ISOC Community Grants are awarded twice each year. The next round of the program will open on September 1, 2010. Additional information about the Community Grants Program and this round of award-winning projects can be found here:

`https://www.isoc.org/isoc/chapters/projects/index.php`

`https://www.isoc.org/isoc/chapters/projects/awards.php?phase=11`

RIPE Community Statement on the Internet Address Management System

At the May 2010 *Réseaux IP Européens* (RIPE) meeting in Prague, Czech Republic, the RIPE community issued the following statement:

"The RIPE community supports all efforts to assist in the deployment of IPv6, especially in developing countries.

However, we note concerns being expressed within the ITU by a few members, most recently in the ITU IPv6 Group, that the current address management system is inadequate.

The RIPE community mandates the RIPE NCC to work with the ITU IPv6 Group, individual ITU members, and the community to clearly identify these concerns and to find ways to address them within the current IP address management system."

This statement will be sent to the *International Telecommunications Union* (ITU) to reiterate the RIPE community's belief that the current address management system works. The RIPE NCC will continue to participate actively in the ITU IPv6 Group and report back to the RIPE community.

For more information see:

`http://www.itu.int/ITU-T/othergroups/ipv6/`
`http://ripe.net/ripe/index.html`
`http://www.nro.net/documents/nro51.html`

Upcoming Events

The *North American Network Operators' Group* (NANOG) will meet in San Francisco, California, June 13–16, 2010.
See `http://nanog.org`

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Brussels, Belgium, June 20–25, 2010.
See `http://icann.org`

The *Internet Engineering Task Force* (IETF) will meet in Maastricht, The Netherlands, July 25–30, 2010 and in Beijing, China, November 7–12, 2010. See `http://www.ietf.org/`

APNIC, the *Asia Pacific Network Information Centre,* will hold its Open Policy meeting in the City of Gold Coast, Australia, August 24–28, 2010. See `http://www.apnic.net/meetings/30/`

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles (“What is...?”), as well as implementation/operation articles (“How to...”). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, “modem tax,” and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

This publication is distributed on an “as-is” basis, without warranty of any kind either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This publication could contain technical inaccuracies or typographical errors. Later issues may modify or update information provided in this issue. Neither the publisher nor any contributor shall have any liability to any person for any loss or damage caused directly or indirectly by the information contained herein.

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

The Internet Protocol Journal
Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is
published quarterly by the
Chief Technology Office,
Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com