A performance view on DNSSEC migration

Daniel Migault, Cédric Girard, Maryline Laurent

A Performance view on DNSSEC migration

Daniel Migault
Orange Labs – France
daniel.migault@orange-ftgroup.com

Cédric Girard
Orange Labs – France
cedric.girard@orange-ftgroup.com

Maryline Laurent
Institut TELECOM, TELECOM SudParis
CNRS Samovar UMR 5157
Maryline.laurent@it-sudparis.eu

***Abstract*—In July 2008, the Kaminsky attack showed that DNS is sensitive to cache poisoning, and DNSSEC is considered the long term solution to mitigate this attack. A lot of technical documents provide configuration and security guide lines to deploy DNSSEC on organization's servers. However, such documents do not provide ISP or network administrators inputs to plan or evaluate the cost of the migration.**

This paper describes current deployment of DNSSEC and provides key elements to consider when planning DNSSEC deployment. Then we focus our work on performance aspects and provide experimental measurements for both DNS and DNSSEC architecture. Experimental results evaluate the cost of DNSSEC for authoritative and recursive server with different implementations.

***Index Terms*—DNS, DNSSEC, performance, migration**

I. Introduction

DNS [21], [22] represents today's Naming System of the Internet and makes communications between names possible, and thus people to communicate though the Internet. Fully Qualified Domain Names (FQDN) are often more stable and easier to remember then IP addresses and people are more likely to deal with names than with IP addresses that define a network localization. On the other hand, DNS is not only used by end users, and the core network also uses DNS. Convergence between traditional telephone service (PSTN) and Voice over IP (VoIP) is expected to be done thanks to E.164 NUmber Mapping (ENUM) protocol which is based on the DNS [19], [12].

As a crucial element for making the Internet useable, the Internet Community is concerned about security issues on DNS. The Internet Engineering Task Force (IETF) started designing DNSSEC in January 97 [11], and a final version was issued in March 2005 [7], [9] and [8].

DNSSEC is the security extension of DNS and provides resolvers with the mechanisms to authenticate the origin of the RRset, integrity protect RRsets, build a chain of trust and prove the non-existence of the FQDN or a specific RRset. DNSSEC and DNS are compatible in the sense that a DNSSEC authoritative or resolving server can treat a DNS request. However DNSSEC comes with so many changes to the architecture, the servers and network security policies that it is better to consider it not as an extension of DNS but rather as a new protocol.

Complexity may be the major drawback of DNSSEC, and one of the main reasons for its slow adoption up to 2008, ISPs and regular firms were hardly considering DNSSEC adoption. In fact, in July 2008 Dan Kaminsky revealed a major flaw in the DNS specifications that makes it sensitive to cache poisoning attacks [17], [16]. At that time DNSSEC was considered to be the long term solution to make DNS robust to cache poisoning attacks and it almost closed the debate about whether or not DNSSEC was worth being deployed.

This paper intends to help those organizations to position themselves towards DNSSEC. At first we show WHY organizations - and ISPs - should start their DNSSEC migration. More specifically, we detail the current postion toward DNSSEC of major actors of the Internet community, and we show that DNSSEC is part of the Internet evolution. We also describe, for organizations, the benefit of migrating to DNSSEC. Then we show HOW organizations should handle DNSSEC migration. This includes considerations on how DNSSEC impacts the network as well as how platforms should be upgraded with regards to a performance point of view. Then we focus our concern on performance and consider how the DNS platform should be upgraded to DNSSEC. We present experimental measurements for various implementations to compare the cost of DNSSEC over DNS with various configurations. These are expected to help organizations define the DNSSEC architecture and the implementations that best fit their requirements as well as clarify how many servers should be added to the DNS platform, how much response time is increased, how many DNS update will we be able to...

The remainder of this paper is organized as follows. Section II presents the current position towards DNSSEC of various actors in the Internet community, as well as considerations of DNSSEC deployment. Section III positions our experimental work. Next, sections IV and V present our experiments. This includes a description of the testing environment, our methodology, DNSSEC impact on end user side with unitary naming resolution, DNSSEC impact on loaded servers considering maximum load and the response time for resolution and update operations. Section VI discusses these results and points that need to be looked at while considering DNSSEC migration.

II. DNSSEC current status

A. DNSSEC deployment

This section provides current DNSSEC deployment for the various actors of the Internet community.

1) Network Information Center (NIC): NIC are part of the most influent actors in the DNS community, and were the early adopters of DNSSEC. [26] provides Registries view on DNSSEC deployment, as well as DNSSEC deployment history. In 2010 TLD that are known to have deployed DNSSEC

are : .se, .ru, .mx, .pr, .bg, .br, .org, .cz, .gov, .na, .tm, .li, .ch, .arpa, .th, .uk, .enum, .pm, .edu, .fr, .re, .nl, .com, .net. and the trend is that most TLDs will implement DNSSEC [28]. On July $16th$, 2010, ICANN annouced that the root zone was signed.

2) Software implementers: DNS related software implementers where also heavily committed into the DNSSEC deployment. In 2010 DNSSEC is part of most DNS implementations - Internet Systems Consortium (`BIND9`), NLnetLabs (`NSD` and `UNBOUND`), Microsoft, Nominum (`ANS` and `CNS`), Secure64:. and powerdns is actively implementing DNSSEC. Administrative tools are also actively developed Opendnssec [4] is designed to manage security of Zones. Other sofware available is listed on [1] and [3].

3) OS implementers: In November 2008, Microsoft announced how DNSSEC would be supported in Windows 7 [27]. Resolvers on Windows 7 do not perform the validation by themselves. In other words, network administrators and ISPs have to deploy DNSSEC with signature check in their resolving servers. In February 2009, Microsoft implemented DNSSEC on Windows Server 2008 R2 [20].

4) ISP: Among ISPs, Comcast [2] is currently the only ISP that is publicly advocating DNSSEC adoption, and that, by the end of 2011, will sign its authoritative domains and proceed to DNSSEC validation on its resolving servers [15].

B. DNSSEC impacts on network

1) DNSSEC compliant infrastructure: First of all DNSSEC is complex and operational teams need to become familiar with that protocol. Procedures are complex and need to be adapted to the operational environment with automatic procedures. Then deploying DNSSEC requires to validate DNSSEC compatibility across all the network equipments as well as with our services.

On servers' side, Comcast reports at NANOG45 that DNSSEC increases memory footprint between 5 and 9 times for the authoritative infrastructure and that the recursive infrastructure requires additional recursive clusters. For middle boxes, like residential Internet router and SOHO firewall devices commonly used with broadband services, [10] shows that only 25% of the tested boxes were fully DNSSEC compliant.

On the user point of view DNSSEC resolution on small devices may slow down web surfing and [18] shows that DNSSEC may not be compatible with the DNS redirect service provided by ISP.

2) Monitoring DNSSEC: DNSSEC adds security to the traditional DNS service. However, DNSSEC also brings its own issues that makes resolution impossible. One common reason is that DNSSEC packets are larger than regular DNS packets, and thus may be dropped by network devices. Resolvers advertise through the EDNS0 option [13] a larger MTU than the traditional DNS 512 bytes MTU. If the indicated MTU is larger than the one accepted by the network for an end-to-end connectivity, then we have to try with smaller MTU. This operation is called the Path MTU walk (PMTU) [6]. [25] monitors DNSSEC zones and traffic and shows that roughly 20% of the monitored zones suffer availability dispersion, and that PMTU walk is necessary for roughly 95% of the DNSSEC zones for 1.5% of the time. Finally, [25] - maybe not any longer up-to-date - shows that in 2008 97% of the DNSSEC zones were isolated, and thus not verifiable, 9% of the authentication chain were broken, and 19% of the zones had data that are still valid according to their signature expiration date, but that do not longer exist in the zone file.

C. ISP toward DNSSEC

1) Attitude toward DNS: ISPs aims at providing Internet connectivity and services to end users. DNS is only one component to provide this connectivity. Until now DNS architectures for authoritative and resolving servers were quite scalable, performed well, and were considered as an operational issue rather than a research issue. This at least explains why they were not that involved at the beginning of DNSSEC deployment and why DNSSEC seems new to them.

2) Cache Poisoning: With the Kaminsky Attack in July 2008, people become aware that their DNS architecture is sensitive to DNS cache poisoning. On the other hand ISP providing email facilities are confronted to the reality of phishing and pharming issues [23] and DNS cache poisoning is one vector for such attacks. The AntiPhishing Working Group (APWG) shows that brand name hijacked is still an increasing issue, [24] reports with concrete examples how valuable are FQDN for companies, and the case of the cache poisoning attack against the Brazilian bank Bradesco [5] in April 2009 shows that cache poisoning attacks are part of the reality. As a result DNSSEC is required to protect companies brand, to protect Internet Services – ISP don't want for example their end user's email being redirected nor to provide corrupted DNS resolution with corrupted cache. As Chris Griffiths from Comcast reports "*Current recursive infrastructure is not vulnerable but we cannot sit back and wait for the next big bug/exploit.*" [14].

3) Position toward DNSSEC: ISP's position toward DNSSEC is balanced between the cost of DNSSEC migration and the impact of not upgrading their Naming System to DNSSEC. Costs for DNSSEC migration are high for organizations, since it impacts operational infrastructure, platform and network performances. However DNSSEC is being deployed by NIC, governemental institutions, OS implementers, and end users ask for more security. As such, DNSSEC is part of the Internet evolution. Delaying its migration may only makes the cost higher in the future. In fact, today DNSSEC traffic is quite low, and is expected to increase with DNSEC deployment of major TLDs, end users OS, organizations... Increase of DNSSEC traffic will make the migration harder, and costs higher. On the other hand not migrating to DNSSEC means that we keep our organization as and unsecure island on the Internet. This includes preventing end user from securing their naming resolution, accepting that end user private data may be redirected to an attacker web site, accepting that our domain name may be hijacked and our services unavailable.

4) ISP's DNSSEC architecture: Migration to DNSSEC can be done in various way for resolving servers. With current DNS configuration, resolving servers only perform DNS resolution. A first DNSSEC configuration can make them perform DNSSEC without validation when requested by the end user. Then this configuration can be extended to all incoming DNS queries. Finally servers can be set to proceed to DNSSEC validation. This paper intends to provide input to evaluate the cost of each configuration.

III. Position of our work

Our work differs from previous work in that we studied the impact of DNSSEC on authoritative servers, resolving servers and resolvers. Perfomance tests are performed on different implementations, with the DNSSEC NSEC3 option that was not available at the time of previous studies.

IV. Testing environment

In this paper, we consider BIND $9.6.0 - P1$, UNBOUND 1.2.1 and NSD 3.2.1. Other DNSSEC implementations were available such as Microsoft DNS, power DNS, Simple DNS plus, Secure64 and Nominum. They were not considered because DNSSEC-NSEC3 was partially implemented (power DNS), they required specific hardware (Secure64), they were not able to work on a Linux platform (Microsoft DNS), or we did not get the binaries. NSD - authoritative server - and UNBOUND -recursive server- are both developed by the NLnet Labs whereas BIND9 is developed by the ISC. Next BIND version, v10, will also be split into different pieces of code for the authoritative and recursive server, which is expected to improve its performance. BIND and NSD have distinct designs. BIND loads its zone file whereas NSD compiles it so that any possible query is handled.

A. Testing environment

For our tests we used Intel Pentium III (@ 1GHz 32 bits) CPU, 384MB of RAM for servers with Debian 5.0 (lenny), Linux kernel 2.6.24. To load the servers we used an Intel Xeon E5420 (Quad-Core @ 2.5GHz 32bits) CPU, 3GB RAM with Ubuntu 8.10 (hardy) 32 bits version with Linux kernel 2.6.27. The tested BIND version was multithreaded, but with one CPU we used one thread. The testing environment was designed to measure DNS / DNSSEC performance for resolving and authoritative servers as represented in figure 2. Time was measured using `Wireshark`. *Client Processing Time* is the time to initiate the query, forge the datagram, and send it to the outbound network interface, as well as the time to receive the response from the inbound interface back to the software. *ISP Network Latency* and *Internet Network Latency* are the time datagrams are on the wired network. *Cache processing Time* is the time a query is received on the inbound interface, processed, and a resolution is handled plus the time to forward the response from Internet interface to the client interface. *Authoritative Processing Time* is the time authoritative servers take to receive a query and send the response.
The data used for the tests were directly hosted on the authoritative server, we did not consider any hierarchy in our Naming architecture, and the naming space was quite flat. Authoritative server signes its zone with a single key, trusted by the resolving server.
In this paper, we consider that authoritative servers can be *DNS* or *DNSSEC* whereas resolving servers can be *DNS*, *DNSSEC* or *DNSSEC with validation*. By default the *DNSSEC* configuration considers that the resolver proceeds to a resolution which involves the additional DNSSEC fields, and sends those to the client, but does not proceed to signature checks.

B. Testing tools

Tests were performed with different tools. `dnsperf` and `resperf`, developped by Nominum have been used to send requests or updates to DNS servers. Performance measurements have been made with `collectl` and network latency measurements with `Wireshark`.

C. Testing methodology

For our different tests, we used the median instead of the mean value. As illustrated in figure 1 which gives the distribution of measurements for a specific test, the median is more representative of the data. However when getting results with `dnsperf`, the returned value is the mean.

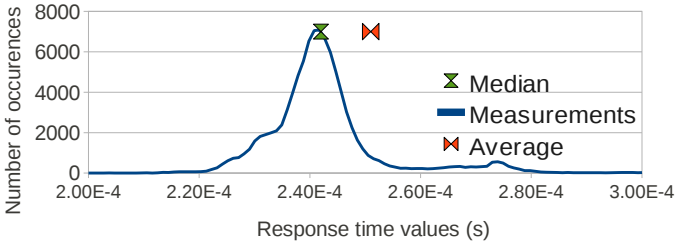
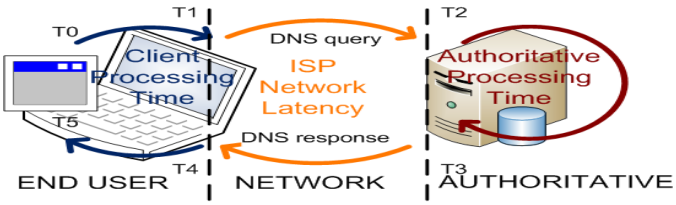


Fig. 1. Measurement distribution for a specific test

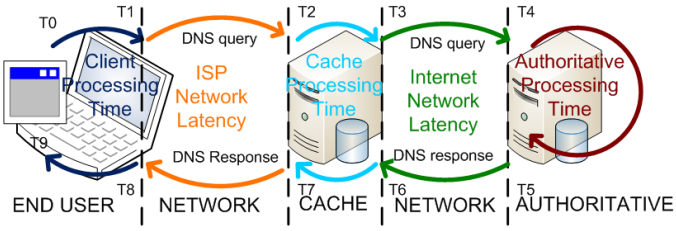
V. Experimental Work

A. Unitary Tests

Unitary tests measure the system performance without load considerations. For authoritative servers (figure 3a), implementation comparison provides that NSD always has better performance over BIND – 60% for DNS and 65% for DNSSEC. NSD also has lower network latency than BIND for DNS and DNSSEC – 8% for DNS and 7% for DNSSEC.
Protocol comparison shows that NSD is less impacted than BIND by DNSSEC – 8% for NSD and 25% for BIND. Network latency also increases by 60% with DNSSEC.
For resolving servers (figure 3b), the implementation comparison shows that UNBOUND lowers BIND performance by 67% for DNS, by 68% for DNSSEC without validation and by 46% for DNSSEC with validation. Migrating from DNS to DNSSEC with no validation adds an extra time of 9% for UNBOUND and 14% for BIND. On the other hand, migrating from DNS to DNSSEC with validation adds an extra 253%



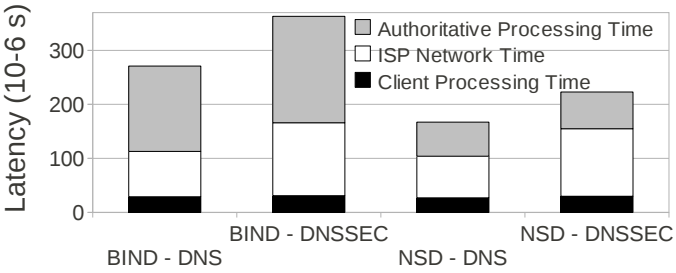
(a) Authoritative



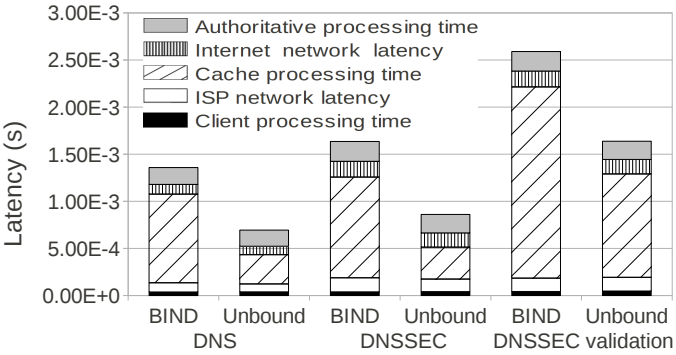
(b) Cache

Fig. 2. Testing environment and time consideration

for UNBOUND and an extra 116% for BIND.
According to unitary tests, NSD is much more efficient than BIND. This can be partly explained by lighter source code for NSD and by the difference of their architecture. BIND10 should enhance its performances by splitting the code for authoritative and resolving servers.



(a) Authoritative



(b) Cache

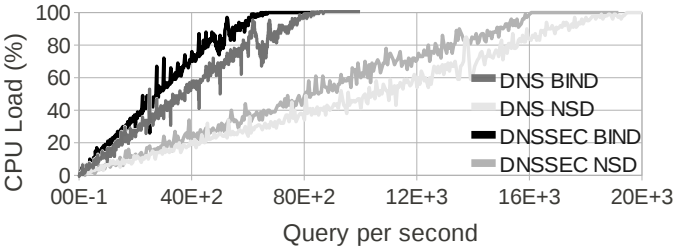
Fig. 3. Unitary test: latency

B. Maximum Load

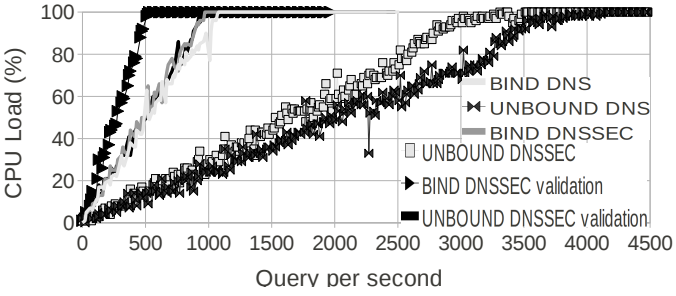
Figure 4 shows the CPU load of authoritative and resolving servers. For authoritative servers, considering the maximum load q_{max}, comparison of the different implementations shows that with DNS the maximum load handled by BIND corresponds to 43% of the maximum load handled by NSD. With DNSSEC the maximum load handled by BIND corresponds to 41% of NSD's maximum load. In other words, with the tested configuration NSD is able to deal with around 2.3 times more traffic than BIND with DNS or DNSSEC.
We also measured the cost for DNSSEC migration for each implementation. The maximum load of with DNSSEC corresponds to 79% of the maximum load with DNS with BIND and 83% with NSD. In other words, with both implementation BIND and NSD, the costs of DNSSEC is estimated roughly at 30% of the DNS traffic.
For resolving servers, with DNS, the maximum query load handled by BIND corresponds to 28% of UNBOUND's maximum query load. With DNSSEC, resolving servers can proceed to a signature check (DNSSEC validation) or not. With DNSSEC without validation, the maximum load handled by BIND corresponds to 29% of UNBOUND's maximum load. With DNSSEC with validation, the maximum load handled by BIND corresponds to 55% of UNBOUND's maximum load. In other words, with validation UNBOUND is able to deal with around 3.4 times more traffic than BIND. With DNSSEC and validation UNBOUND deals with 1.8 times more traffic than BIND. Validation lowers the differences between BIND and UNBOUND. A possible explanation is that signature check is costly, and has equivalent performance on both implementations.
While comparing DNSSEC cost for a given implementation, we can see that BIND with DNSSEC (without validation) the maximum traffic load (without validation) corresponds to 90% of the maximum load with DNS. For BIND and DNSSEC with validation the maximum load corresponds to 49% of the maximum load with DNS. For UNBOUND with DNSSEC without validation, the maximum load corresponds to 86% of the maximum load with DNS. With DNSSEC with validation, the maximum load corresponds to 25% of the maximum load with DNS. In other words the cost of DNSSEC without validation represents between approximately 10% and 14% of the DNS traffic for both implementations. When validation is involved, the cost varies from 75% and 51% of the DNS traffic. The cost of DNSSEC with resolving server varies more across implementation then it does with authoritative servers. BIND has lower performance then UNBOUND, but seems less impacted then UNBOUND by DNSSEC.

C. Network Latency & Response Time

Response Time directly impacts the end user. Figure 5 provides the server processing time of resolving and authoritative servers regarding the load. For authoritative servers and load below 40%, the response time is quite constant, and NSD response time is around 50% of BIND's response time with DNS and 45% with DNSSEC. Migration to DNSSEC increases response time of 20% for NSD and 10% for BIND. For resolving servers, and CPU time lower than 50%, the



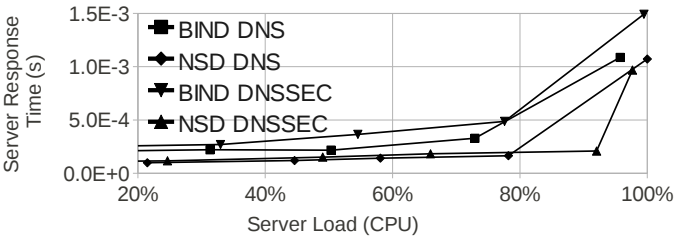
(a) Authoritative



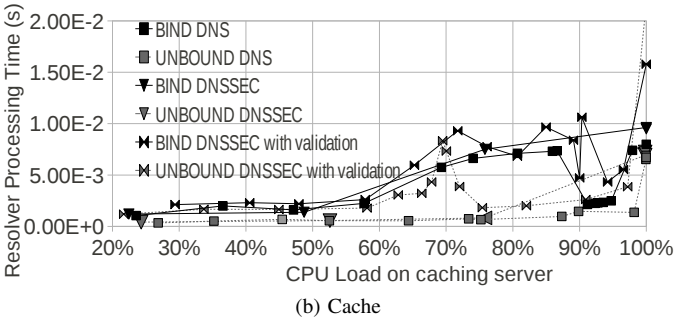
(b) Cache

Fig. 4. CPU load for authoritative and resolving server

response time is quite stable. UNBOUND response time is around 35% of BIND's response time for DNS, 30% for DNSSEC and 75% for DNSSEC with validation. Migration from DNS to DNSSEC, either with BIND or UNBOUND, does not significantly change latency. With validation, migration increases response time by 35% for BIND and 215% for UNBOUND, compared to DNS.



(a) Authoritative



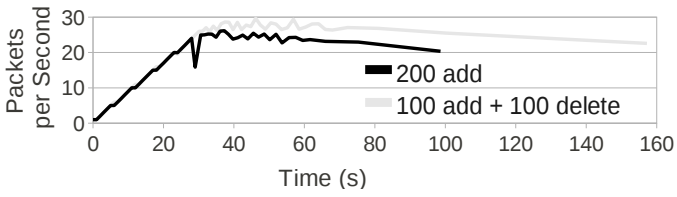
(b) Cache

Fig. 5. Response Time

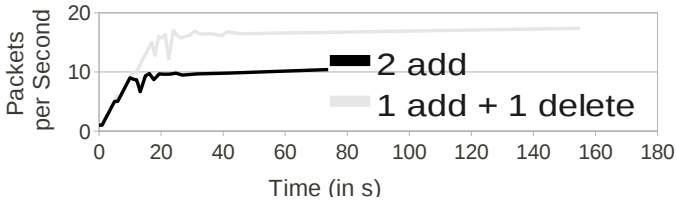
D. Update Operation Cost

Updates are performed using the `nsupdate` command on BIND only (NSD does not treat dynamic updates). Possible operations are : `add` or `delete`. We first compare costs of `add` and `delete` operations. Since `delete` must follow an `add`, to compare the respective cost of those operations, we actually compared $2.n(\texttt{add})$ and $n(\texttt{add} + \texttt{delete})$ operations. Figure 6 shows that $t_{add}^{DNS} = 1.75\,ms$ and $t_{delete}^{DNS} = 0.5\,ms$, so `delete` requires 3.5 more time. With DNSSEC $t_{add}^{DNS} = 116.8\,ms$ and $t_{delete}^{DNS} = 168\,ms$, so `delete` costs 1.43 more time. DNSSEC cost makes `add` operation 66 times longer and `delete` operation 335 longer.

Tests are performed for one operation, but `nsupdate` can perform multiple operations at a time. Figure 7 shows sending multiple updates is more efficient, both with DNS and DNSSEC.



(a) DNS



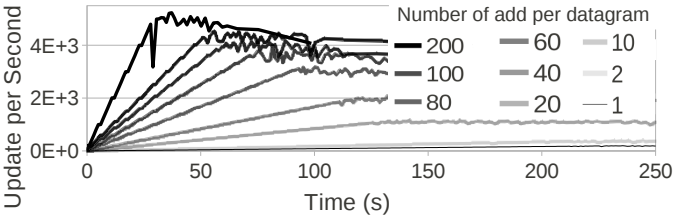
(b) DNSSEC

Fig. 6. Update rate with different actions

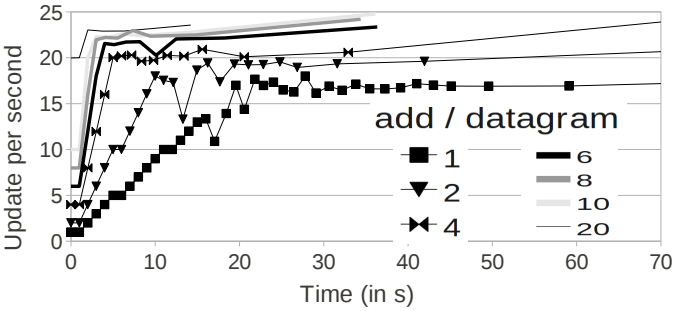
E. Impact of Cache Hit Rate

Resolving servers have caches and proceed to a resolution only when a cache miss occurs, and all previous tests considers a Cache Hit Rate (CHR) of 0%. To measure the CHR impact on resolving servers, we generate traffic with different CHR and for each traffic figure the CPU time as a function of the query rate q. Then from the various curves, we computed the Added Query Ratio (AQR) $AQR(CHR) = \frac{q_{CHR}^{CPU} - q_{CHR=0}^{CPU}}{q_{CHR=0}^{CPU}}$.

To generate a DNS traffic with a given CHR we consider two lists of FQDNs : $FQDN_l$ list with long TTL and $FQDN_s$ list with short TTL, then we load $FQDN_l$ and generate DNS traffic from the two lists as follows : $CHR\,FQDN_l + (1 - CHR)\,FQDN_s$. Figure 8 plots results for a CPU time fixed to 100% and shows that CHR is a major parameter on DNS platform performance. As expected, the more CPU time is required for a resolution, the more the CHR enhances performance. As a result, with $CHR = 100\%$, DNSSEC_VAL has an AQR that varies from 1149% to 1779%. DNSSEC and DNS has an AQR varying



(a) DNS



(b) DNSSEC

Fig. 7. Update rate with different packet size

from 374% to 592%.

For a given implementation, the AQR has similar values with DNS and DNSSEC without validation. Although implementations have different performances with DNS and DNSSEC, the CHR impacts those performance in a similar manner. Comparison across the different implementations shows that UNBOUND has a greater AQR than BIND with DNSSEC_VAL. With DNS and DNSSEC BIND has a greater AQR.

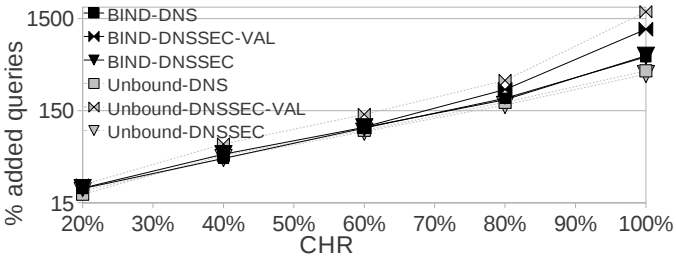


Fig. 8. Cache hit rate influence

VI. Conclusion

DNSSEC is deployed to make the Internet more reliable. The road to DNSSEC is still long and ISPs as well as other networks administrators will have to dive soon into it. At first one must be aware that DNSSEC is not a trivial option. People must plan this migration and consider DNSSEC as a new protocol with its own issues, its own engineering rules... rather then an option of DNS. However DNS is still compliant to DNSSEC which ease the transition, and migration should be much faster than IPv4 to IPv6 transition. Then people should not underestimate the change on the operational procedures. This includes, the signing procedures for authoritative servers, but also monitoring both traffic and deployed DNSSEC zones - at least in the beginning, so to avoid false positive. Then, DNSSEC deployment on resolving infrastructure should be done step by step, and opt-in trial is probably the most relevant thing to start with. At last, however difficult DNSSEC migration is now, DNSSEC is on its way to be deployed and migration will become even harder in the future.

We would like to express our sincere thanks to Francis Dupont (ISC), Stéphane Bortzmeyer and Mohsen Souissi (AFNIC).

References

[1] Dnssec deployment. http://www.dnssec-deployment.org/wiki/index.php/Tools_and_Resources.
[2] Dnssec information center. Comcast.net. http://www.dnssec.comcast.net.
[3] Dnssec.net. http://www.dnssec.net/.
[4] Opendnssec. http://www.opendnssec.org/.
[5] Cache-poisoning attack sends top brazilian bank users to scam sites. CyberInsecure.com, apr 2009.
[6] Dnssec reply size test server. Domain Name System Operations Analysis and Research Center (DNS-OARC), jul 2009. https://www.dns-oarc.net/oarc/services/replysizetest.
[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, Mar. 2005.
[8] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, Mar. 2005.
[9] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, Mar. 2005.
[10] R. Bellis and L. Phifer. Dnssec impact on broadband routers and firewalls. Test Report. Nominet, sep 2008.
[11] D. Eastlake, 3rd, and C. Kaufman. RFC 2065: Domain name system security extensions, Jan. 1997.
[12] P. Faltstrom. E.164 number and DNS. RFC 2916, Sept. 2000.
[13] P. Faltstrom and M. Mealling. The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM). RFC 3761, Apr. 2004.
[14] C. Griffiths. Comcast dnssec trail test bed. North American Network Operator' Group (NANOG45), jan 2009.
[15] C. Griffiths. Comcast voices : Dnssec. ComcastVoices, 23 feb. http://blog.comcast.com/2010/02/dnssec.html.
[16] D. Kaminsky. Details. DoxPara Research, July 2008.
[17] D. Kaminsky. Its the end of the cache as we know it. or: 64k should be good enough for anyone. IOActive, July 2008.
[18] J. Livingood, T. Creighton, C. Griffiths, and R. Webe. Recommended Configuration and Use of DNS Redirect by Service Providers, jul 2009.
[19] M. Mealling and R. Daniel. The Naming Authority Pointer (NAPTR) DNS Resource Record. RFC 2915, Sept. 2000.
[20] Microsoft. Domain name system security extensions. Microsoft, Feb. 2009.
[21] P. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Standard), Nov. 1987.
[22] P. Mockapetris. Domain names - implementation and specification. RFC 1035 (Standard), Nov. 1987.
[23] G. Ollmann. The pharming guide : Understanding and preventing dns-related attacks by phishers. NGSSoftware Insight Security Research, aug 2005.
[24] G. Ollmann. Measures to protect domain registration services against exploitation or misuse. ICANN SSAC, aug 2009.
[25] E. Osterweil, M. Ryan, D. Massey, and L. Zhang. Quantifying the operational status of the dnssec deployment. Internet Measurement Conference, oct 2008.
[26] W. Rickard. The Long Road to DNSSEC Deployment. volume 5 of *IETF Journal*. Internet Society, Internet Society, Sept. 2009.
[27] S. Seshadri. Dnssec on windows 7 dns client. Nov. 2008.
[28] P. Wouters. World wide dnssec deployment. Xelerance, 2010. http://www.xelerance.com/dnssec/.