Studying Global DNS Performance for ENUM Protocol *

Sandoche BALAKRICHENAN, Thomas BUGNAZET, Monique BECKER
CNRS/SAMOVAR
National Institute of Telecommunications
9, rue Charles Fourier, 91011 Evry Cedex, France
sandoche.balakrichenan@int-evry.fr, thomas.bugnazet@int-evry.fr, monique.becker@int-evry.fr

Abstract

ENUM (Electronic NUMbering) is a new protocol designed to translate numbers into information by using the DNS (Domain Name System) architecture. Its success will depend on whether the DNS can achieve a performance similar to the database used in the current classical voice services. The paper[1] shows different causes influencing the retrieval time of a World Wide Object. Of all the different factors, the DRD (DNS Resolution Delay) is the only one metric that cannot benefit from the lower Round Trip Time and loss rate in current networks. Another article[2] shows the different factors causing this DRD. In order to study their influences we measure and model each of these parameters. We had to develop tools (active measurement) to measure the performance of a local DNS server and an Internet link. We analyzed and modeled the data sets we obtained. By this process parameters of the mathematical model were calculated. We then created a simulation model which enables us to simulate the global DRD for ENUM in different scenario by varying each parameter one by one. This will lead to recommendations for the new ENUM Protocol to achieve the best performance.

First, we will present an overview of the ENUM protocol/technology. Then we will explain why DRD matters for the success of ENUM and the parameters that can decrease the DNS performance. In the third section we will briefly present the methods we used to measure both the local performance of a DNS server and the performance of a link. Before concluding, in the fourth section we will explain the modules we have developed for the NS-2 simulator in order to study the global performance of the DNS with ENUM.

Keywords : DNS, ENUM, Hidden Markov Models, Performance, Measurements, Simulation

1 Introduction

ENUM Protocol is a mean of making a Telecommunications Network interoperable with Internet. It's a combination of IP(Internet Protocol) based technologies designed to make the global PSTN (Public Switched Telecommunication Networks) telephone numbers known as "E.164" identifiers in the domain names. By implementing ENUM, communication service providers can leverage the existing base of PSTN telephone number users while leveraging the cost benefits and service possibilities of Packet Switched IP networks. Linking voice users with IP based services will accelerate network convergence and adoption of new services integrating voice and data.

ENUM relies on the DNS to distribute data about subscriber services. But ENUM has unique requirements for the DNS infrastructure, partly because of the volume and type of data, and partly due to service level expectation of customers used to PSTN performance. When deploying ENUM we need DNS services capable of delivering scalable performance, availability, reliability and security that is currently available to classical voice services. Whether ENUM will succeed also depends on the ability of the DNS to give response times similar to the ones given by real time databases used in the telecom world.

1.1 ENUM Protocol

ENUM[3][4], is best described as a protocol and a database enabling a service which translates an E.164 PSTN number to an URI (Uniform Resource Identifier). E.164 is an international numbering plan for public telephone systems in which each assigned number contains a CC (Country Code) , a NDC (National Destination Code) , and a SN (Subscriber Number). There can be up to 15 digits in an E.164 number. The E.164 plan was originally

*This work was supported by a French National Research Project : RNRT Numerobis

developed by the ITU (International Telecommunication Union). An URI can point to various types of resources such as mail address (mailto : user@int-evry.fr), a webpage(www.int-evry.fr) or a SIP address (SIP:user@int-evry.fr). The purpose of ENUM protocol is that if a person's telephone number is known (i.e the E.164 number), it is possible to look up different ways in which we can reach this person by a priority indicated by him . The result of an ENUM query, is a series of DNS Naming Authority Pointer (NAPTR) Resource Records (RRs)[5], which can be used to contact a resource (URI) associated with that number.

1.2 Retrieving ENUM Information

ENUM query is processed as follows. An ENUM compliant client device such as phone or VoIP (Voice Over IP) PBX (Private Branch eXchange), translates the E.164 (PSTN) number into a domain name by reversing the digits and putting periods between each of them. The phone number +33 641000001 becomes a FQDN (Fully Qualified Domain Name) 1.0.0.0.0.0.1.4.6.3.3.e164.arpa. by this process. The client device queries a *Cache Name Server* with the transformed telephone number asking for all NAPTR resource records associated with the telephone number. If the caching server doesn't have the data locally it will find and ask an *Authoritative Name Server* for the NAPTR records in the appropriate ENUM domain, returning the information to the client device. This is exactly similiar to the classical DNS delegation. For a deeper understanding readers are referred to this book [7].

The authoritative name server contains the ENUM data, or referrals to other name servers which contain the ENUM data. In Public ENUM, the E.164.arpa contains referrals pointing to other authoritative name servers which serves the E.164 CC (e.g. "3.3" for France or "4.4" in the United Kingdom). How domain delegation occurs within the country code is the responsibility of the organisation to which the country code delegated is in charge of. Within the DNS, a set of NAPTR RRs contains information about services available to that number. For e.g. in VOIP the information might include the address of SIP or H.323 service. On the client device an *ENUM Resolver* is for processing the NAPTR RRs and using the information in the DNS record to establish the appropriate communication mode .

Figure1 explains the architecture of a French ENUM database. The ENUM architecture is distributed and multiple tiers are responsible for different parts of the DNS tree. In this tree, Tier0 corresponds to the base of the inverted tree that forms the Internet domain name space designated for ENUM, i.e. .e164.arpa.. Tier1 is a level below Tier0 and corresponds to the E.164 CC, i.e. [CC].e164.arpa. (For France it will look like 3.3.e164.arpa.). The next level, Tier2 is managed by the telecom operator as they rent chunk of numbers to give them to their clients. (e.g. numbers from +33-1-60-76-00-00 to +33-1-60-76-99-99 are assigned to one operator and numbers from +33-6-41-00-00-00 to +33-6-41-99-99-99 are assigned to another operator). This level is divided into two parts the Tier2chunk which handles delegation for chunk of numbers and the Tier2number which contains the real information linked to a phone number (NAPTR RRs).

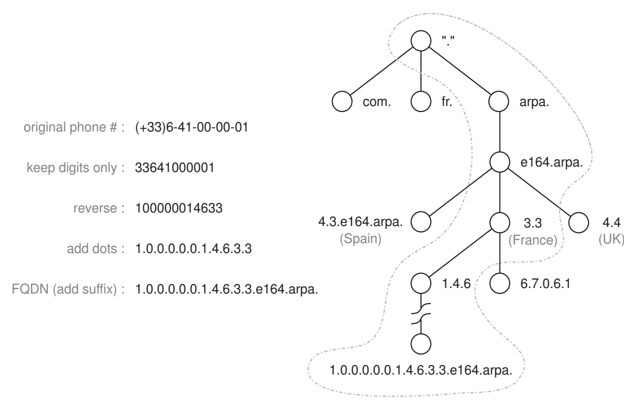

Figure 1: The ENUM Tree

2 Importance of Studying DNS Performance

2.1 In Classical DNS

For an End user launching a DNS query, only the Global DRD matters. DRD is the sum of all local response times, the latencies generated by the link and network equipments and retransmission, when a packet is lost. [1] and [2] gives typical cumulative frequency distribution of the DRD which shows small steps that highlight the impact of packet loss. In [2] Nigel Walker gives the parameters impacting the DRD by decreasing order of importance.

- Timeouts (loss in the network and in the servers). Most DNS traffic is in UDP, which is connectionless, the only way to detect a loss is to set timeouts.
- Numbers of Hops (numbers of servers contacted to have the answer)
  1. for a *hit* i.e. the cache server has the information
  2. or more for a *miss* if the cache server has to contact other authoritative nameserver(s) to get the answer from. The probability to have a hit (more efficient) depends on the TTL (Time To Live) value of the information asked and its popularity among the pool of users of the DNS Cache (One says that to improve performance a cache server must have at least 20 users). The number of delegations can increase with the depth of the DNS tree hence the number of hops.
- Service time in network and servers. routers, firewall etc..
- Speed of light (geographical distance)

In [1] the authors show the impact of the DRD in the retrieval time of a World Wide Web object. The tool *webperf* from Telcordia breaks it down into four parts :

- The DRD = time of a dig() command.
- The TCP Connection Delay (TCD) = three way handshaking which is a bad overhead for a small file
- The Wait for the First Bit (WFB) = one Round Time Trip (RTT) and the web server computation delay
- The Download Time (DT) = TCP Slow-Start penalises small file.

All of these parts benefit from lower RTT and loss rate in networks but it seems that the DRD percentage is increasing. For a full QoS, one must take care of this issue and analyse reasons for the different factors impacting DRD . The deepness of the DNS Tree, the TTL and popularity distribution are responsible for the number of hops.

2.2 The ENUM Case

The ENUM DNS Tree seems deeper than the classical DNS tree but the number of hops can be kept low if TTL values are well chosen and the number of delegations should not be too high either. The distribution of the dialed numbers may be significantly different from a Zipf Law. The use of DNSSec is also nearly mandatory for integrity reasons.

3 Modelling

3.1 Local Performance of a DNS Server[9]

3.1.1 Our Study

We have explained in section(2.1) the different parameters(timeouts, popularity, service time, and distance) affecting DRD. In order to study these parameters we required a tool. *QueryPerf* is a tool used by DNS Administrators to measure maximum load of a DNS Server. It is designed to work at the limit(both Hardware (HW) and Software (SW)) of the DNS server. In QueryPerf, if the load becomes too high, then there will be a loss reducing the load for a time to make the system stable.

Our requirement is a tool that enables us to send a given load on to a server so that we can explore the behaviour of a local DNS server from a small rate to a high query rate. We managed to do so by altering the code of *queryperf*. The new tool does not check for timeouts anymore. Its only purpose is to stress the local DNS Server. We used this tool to stress the server at different rates from low (1000 -2000 queries per second(qps)) to high (50000 - 80000(qps)).

While this load program was running we sampled the performance (loss rate and response time) of the target DNS server by sending monitored queries to it. We re-conducted this experiment for different values of query rate thus ending with the local behaviour of the DNS Server for a large load spectrum. For each query rate we measured the minimum and maximum response time, its average value $\overline{X} = \frac{1}{N}\Sigma_{i=1}^{N} X_i$, Standard Deviation $\sigma = \sqrt{\frac{\sum_{i=1}^{N}(X-\overline{X})^2}{N}}$ as well as the loss rate. The gaussian distribution of the response time population was also checked with the computation of the *skewness* $g1 \simeq 0$ and *kurtosis* $g2 \simeq 0$

$$g1 = \frac{m_3{}^{(\frac{3}{2})}}{m_2} \quad (1)$$

$$g2 = \frac{m_4}{m_2^2} - 3 \quad (2)$$

$$where \quad m_i = \Sigma_{i=1}^{N} \frac{(Xi - \overline{X})^i}{N} \quad (3)$$

Thus we got real data results from these experiments[9].

3.1.2 Our Model

At a given load the behaviour of a local server can be evaluated by three metrics, namely the loss rate, the standard deviation and the average of the response time as the response time distribution is a gaussian (a different one for any given query rate). Each of these parameters increases with the query rate and we found that the best way to describe their evolution over the load is to use a mathematical function defined in two parts as shown in figure 2. The first part of the function shows an exponential growth $f(x) = e^{\alpha x}$ whereas the second part is presented in the form of a linear growth. $f'(x)= a * x + b$. For each experiment we calculated the three parameters (α , a and b) to fit best the dataset measures. So that in the end, for each experiment the local DNS performance is evaluated by only 9 parameters : three sets of parameters for the three metrics. We are aware that the parameters found are not

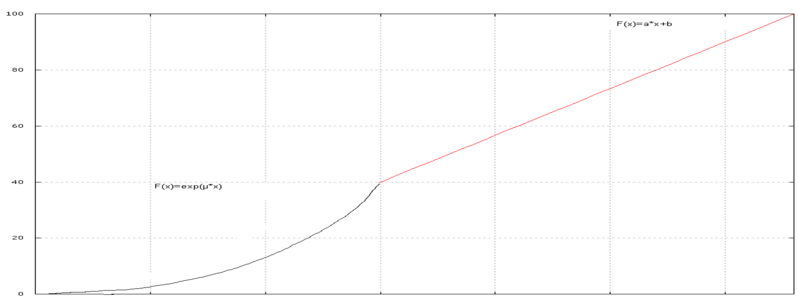

Figure 2: Mathematical Function Model to Fit the Charts

generic and depends largely on the type of HW used, the SW and its configuration especially the size and type of the DNS database and the activation or not of DNSSec. So for a thorough analyse, we need to vary many things. In the other hand we think that our disposable dataset and model are sufficient for simulation purpose.

3.2 HMM (Hidden Markov Model) Links

3.2.1 Our Study

Similar to Internet users QoS also affects DNS traffic. Active measurements, even though it's intrusive (one sends packets into the network) is the best technique to know the QoS of a link because one can monitor the effect of the network upon a small flow of packets along its route. Active measurements lead to a black box model that some refer to as an Internet cloud. Since we only need such a model for simulation purpose we'll stay at this layer, but our model can also be explained as a constructive model (white box) where all the parameters have a meaning. We decided to focus on the two most important metrics for the DNS traffic, namely the one way delay (OWD) and the loss rate.

So we designed a set of tools to get trace files to analyse. A sender/sniffer program located on the source host sends UDP packet of small size each 20 ms(by average) and records their departure timestamp, the interdeparture times being exponential. At the destination host the receiver also captures the arrival time. We used the libpcap and libnet libraries for the measures to be more accurate. The comparison of the two traces leads to a list of bits (1 if the packet is lost else it is 0) and a list of values for the delay. Since the two hosts were not synchronised (with GPS for instance, NTP being not accurate enough) we had to use the Convex Hull method and add half the minimum RTT as a post mortem treatment to resynchronise the traces.

3.2.2 Our Model

It appears that HMM is the best model for our data since they are dynamic ; Weibull distribution and simple loss rate cannot show the network state change over time. Since Internet links are assymetric (due to the client/server nature of the Internet), and since the loss rate and delay are not correlated (loss appears when queues are full : delay there is more or less constant) we need four HMMs to model a single link : one delay HMM and one loss rate HMM for each direction.

For all HMMs, the transitions from one state to another are exponentially distributed (by averaging the sample rate) along time. Each state of a loss HMM has a loss rate value. Each state of the delay HMM has two values the average and the standard deviation delay as the delay in a given state is modelled by a truncated gaussian (the minimum value is half the minimum RTT).

In order to find the parameters (number of state, routing table) that fit the model to the different stochastic process we used the EM (Expectation-Maximisation) algorithm. This HMM models are used as Internet cloud links which can be seen in figure 4.

4 Simulation

4.1 Usage of Simulator

It is difficult to test different real delegation models, because it involves changing the contents of the databases in the servers that forms the Zone which is explained in this book[7]. With a simulator the configuration files of the Zone and also all the data relative to the structure of the DNS tree are concentrated in one machine.

In real environment the bandwidth can vary among different systems connected to the DNS. Using a simulator it is possible to test with different bandwidths according to our convenience. Also in the same manner a simulation model enables to multiply the number of servers without any additional cost. By using the model we construct and observe the behavior of a real phenomenon. This is how we can evaluate the performance of complex systems. This model enables to study the existing as well as non existing systems.

Nevertheless it is necessary to validate the simulator and the model used and compare the simulation results got from the simulation with the measures obtained from measuring the real architecture. The results of the simulation model are compared with the measured results. If the results are not close enough, then the simulator is refined until the simulation results converge close to the measured results. Until then an iterative process is followed to compare and refine the model as shown in the figure 3. Finally as the simulation results are close enough to the real measurements, the simulation model is validated and can be used to test other scenarios.

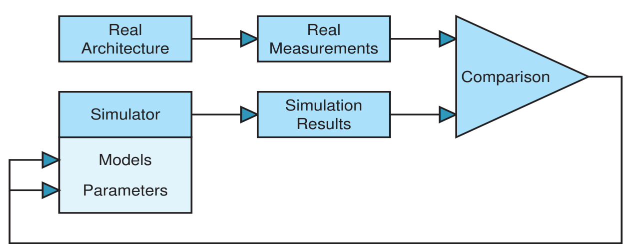

Figure 3: Simulation Model Verification

There were no existing simulator softwares which had support for the DNS architecture. We chose NS-2 Simulator and added the modules required for our model on it.

4.2 Modules Developed

NS-2 architecture involves two programming languages. One is C++ and another one is OTCL. The script written in OTCL permits to control the simulation. The usage of C++ is particularly recommended in modules which involve analysis of packet. We have four modules developed in C++ and one module in Tcl.

The first module *client* is used for traffic generation. It is very complex to model the user behavior and it is quite possible that the distribution chosen for traffic generation in the simulation model does not quite adhere to the real scenario of traffic generated. This module is designed to generate ENUM queries with chosen frequencies and given popularity distribution among telephone numbers.

The second module is the *resolver* module. This module receives the generated traffic from the clients and forwards this query to the cache server and waits for the response. It is an iterative query from the resolver to the

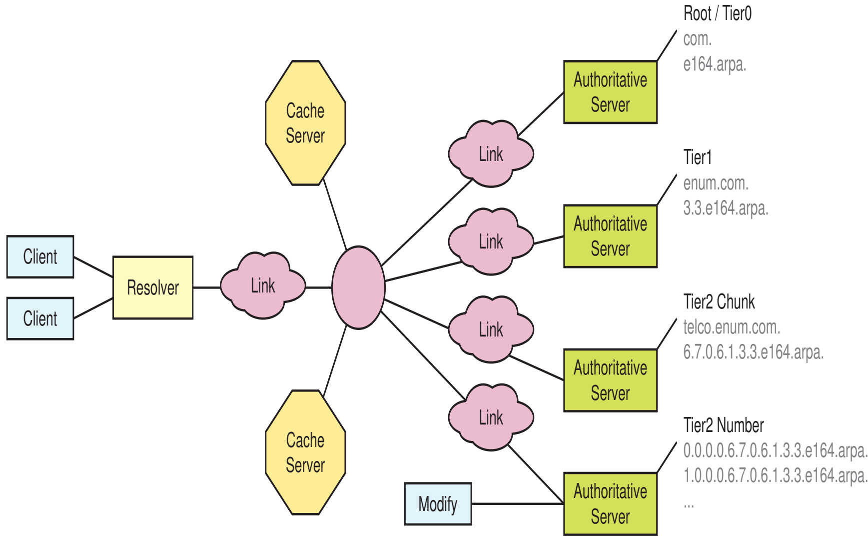

Figure 4: Simulation Model

cache server. The burden of finding an answer to the query is placed on the cache server. Adding more than one resolver is quite easy with the help of the simulation model. The resolver and the client facilitates to load the cache server on a wide range.

Most of the DNS Resolution is processed by the *cache server* module. It receives recursive queries from the resolver. If the information is present in its database there's a cache hit and it can answer directly to the resolver ; otherwise in case of a cache miss it has to send iterative querie(s) to authoritative server(s) before it can answer. Since it only knows the name and place of the root-server(s) in the beginning, it has to send quite a few iterative queries until it finds the response for the resolver's query (high number of hops). But this process is shortened (fewer hops) when the cache get populated over time as data is stored in the cache database for the TTL period. Readers can read this book [7] to have the whole overview.

The *server module* receives the recursive query from the cache server and searches for the query in its hash table. If it has the response it sends back, otherwise it responds with information about the name server which can possibly have information about the query. The *internet cloud* in figure 4 is developed on the basis the HMM model which is explained in section 3.2

5 Conclusion

The results obtained i.e parameters from real measurements and the model can be used as an entry to run simulations even though we only did few experiments. Those simulations can help to improve the QoS perceived by the end user by isolating the impact of each parameter (delay and loss in the network server, composition of the DNS tree). As far as we can see the load is no more a problem since current HW and SW configuration can handle a high rate of queries and DNS managers can still use load balancing techniques. The issue is to reduce the global DRD, especially for a new protocol such as ENUM.This performance study can also help the classical DNS as studies [6][8] highlight the bottleneck DNS is becoming. Another concern is to keep the DNS working even with all the new extensions arising [8].

References

[1] Craig E. Wills and Hao Shang, *The Contribution of DNS Lookup Costs to Web Object Retrieval* wills00contribution(WPI-CS-TR-00-12) http://citeseer.ist.psu.edu/wills00contribution.html

[2] Nigel Walker *Name resolution and performance of DNS* In the Proceedings of Multi-Service Networks ,1999

[3] P. Faltstrom *E.164 number and DNS* RFC 2916 ,Sep 2000

[4] P. Faltstrom *The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)* RFC 2916 ,April 2004

[5] M. Mealling, R. Daniel *The Naming Authority Pointer (NAPTR) DNS Resource Record* RFC 2915 ,Sep 2000

[6] C. Huitema and S. Weerahandi, *Internet Measurements: the Rising Tide and the DNS Snag* In Proceedings of the 13th ITC Specialist Seminar on Internet Traffic Measuremnet and Modelling, 2000.

[7] C.Liu and P.Albitz, *DNS and BIND*, 4th edition. oreilly, April 2001.

[8] Randy Bush, *The DNS today: Are we overloading the saddlebags on an old horse?* 49th IETF Meeting Plenary Presentation, December 2000

[9] Thomas Bugnazet, Sandoche BALAKRICHENAN , Monique Becker, *A New Approach to Model and Measure Local DNS Server Performance* Third EuroNGI Workshop on New Trends in Modelling, Quantitative Methods and Measurements , June 22-23, 2006, Torino, Italy.