Studying ENUM Performance with Modeling and simulation*

Sandoche BALAKRICHENAN, Thomas BUGNAZET, Monique BECKER
CNRS / SAMOVAR
Institut National des Télécommunications
9, rue Charles Fourrier
91000 Evry, France

ABSTRACT

The new IETF protocol ENUM uses the DNS to bind existing phone numbers into set of information's (email address, SIP phone number etc.). Success of ENUM will depend on whether the DNS can achieve a performance similar to the database used in classical voice services. In order to study the DNS performance we designed a new way to measure and model a single DNS server behavior. We also modeled and measured the IP links connecting the DNS devices: Resolver, Cache server and Authoritative servers. Finally we created a simulation model which enables us to simulate ENUM traffic. We then use the parameters obtained from the previous two models as input values into this simulator. The numerical results obtained from the simulation, was compared to real measurements in order to validate the global model. We plan to use the simulator to study different scenarios by varying different parameters. This will lead to recommendations of the new ENUM protocol to achieve the best performance.

1. Introduction

ENUM Protocol (RFC 2916) is a mean of making telecommunication network interoperable with the Internet. By implementing ENUM, communication providers can leverage 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 uses Domain Name System (DNS) service to map the E.164 identifiers into domain names. Normally the DNS does a database look up and translates a Uniform Resource Locator (URL) (such as http://www.int-evry.fr) into IP address (157.159.11.8). In the ENUM case it acts as an overlay and aggregates more information (in addition to the URL) such as telephone number, fax number, instant messaging address etc. into the DNS. With DNS service, ENUM based applications can use a single phone number to contact a persons fax, email SIP phone etc.

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

There has been number of research done on studying DNS performance [1] [2]. According to our knowledge we are the first one to study DNS performance using ENUM architecture. In this paper we draw attention to why it is important to study DNS performance for the success of ENUM.

An overview of ENUM is presented in section 2. Why it is important to study the response time impact on DNS resolution is explained in section 3. Measurements are made on a local DNS server and a model is evolved from these measurements in section 4. Similarly from measurements a model is evolved for the Quality of Service (QoS) of IP links connecting the different DNS devices in section 5. In section 6 a simulation model is developed and parameters obtained from the models of section 4 and section 5 is used as input in the simulation model to study the global performance of the DNS server. Finally the conclusion is given in section 7.

2. Background

ENUM is best described as a protocol and a database which translates an E.164 PSTN number to an URI.

E.164 is an international numbering plan for public telephone systems in which each assigned number contains a Country Code (CC), a National Destination Code (NDC) and a Subscriber Number (SN). For example in France a fixed number according to E.164 numbering plan is as follows: "33-1-60760000". "33" is the CC. "1" is the NDC for Paris area and "60760000" is the SN. There can be up to 15 digits in an E.164 number.

An URI is a sequence of characters which make it possible to identify resources such as a document, image, file, database, email address or other resource or service presenting a common format. An URI can point to various type of resources such as mail address (mail to: user@int-evry.fr) a web page (www.int-evry.fr) or a SIP address (SIP:user@sip.int-evry.fr).

*This work was sponsored by French National Project: RNRT Numerobis

Proceedings of the First Asia International Conference on Modelling & Simulation (AMS'07)
0-7695-2845-7/07 $20.00 © 2007 IEEE

2.1. An Example of ENUM

Let's say user “X” internet telephone services are mapped to the E.164 address “+33-1-60760000 “. When user “Y” wants to call “X”, he just has to call X telephone number (i.e. the E.164 number). The telephone network routes the call to the Internet Gateway that is the nominated service agent for this E.164 number. The internet gateway takes the call set up request with “X” number and first reverses the digits, then inserts a “.” between each digit and finally appends “e164.arpa” at the end. The resultant DNS string is the Fully Qualified Domain Name (FQDN) 0.0.0.0.6.7.0.6.1.3.3.e164.arpa. This name is then passed as a query to the DNS, to retrieve all associated Naming Authority Pointer (NAPTR) DNS Resource Records (RRs). The URI RRs used by ENUM are NAPTR records (RFC 2915). Let's suppose user “X” has the following entries into the DNS.

IN NAPTR 100 10 “U” “E2U+SIP” “!^.*$!sip:user@sip.int-evry.fr!”
IN NAPTR 100 10 “U” “mailto+E2U” “!^.*$!mailto:user@int-evry.fr!”

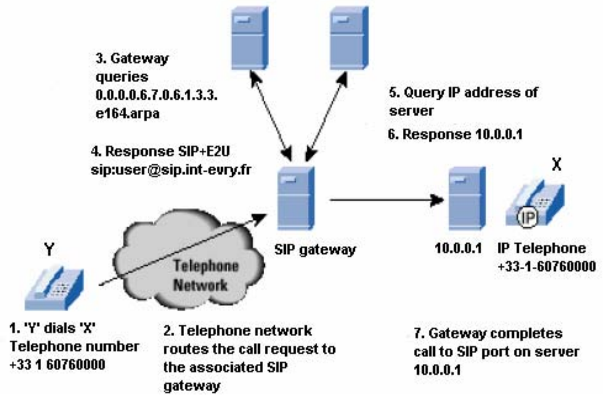

3. Gateway queries 0.0.0.0.6.7.0.6.1.3.3. e164.arpa
4. Response SIP+E2U sip:user@sip.int-evry.fr
5. Query IP address of server
6. Response 10.0.0.1
X
Y
Telephone Network
SIP gateway
10.0.0.1
IP Telephone +33-1-60760000
IP
1. 'Y' dials 'X' Telephone number +33 1 60760000
2. Telephone network routes the call request to the associated SIP gateway
7. Gateway completes call to SIP port on server 10.0.0.1

Fig. 1. Example of using ENUM to call an IP Phone

In this case there are two DNS entries which use an order value of 100 and a preference of 10. Since both of the entries have the same preference the first entry is taken. The "u" flag indicates that the rule is terminal and that the specified URI is to be used. The service field specifies that the SIP protocol is to be used, in conjunction with the E.164 to URI (E2U) resolution service (RFC 2543). The operation of the regular expression produces the URI of the form sip:user@sip.int-evry.fr. For this call request, the gateway picks the sip+E2U service and performs the associated regular expression transform using the original E.164 number and the regular expression. This produces the sip: URI. The gateway then uses the DNS a second time to translate the domain part of the URI, sip.int-evry.fr, into an IP address using a DNS A record. The gateway then opens up a session with UDP port 5060 on this SIP server to complete the call setup, requesting a voice session with the user “X” on this server (Fig. 1).

2.2. ENUM Architecture

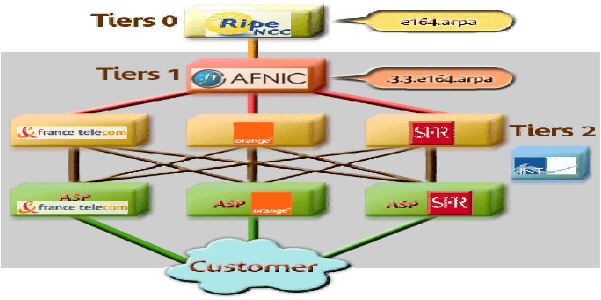


Fig. 2. The French ENUM Architecture

To align the assignment path of E.164 numbers with the delegation procedures in DNS, the ENUM tiered architecture (Fig. 2) has been created:

- The ENUM Tier 0 corresponds to the base of the internet domain space that is designated for ENUM (i.e. e164.arpa). Records at this level contain pointers to the ENUM Tier 1 for an E.164 Country Code or portion thereof.
- The ENUM Tier 1 is the level in the tiered architecture corresponding to the E.164 Country Code (CC), i.e. <CC>.e164.arpa. Records at this level contain pointers to the ENUM Tier 2 for a full E.164 number.
- The ENUM Tier2 is the level in the tiered architecture corresponding to the E.164 number, i.e., <Subscriber Number>.<CC>.e164.arpa. Records at this level contain NAPTR records for a full E.164 number.

3. Importance of studying the performance of DNS

For ENUM to be successful it is important that the global response time should be in correspondence with the response time in the telecom world. The equipments used in the telephone world are used to work with the real time databases. But in a classical DNS request via the Internet, the response time is not short as the ones compared with the real time databases. PSTN call establishment is expected to take less than 200 milliseconds [3]. So this must be the upper bound of ENUM based look ups regardless of the amount of data the DNS server is offering or query load the DNS server is experiencing.

The total response time (also known as DNS Resolution Delay (DRD)) of traditional architecture DNS is limited by the following reasons (by decreasing order of importance) [4]:

- Consecutive retransmissions with losses on the network (timeout) on the server level
- Number of hops caused by following factors:

  - Structure of the DNS tree (height)
  - Popularity of the name (Influence in the cache) and the associated Time To Live (TTL).
- The delay caused by following factors:
  - RTT in the network
  - Service time of the server
- Speed of the light (Geographical Distance)

It is worth examining to identify the cause of the DRD increase, for improving the response time. In the ENUM case, the ENUM tree is most of the times deeper than the classical DNS tree, but the *number of hops* (which is one of the factors influencing DRD) can be kept low if TTL values are well chosen. The distribution of E.164 numbers may be different from domain name distribution which follows a Zipf Law [1]. In classical DNS, a small number of domain names are very popular and even small degree of cache sharing can take advantage of this. This method may not be effective in ENUM case. In both the cases (Either classical DNS or ENUM) it is important to study how parameters such as timeouts, number of hops, popularity can be varied to reduce the DRD.

4. Local performance of a DNS server

Initially we studied the behavior of a single server i.e. the *local DNS server*. For this we designed a new way to measure and model the performance of this local DNS server.

The first tool that we have developed for this test is to create the addresses of ENUM format which looks as below.

```
[…]
8.5.5.4.6.7.0.6.1.3.3. e164.arpa.    NAPTR
1.8.7.4.6.7.0.6.1.3.3. e164.arpa.    NAPTR
[…]
```

The file containing this address forms as an entry for the configuration file of a tool called Queryperf which in turn uses these parameters to send as DNS requests to a local DNS server.

For years DNS administrators have been using the tool *Queryperf*, given with the Bind DNS Software distribution, to measure the maximum load of their DNS Server. Basically, the Queryperf program will maintain a fixed number of queries (default value is 20) running on a target DNS Server (the asked FQDN and type of RR are taken randomly from a config file). Whenever a query takes more than the timeout value (default is 5 seconds) to complete, the query is considered lost and a new one is issued .At the end this tool gathers statistics and among them the number of queries per second (qps) metric.

Queryperf tool is designed to work at the limit (both Software and Hardware) of the DNS Server, if the qps load becomes too high then there will be a loss reducing the load for a time. As a result the server can't be tested on a large range.

Since Queryperf has a limited test load range and we wanted to study the behavior of a local DNS server on a wider range, we altered the source code of Queryperf. The recompiled program enables us to choose the qps load (any range) to stress the local DNS server with. This changed version of Queryperf no more checks for timeouts, its only purpose is to stress the local DNS Server. We used it to load the server at different low (1,000-20,000 qps), medium and high (50,000-80,000 qps) rates.

4.1. Methodology

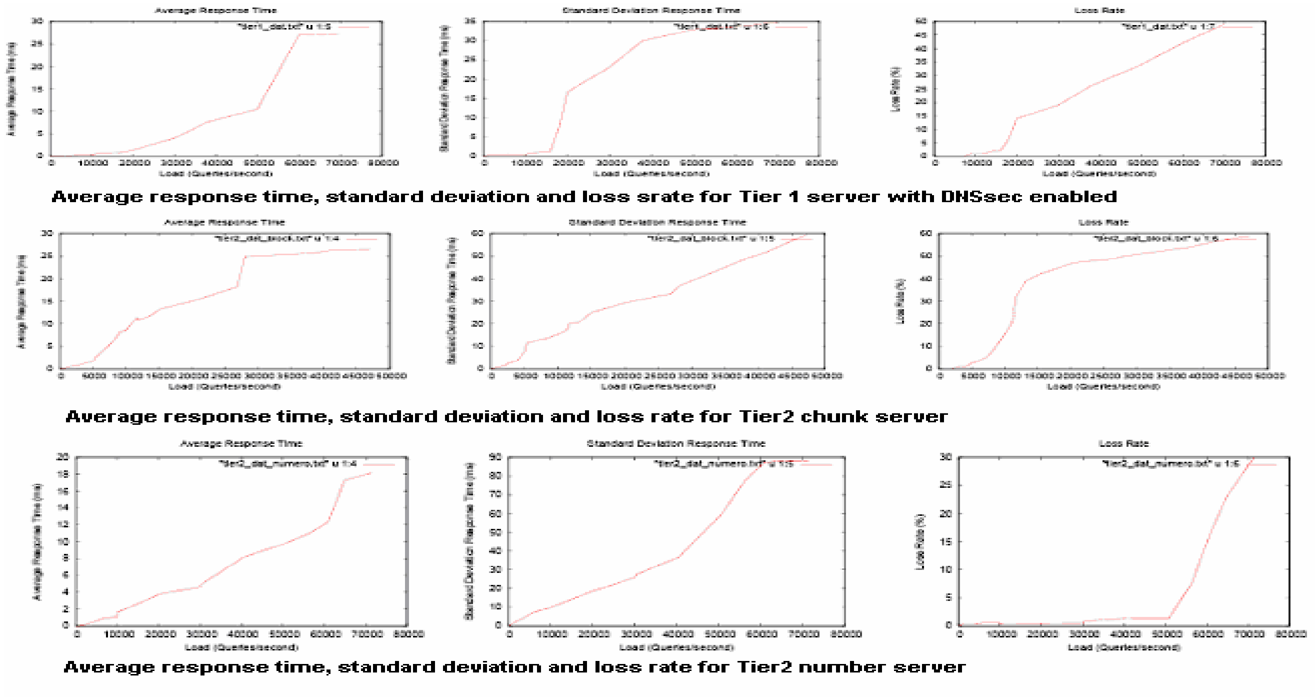
We used the tools that we explained in the previous section to generate valid ENUM addresses and to stress the local DNS server. While the stress program was running and sending queries to the local DNS server at a given rate, we also logged the performance (loss rate and response time) of the target DNS servers by sending monitored queries (issued with a script using the “dig” command at a lower rate and watched by a libcap software). These measurements were made on three types of target DNS servers (In order to resemble the French ENUM database Fig. 2).

The DNS Security Extensions (DNSsec) function is activated for Tier1 server. DNSsec is a method by which DNS servers can verify that DNS data is coming from the correct place, and that the response is unadulterated. DNSsec are used in order to maintain the integrity and authoritative nature of the ENUM zone files.

We conducted this experiment using different qps values. For each qps value we measured the average response time, its Standard Deviation response time as well as the loss rate for the target servers (Fig. 3).

To model the real measurements we first look at the loss metric. Loss is observed beyond a certain number of requests (i.e. Load) which could be taken as the threshold value. This threshold value depends on number of parameters like the hardware configurations of the server as well as the software used (like BIND or any other DNS software). The loss rate can be modeled by a function depending on the load directed at the server. From the real measurements (Fig. 3) one notes that this function can be defined by two parts. The first part resembles an exponential growth $f(x) = e^{\alpha x}$ and the second part resembles a linear growth $f'(x) = a * x + b$ as shown in Fig. 4. The exponential growth was approximated on the loss rate for Tier 2 chunk server as shown in figure 5. Fig. 6 shows how the linear curve (for loss rate) was approximated for Tier1, Tier2 chunk and Tier2 number server.

From Fig. 3, we can see the response time slope evolves in



Average response time, standard deviation and loss srate for Tier 1 server with DNSsec enabled

Average response time, standard deviation and loss rate for Tier2 chunk server

Average response time, standard deviation and loss rate for Tier2 number server

Fig. 3. Measurements made on the local DNS server

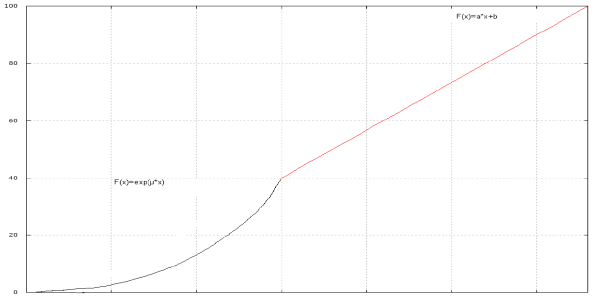


Fig. 4. Mathematical Function to fit the charts

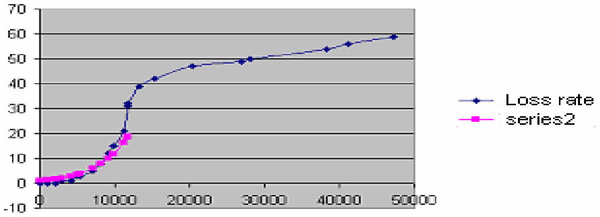


Fig. 5. Approximation of loss rate for Tier2 chunk server for exponential growth $f(x) = e^{\alpha x}$

the same manner as the loss rate except that the linear function is much lower than the slope of the loss rate. In the similar manner as loss rate, the standard deviation response time and average response time was approximated.

For each experiment we calculated the three parameters (α, a, and b) to best fit the data measures. So that in the end, for each experiment the local DNS performance is given by only 9 parameters: three set of parameters for the three metrics (average response time, standard deviation response time and the loss rate).

This model characterizes the whole behavior of a local DNS server under a wide range of load. The values of the parameters of this model depends on the type of hardware used, the embedded software, the size and type of the DNS database, the type of queries the DNS server receive, the activation or not of DNSSec.

5. IP link performance

The performance of the IP links between DNS servers can affect the global DRD (section 3). We decided to measure and model the two most important metrics which are the delay and the loss on these links.

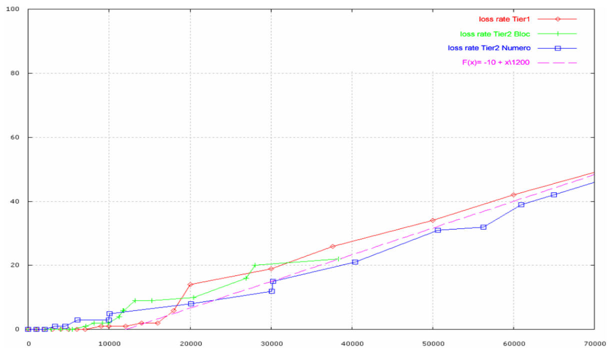


Fig. 6. Approximation of loss rate for all the servers for the linear curve ($f'(x) = a * x + b$)

5.1. Measurement

Since Internet data transfer is of asymmetric nature, the actual one way delay (OWD) cannot be estimated accurately from RTT (by dividing it by 2). So we decided to measure the OWD by sending probes packet for each 50ms on the link (active measurement).

Each packet that is sent is numbered so we can check for loss. The packets are time-stamped at both the ends. This leads to two stochastic process: one for loss and one for the delay (arrival time – departure time). Libnet library was used to ensure the accuracy of inter-departure distribution and libcap to ensure the accuracy of the time stamp. Since the two hosts were not synchronized accurately (since NTP is not accurate enough) the convex hull technique [5] [6] and the half the minimum RTT (during the measurements we also calculated the RTT between the two hosts) was used to resynchronize the traces.

5.2. Model

We wanted the model to reflect the features of an IP link. So we designed an asymmetric Internet cloud model (client-server nature of the Internet), where the delay and loss are correlated. Since we want the internet cloud to reflect the dynamicity and long-term dependence of the internet traffic we decided to use Hidden Markov Models (HMMs). HMMs are a great trade-off between simplicity and realism (Poisson modeling is easy to deal with, but not enough realistic, on the other hand fractals or wavelet analysis are an excess of what is required). Our model of an IP link has four HMMs, one for the delay and one for loss on both direction of the IP link.

To validate the model, we ran simulation of the HMM driven process and applied the methodology (Expected – Maximization algorithm). The parameters of the HMM found through this methodology were very close to the measurements data, thus validating the model.

6. Simulation

In real time it is difficult to test various models of DNS delegation because in doing so there is a necessity of modifying all the configuration files of the servers used. Also the solution which consists in using a simulator is interesting in the direction where the whole of the data relating to the structure of tree DNS is concentrated on only one machine.

The large size of the band-width access to the Internet used in various machines deployed within the framework of the project does not make it possible to test architecture with certain loads. The use of a simulator makes it possible to mitigate this limitation. Same manner, it is also possible to multiply the number of servers without cost overload.

Nevertheless it is advisable to validate the simulator and the models used, by comparing the results resulting from simulations with the measurements taken on a real architecture. Once the simulator is validated, it can be used to test in a simpler and more controlled way different scenarios.

According to our knowledge, there were no existing simulator software's which had support for the DNS architecture. Modules required for the simulation of our experiments were developed and integrated with NS-2 [7]

6.1. Implementation of the simulation model

The topology of our simulation environment is as shown in the fig. 7 .

The *client* module is used for traffic generation. From a configuration file a sequence of numbers of E.164 type is generated by the client which is sent to the resolver at a Random interval. It is very complex to model the user behavior and it is quite possible that the distribution chosen for traffic generation in the simulated model does not quite adhere to the real scenario of traffic generated.

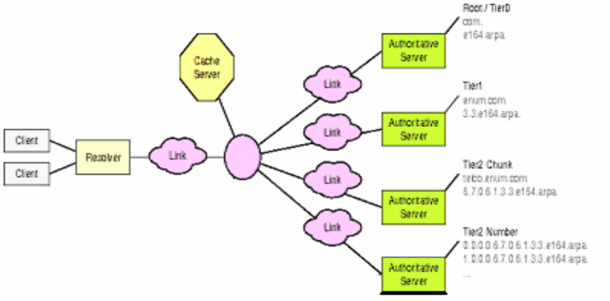


Fig. 7. Simulation Topology

The *resolver* module receives request from the client. It creates a packet type DNS query with the data obtained from the client. The resolver makes a recursive query to the cache server (the cache server is identified from resolv.conf configuration file). The burden of finding the answer to the query from the resolver 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 facilitate 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 the database then 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 to the resolver. Since it knows only the name and place of the root server(s) in the beginning, it has to send a few iterative queries until it finds the response for the resolve's query (high number of hops). But the process is shortened (fewer hops) when the cache get populated over time as data is stored in the cache database for a TTL period. For a deeper understanding of DNS cache server concepts we refer to book [8].

The *authoritative* server module receives the query from the cache server and searches for the query in its hash table. If it has the response it replies, otherwise it responds with information about the name server which can possibly have information about the query.

The internet cloud in the Fig. 8 is developed on the basis of the HMM model to model the loss and delay of the IP link. Explanation of this module is given in the section 5.

6.2. Model verification

In order to validate the simulator we compared the cumulative frequency distribution of real measurements with the simulated results. The simualtion script was written so that all parameters (size of the DNS database , query rate, performance of IP link (delay and loss) performance of local DNS server (loss rate, the average response time and standard deviation) reflect what we measured previously on the real architecture(section 4 & 5).

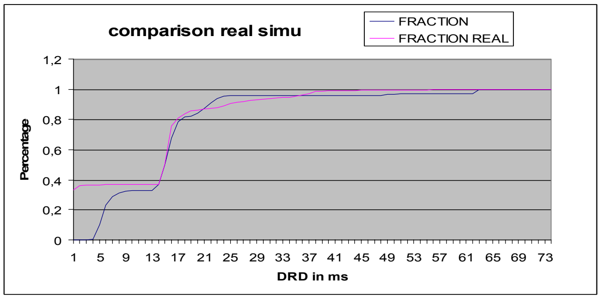


Fig. 8. Comparison of real & simulation results

The results are for first 5000 queries both in the real measurements and for the simulation model and their comparison is shown in Fig. 8.

7. Conclusion

We modeled and measured the performance of a local DNS server and also the QoS perceived on an IP link with HMM. We developed a simulation model for the DNS architecture and used the parameters obtained from real measurements in this model. To validate the simulation model we compare the real measurements and simulated results. These simulations can help to improve the QoS perceived by the ENUM end user by isolating the impact of each parameter (delay and loss in the network server, composition of the DNS tree etc.). We plan to use this simulator to study different scenarios which will lead to recommendations of the new ENUM protocol to achieve the best performance.

8. References

[1] J.jung, E.Sit, H.Balakrishnan and R.Morris, "DNS Perfromance and the effectiveness of caching" *Proceedings of the ACM SIGCOMM Internet measurement Workshop*, November 2001.

[2] H. Shang, Craig E. Wills Using related Domain Names to Improve DNS Performance " *Technical Report WPI-CS-TR-03-35,* Worcester 2003.

[3] Getting underway with ENUM: Building a DNS infrastructure for the convergence of Telecommunication Networks, *Nominium White Paper*

[4] Nigel Walker, "Namre resolution and Performance of DNS," *In the Proceedings of Multi-service Networks*,1999.

[5] J.wang, J.yang, H.Zhou, G. xie and M.zhou, "Measuring one-way delay with multiple clock dynamics," *Parallel and distributed computing aplications and technologies*, 2003.

[6] L. Zhang, Z. Liu, and C.H. Xia, "Clock Synchronisation algorithms for network measurements," in *Proc. Of INFOCOM*. 2002.

[7] VINT – Virtual InterNetwork Testbed, http://www.isi.edu/nsnam/vint/index.html

[8] C. Liui and P. Albitz, DNS and BIND, 4th edition. Oreilly, April 2001.

[9] K. Salamatian, B. Baynat and T.Bugnazet , "Interpretation of losses observed on the Internet by infering traffic characteristics" *DIMACSWorkshop on Internet and WWW Meaasurement., Mapping and Modeling,* Feb 2002.