A Simulation Tool to Study ENUM Protocol

Saulo H. da Mata*, Johann M. H. Magalhaes*, Alexandre Cardoso*, Helio A. Carvalho**, Paulo R. Guardieiro*
*Federal University of Uberlandia, Brazil
**CPqD, Brazil
damata.saulo@gmail.com, johann@iftm.edu.br, alexandre@ufu.br, helioac@cpqd.com.br, prguardieiro@ufu.br

***Abstract*— ENUM protocol has been gaining prominence as the most promising solution to build a bridge between IP and traditional communication networks. However, there is a shortage of tools for study of this protocol. In this context, this paper presents a simulation tool that enables the study of ENUM protocol. The design of the simulator was based on experiments carried out in laboratory, mathematical models and, subsequently, implementation, integration and validation of the simulator. Results indicate that the simulation tool rightly describes the behavior of the real system considered. Results also proved the success of the proposed modeling strategy. In addition, we evaluated the performance of BIND server and built new modules for NS-3 simulation environment.**

***Index Terms*— ENUM, BIND, NS-3, modeling and simulation.**

I. Introduction

Nowadays, there is a wide range of options when the issue is communication. There is a common vision that, in the future, all these communication services will be based on IP network, since this technology allows the integration of voice, data and other services in a unique network with a lower cost than the currently offered by traditional communication networks [1]. The best example of this scenario is Voice over IP (VoIP) application.

However, this migration may take a long time, what implies a transition period and the necessity of building a bridge between these networks. It is in this context that the use of Electronic Number Mapping (ENUM) protocol stands out as a plausible solution.

ENUM [2] is a technology that maps E.164 telephone numbers [3] used in PSTN into Uniform Resource Identifiers (URIs) [4] used on the Internet. Although ENUM is based on existing DNS infrastructure, it has certain features, for example, a database with a huge number of records and the requirement of short resolution time to reach a PSTN similar performance. In addition, it causes an increase in network traffic, since the size of the response message to ENUM queries is greater than in classic DNS. Therefore, it is of fundamental importance a study of how ENUM servers perform on this new context.

This performance study can be accomplished in two ways: using a real scenario or through modeling and simulation environment. In the first option, it is necessary to configure various machines and the process of measurement is neither easy nor cheap due to the size of the system. On the other hand, the modeling and simulation approach offers a study environment with many possibilities with a lower cost and results close to those found in actual measurements. However, looking for simulation tools that allow the study of ENUM protocol, we observe a shortage of tools to accomplish this task. In [5], we can find a simulation tool based on French ENUM architecture and implemented on the Network Simulator 2 (NS-2) environment. Although [5] presents a successful modeling process, the simulation tool is inflexible with a restricted range of values for the evaluated parameters. In [6], we find a second simulation tool based on the former cited work. In this second approach, the range of values for the evaluated parameters were improved and new scenarios were presented. Nevertheless, this second tool was also implemented on the NS-2, and presented a high consumption of memory as well.

In this context, this paper presents the development of a new simulation tool with accuracy and efficiency in mind. To accomplish this task, we evaluated the performance of the name server BIND in a test bed. Results show that BIND has a great performance for the evaluated parameters. Next, we proposed mathematical models to describe BIND behavior and finally we implemented, integrated and validated the simulation tool on the Network Simulator 3 (NS-3) environment.

The rest of the paper is organized as follows: Section II describes our test bed and methodology. Section III presents BIND performance evaluation. In Section IV, we present our simulation model. Section V describes the simulation tool implementation process. In Section VI, we summarize our simulation results. Finally, concluding remarks are offered in Section VII.

II. Description of Test Bed and Methodology

A. Hardware Profile

The test bed consist of two Linux machines, running Ubuntu 11.10 and Ubuntu Server 11.10. Each machine has two processors Intel Xeon 6-Core E5645 HT 2.40 GHz, 16 GB of RAM and two hard drives of 500 GB. Both machines are connected via 1000 Mbps Ethernet connections.

B. ENUM Server

We selected BIND server for performance evaluation, since it is a stable server, with many functionalities and documentation. In addition, BIND has been named as the most used DNS server on the Internet. In our tests we used BIND version 9.8.1.

C. EnumBenchTool

EnumBenchTool is a test management tool for benchmarking of ENUM servers developed by the Network Computer Lab (NCL) of the Federal University of Uberlandia. This tool has been developed to automate, standardize and validate the tests and to facilitate the achievement of results.

At the current stage of development, EnumBenchTool admits the servers: BIND, MyDNS-NG, NSD and PowerDNS. This tool is not responsible for the benchmarking test itself, but it packs several other existing tools, simplifies the benchmarking management and makes configuration, synchronization and validation processes transparent to user. Among these packed tools, we highlight DNSPerf [7], which is a software developed by Nominum, widely used to analyze the performance of authoritative DNS servers.

D. Performance Metrics

Two performance metrics were defined for the tests:

- Throughput: the number of queries which the ENUM server can respond successfully per unit of time, i.e. queries per second (qps).
- Response Time: time between sending of query to server and the moment the client receives the answer to that query.

E. Layout of test bed

The layout of test bed is based on [8] and is illustrated in Figure 1.

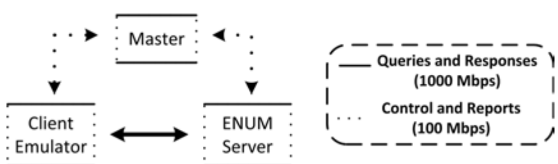


Fig. 1. Layout of test bed.

Q

F. Database Record Sets

To check whether the server meets the requirement for scalability, two database record sets with 500 thousand (500k) and 5 million (5M) of records were defined.

G. Query Files

Using the EnumBenchTool, we created query files for each database record set. For each query file, the Fully Qualified Domain Names (FQDNs) were generated in a random manner to avoid cache memory interferences.

H. Zone Files

Each record set was divided into ten zones, with each zone consisting of one tenth of the total number of records. This division method, reported in [9], is a practical and efficient generation process for different types of records. Each zone file follows the syntax of BIND zone files configuration. Each FQDN in query file has only one correspondent Naming Authority Pointer (NAPTR) record in zone file. An example record is as follows:

```
$ORIGIN 0.3.3.4.3.5.5.e164.arpa.
0.0.0.0.0 NAPTR 0 0 "U" "sip+E2U" \
"!^.*$!sip:553433000000@enum.ufu.br!" .
1.0.0.0.0 NAPTR 0 0 "U" "sip+E2U" \
"!^.*$!sip:553433000001@enum.ufu.br!" .
```

I. Number of Clients

We established that the number of emulated clients, who make simultaneously queries to server, is in a range of 2 to 1000 clients. From results, this range proved to be sufficient for stability of the system, since the server entered in saturation condition with about 300 clients, and we did not observe great changes on the throughput from 300 to 1000 clients on the system.

III. BIND Performance Evaluation

The tests carried out at NCL were constituted of multiples steps. For each of these steps, the number of emulated clients was incremented, according to commands sent by the master entity. Each step had a duration of 60 seconds, in which DNSPerf sent queries to ENUM server. DNSPerf has a timeout of 5 seconds to consider a query lost. So, at the end of the step, there was a timeout of 10 seconds, before the starting of the next step, to assure that the queries from the earlier step would not interfere in the results of the current step. As we did our tests in a laboratory environment, this timeout value proved to be suitable for our purpose.

Figure 2 shows BIND performance for different metrics, considering both record sets previously defined. In Figure 2a, it is possible to observe that BIND presents a great performance for throughput, reaching a maximum throughput over 300000 queries per second (300 kqps). Figure 2b shows BIND average response time. We can note that response time does not exceed 1 ms. This low response time is the main cause for the high throughput presented by this software.

Figures 2a and 2b indicate that BIND reached saturation when 300 clients were sending queries to ENUM server. At this instant, the throughput of the server stops to increase and enters in an unstable zone. Similarly, the response time stops to increase after this saturation point. However, the initial prevision was an increasing of response time, since for each successive step a greater number of clients will send queries to server, and it can attend only part of these queries.

The explanation for this behavior lays on the way that DNSPerf calculates the response time. This tool just considers the successful requests to calculate the response time and, moreover, there is not a retransmission process for lost queries, as generally occurs with a resolver. Thus, even on the saturation zone the server answers the queries with a response time near those found for maximum throughput and, therefore, the response time on the saturation zone shows only a small variation. However, as only one part of this increasing number of queries is attended, the system starts to experience increasing losses, as shown in Figure 2c.

Figure 2d shows the behavior of response time standard deviation. This information is important to modeling and implementation process. Figure 2e indicates BIND CPU usage. We can note that BIND has not overpassed the maximum CPU capacity of the server, confirming that the saturation of the server was caused by the software and not by hardware restrictions.

In order to verify if other variables interfered on the results, two auxiliary metrics were considered: DNSPerf CPU usage and network usage. Figure 2f shows that only part of the total CPU resources were used by DNSPerf on the client machine, excluding the possibility of interference. Similarly, Figure 2g illustrates the network usage, and we can observe that the average value of usage has not overpassed the maximum capacity of the link (1000 Mbps).

IV. Simulation Model

The simulation tool is constituted by simulation modules that represent each of the elements present in actual system, such as clients and authoritative servers. Thus, the simulation model is formed by mathematical models for each one of these actual elements.

A. *Dnsperf Model*

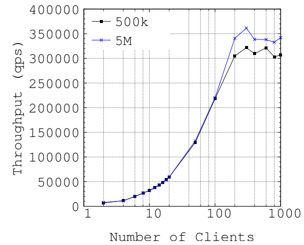
Dnsperf is an open source tool written in C. The model for this tool is a re-written version in C++, according to NS-3 specifications.

B. *Authoritative Server Model*

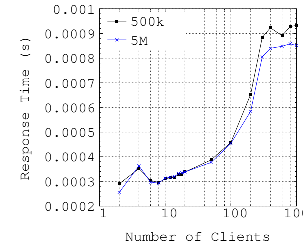
The proposed model for authoritative server is based on graphical analysis, i. e., we built our model from the behavior of the curves depicted in Figure 2. We evaluated two approaches. In the first approach we take in account the fact that, observing Figure 2, we can note that in most of the cases the behavior of the server is given by a combination of different functions. Thus, we can split this curve in multiples segments and find a suitable function that has a similar behavior, for each segment. To find this function we used linear or exponential regression, depending on the behavior of the segment of curve. This approach was reported first in [5].

In the second approach we used the samples obtained in the test on NCL to split the original curve in a set of small segments. When the simulation tool needs a parameter value (e.g. Response Time) for a given number of clients, it checks first whether this number of clients was sampled. In positive case, the simulation tool uses this sampled value directly. In negative case, the simulation tool looks for the samples immediately below and above the given number of clients and calculates the required parameter value through interpolation.

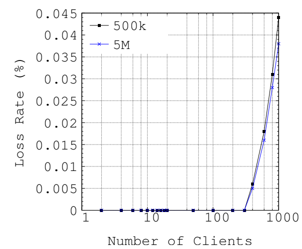
In short, the difference between these two approaches is the size and number of the segments that the curve is split. Table I shows the prediction equations found from the regression process. In this table, x represents the number of clients in the system. One can note that in this example, the original curve was split in maximum three segments. On the other hand, in the sample approach the curves were split in 1000 segments, that was the number of samples that we got in the experiments.



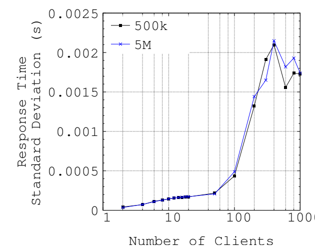
(a) Throughput



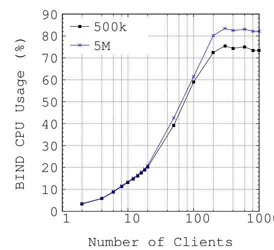
(b) Response Time



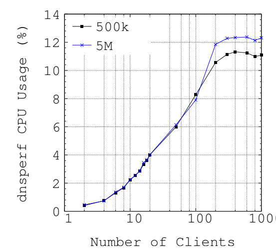
(c) Loss Rate



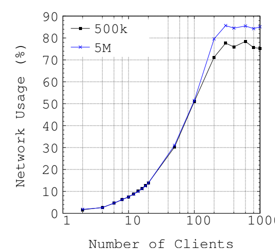
(d) Response Time Standard Deviation



(e) BIND CPU Usage



(f) dnsperf CPU Usage



(g) Network Usage

Fig. 2. BIND performance.

C. *Link Model*

The NS-3 simulation environment already has a simple link model. This model allows configuration of bandwidth and propagation delay. Although simple, this model is suitable for our study. We set our model to use a bandwidth of 1000 Mbps with 25 ns of propagation delay.

V. Implementation of Simulation Tool

Although NS-3 is a powerful environment for network simulation, it is relatively new and it is common for researchers to develop the modules that they need. For our simulation tool

TABLE I

PREDICTION EQUATIONS FOR THE ENUM SERVER CONSIDERING 500K OF RECORDS.

Average Response Time	
Interval	Equation
$0 \leq x < 303$	$y = 0.000311e^{0.003716x}$
$303 \leq x < \infty$	$y = 0.000870 + 7.20 * 10^{-8}x$
Standard Deviation of Average Response Time	
Interval	Equation
$0 \leq x < 79$	$y = 0.000089 + 0.000002x$
$79 \leq x < 303$	$y = -0.000207 + 0.000007x$
$303 \leq x < \infty$	$y = 0.001946 - 2.66 * 10^{-7}x$
Loss Rate	
Interval	Equation
$0 \leq x < 303$	$y = 0$
$303 \leq x < \infty$	$y = -0.018875 + 6.26 * 10^{-5}x$

we developed two simulation modules and one configuration module:

- DnsQueryTool: this module is responsible for executing the functions performed by dnsperf. From the number of clients indicated by user, this module simulates the queries sent to DNS/ENUM authoritative server.
- DnsAuthoritativeServer: this module includes some of the functionalities of an authoritative name server based on BIND software. The behavior of this module is led by the simulation models previously described.
- DnsConfigTool: this module is responsible for generating the configuration files required by simulation modules.

Figures 3 and 4 illustrate the flow diagram for modules DnsQueryTool and DnsAuthoritativeServer, respectively.

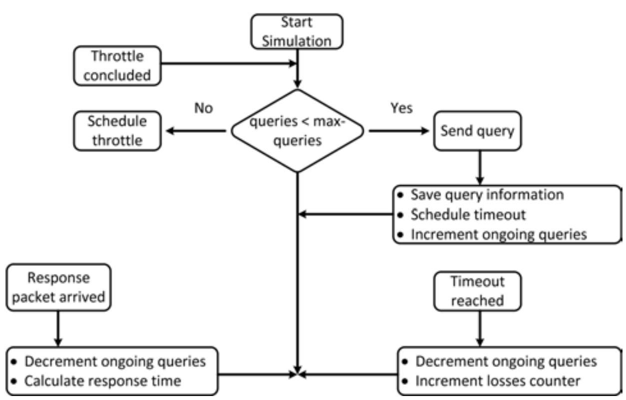


Fig. 3. Flow diagram for DnsQueryTool module.

Figure 5 illustrates the topology of the simulation tool.

Besides these modules, we had to implement other auxiliary classes to properly connect these modules. Figure 6 illustrates all the classes developed and how they were integrated to NS-3 structure. It is important to mention that one of the main features of NS-3 is the closeness of its modules to the actual system. In this sense, we designed our modules to be as close as possible to the real system. The main examples are the DNS query and response messages. These messages in our

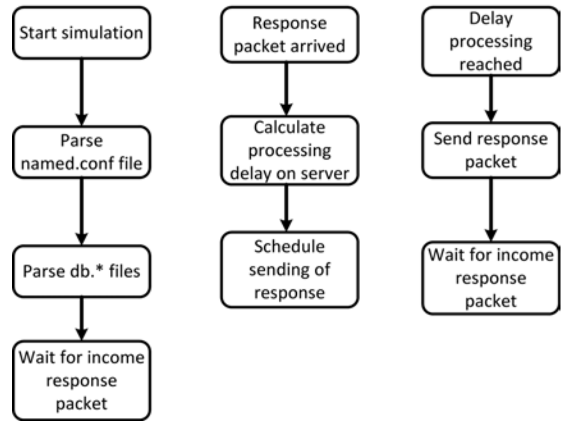


Fig. 4. Flow diagram for DnsAuthoritativeServer module.

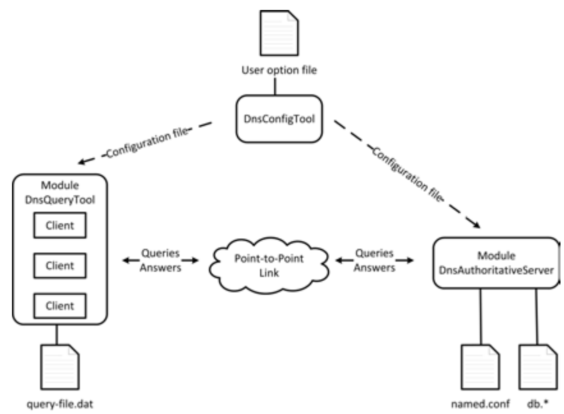


Fig. 5. Simulation tool topology.

simulation tool are exactly equal to the ones implemented in a real system. To verify this fact, we captured packets from real and simulated environments and compared their contents.

VI. SIMULATION RESULTS

For our simulation tests we set the parameters as we did for test at NCL, i.e. link of 1000 Mbps and 25 ns, dataset with 500k of records and simulation time of 60 seconds. Simulation tests were constituted of multiples steps. For each of these steps, the number of simulated clients was incremented. Each step was run five times with different values of seed to ensure the randomness of the results. The final result is the average value of these five samples.

We evaluated two different simulation models, and the results are depicted in Figures 7 and 8. As we can see, both strategies presented a good performance to model the behavior of the DNS/ENUM server. However, to verify which approach got the best performance we calculate the error made by each one of them, as shown in Figure 9. In this figure we note the samples modeling approach presents results closer to actual results. Since the samples modeling approach splits the original curve in small parts, this strategy has a better response to variation of the original curve, and consequently, it has a

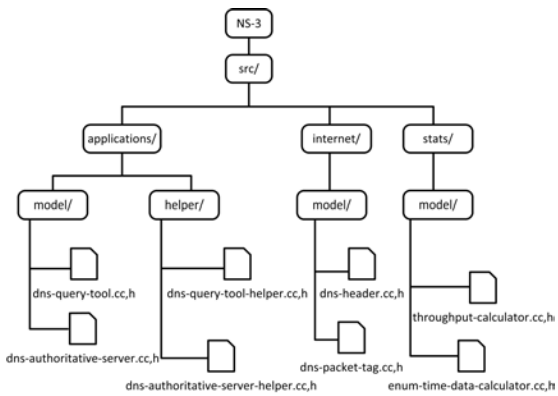
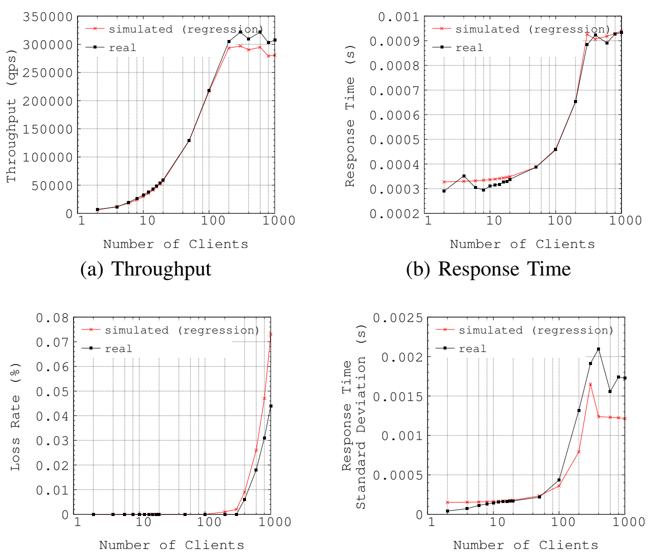


Fig. 6. Integration of modules to NS-3.

lower error.

Figure 10 shows the confidence interval (95%) for throughput simulation results. In Figure 10a we can see that the range of confidence interval is so narrow that it is only partly visible on the saturation zone. Even in this zone, the range remains in satisfactory values, as shown in Figure 10b. Since the error and confidence interval stay within a satisfactory range, we consider that the simulation tool was successfully validated.

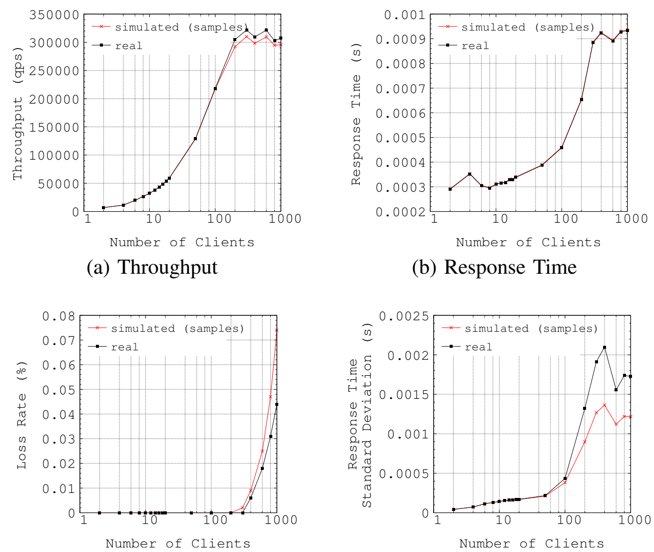


(a) Throughput (b) Response Time

(c) Loss Rate (d) Response Time Standard Deviation

Fig. 7. Simulation results for regression modeling approach.

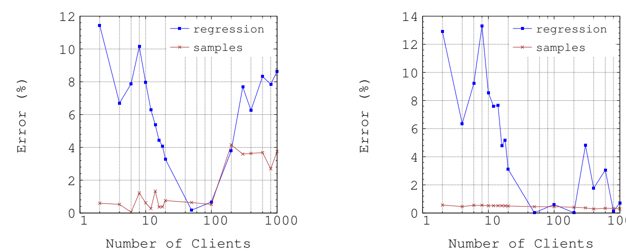
Comparing our results with other previous works is not an easy task, since it is difficult to build a fair environment due to the differences between scenarios and simulation tools. However, to attest the importance of our work, we can highlight the main differences between our and related works. The first difference is the use of a new simulation environment. All related works implemented their tools in NS-2. We did not compare the performance and efficiency of our tool with related tools, but there are comparative studies, for example [10], that attest the superiority of NS-3 over NS-2. In second place,



(a) Throughput (b) Response Time

(c) Loss Rate (d) Response Time Standard Deviation

Fig. 8. Simulation results for samples modeling approach.



(a) Throughput (b) Response Time

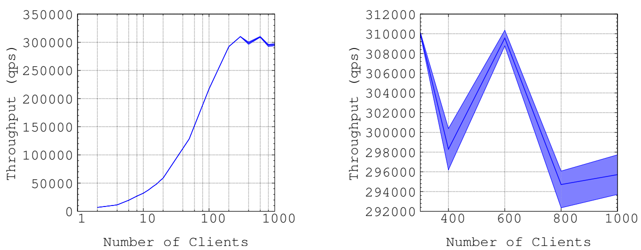
Fig. 9. Simulation error for modeling approaches.

our tool is based on high performance machines and more updated version of the ENUM server. This is very important to offer more realistic and updated results. Moreover, due to these former cited aspects, our simulation tool allows the user to evaluate the scenario with a greater range of the parameters. On the other hand, we are aware that at the current version, our simulation tool supports a simpler scenario than the related works. However, it offers a more efficient and reliable base for future scenarios implementations.

VII. Conclusion

Due to the increasing necessity of integration between the different existing communication networks, ENUM protocol has gained focus as the most promising solution. In face of the challenges to deployment of this protocol, there is a demand for studies of its behavior. Thus, in this paper we proposed a new simulation tool for study of ENUM protocol.

We executed tests in a laboratory to evaluate the performance of BIND under a high workload. The results showed that BIND had a satisfactory performance. In our tests, BIND reached a maximum throughput over 300 kqps for both record sets with a response time lower than 1 ms. These results show that BIND has great scalability for the database, since the



(a) Full observed interval (b) Saturation zone

Fig. 10. Confidence Interval (95%) for samples modeling approach.

throughput was not negatively affected even with a database ten times bigger than the first. This great performance is reached due to investments made in hardware and software that enabled maximum throughput to be 50 times greater than the throughput reported in [8]. This scenario brings good expectations for the future of ENUM protocol, since the challenges for its deployment are being gradually overcome with technological advances.

We evaluated two modeling approaches. Results showed that samples modeling approach performed better than the regression approach. However, the former approach requires the knowledge of data from tests in laboratory. On the other hand, regression approach allows prediction of scenarios not contemplated in tests in laboratory. Thus, the simulation tool makes use of both approaches, according to the scenario under study.

Our simulation tool was implemented to work with NS-3 simulation environment. We designed and implemented three new modules for this environment to reach our goal. It is important to highlight the realistic approach in the project of these modules that improved overall simulator reliability.

Simulation results proved the success of our simulation tool, since we got a small error when they were compared to the actual results. In short, we evaluated the performance of an important ENUM server. Besides, we built a new simulation tool, based on new modules for NS-3, offering new possibilities for researchers interested in ENUM protocol.

VIII. Acknowledgements

The authors gratefully acknowledge financial support from the Studies and Projects Finance Organization (FINEP), under project No. 01.10.0560.00.

References

[1] N. Beijar, "Telephony Routing with Support for Number Portability in Interconnected Circuit and Packet Switched Networks," Master's thesis, Helsinki University of Technology, Espoo - Finland, 2004.

[2] S. Bradner, L. Conroy, and K. Fugiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)," RFC 6116, 2011.

[3] ITU-T, "The international public telecommunication numbering plan," Tech. Rep., 2005.

[4] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, 1998.

[5] S. Balakrichenan, T. Bugnazet, and M. Becker, "Studying ENUM Performance with Modeling and Simulation," in *Modelling Simulation, 2007. AMS '07. First Asia International Conference on*, 2007, pp. 82–87.

[6] S. H. da Mata, P. R. Guardieiro, A. Cardoso, E. Lamounier, and L. C. Theodoro, "Performance Study of ENUM Service Based in Modeling and Simulation," in *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, 2011, pp. 1–5.

[7] *dnsperf: DNS Performance Tool Manual*, Nominum, 2007.

[8] C. Shen and H. Schulzrinne, "Measurement and Evaluation of ENUM Server Performance," in *Communications, 2007. ICC '07. IEEE International Conference on*, 2007, pp. 1967–1972.

[9] C. Shen and H. Schulzrinne, "Evaluation and Comparison of BIND, PDNS and Navitas as ENUM Server," Columbia University, Tech. Rep. cucs-029-06, 2006.

[10] E. Weingartner, H. V. Lehn, and K. Wehrle, "A Performance Comparison of Recent Network Simulators," in *Communications, 2009. ICC '09. IEEE International Conference on*, 2009, pp. 1–5.