Design and Implementation of a Network Management Proxy Based on MTNM Solution Suite

Ghasem Karami

Department of Computer Engineering, Qazvin Branch Islamic Azad University, Qazvin, Iran
ghasemkarami87_36@yahoo.com

Abstract: The main protocol that is currently used to manage computer networks is known as SNMP. Ease of use and expansion are the main advantages of applying the protocol to the equipments and computer networks. Despite the benefit of these advantages, use of this protocol in networks that generate high traffic and has a wide range of network devices should be faced with serious challenges. To overcome the difficulties of this protocol, several methods have been proposed. One of the best available techniques is a standard under name MTNM that it has been proposed as a solution suite and it has been formed according to distributed architecture, CORBA. In this paper, we intend to design and implement a proxy based on this standard. The main task of this proxy is to extract the data using SNMP commands to be placed in data structures of MTNM solution suite. Evaluation results clearly show the better performance of the proxy than SNMP protocol, in the term of traffic and run time.
[Ghasem Karami. **Design and Implementation of a Network Management Proxy Based on MTNM Solution Suite.** *Academ Arena* 2017;9(10):34-40]. ISSN 1553-992X (print); ISSN 2158-771X (online). http://www.sciencepub.net/academia. 5. doi:10.7537/marsaaj091017.05.

Keywords: Network Management; SNMP protocol; CORBA; MTNM Solution Suite v3.5; Distributed programming

1. Introduction

In recent years, The structure and application of computer networks has seen enormous changes so that it is possible to support a wider range of technologies into a single network. Technologies such as TVIP and VOIP are examples of this type. Generally, networks which are not only limited to a specific technology and support a wider range of technologies known as Next-Generation Networks (NGNs).

Providing new technologies that are mentioned within a single network (NGNs) lead to reduced construction and maintenance costs of dedicated networks for a variety of different technologies, although this has led to the emergence of new aspects of complexity. Because of these complexities, management and control of these networks is very important.

Management concept in TCP/IP networks, as one of the main components of next generation networks, is based on SNMP protocol. This protocol is located at the application layer and it uses UDP protocol for data transmission. Due to the high speed, low memory usage, and ease of use, this protocol is used widely in network equipments. Despite take advantage of these benefits, this protocol is problematic.

Problems of this protocol are as follows: need to exchange many messages to transmit huge volume of management data, weak encryption method, long prefix in OIDs naming schema, weak security mechanisms, lack of a mechanism to overcome increasingly managerial data, monitoring is not notification driven and centralized management method [1][2].

Indeed, the problems of the protocol is such that the use of it in next generation networks due to the high volume of traffic generated, will be faced with serious challenges.

A basic approach for managing TCP / IP networks, as one of the next generation network infrastructure, is to use standard interfaces. Standard interfaces are key elements in the management of heterogeneous networks because they reduce the cost of adding infrastructures and make it easier to complete the operations. Moreover, standard interfaces solve problems related to the lack of scalability and independent of the framework can provide.

CORBA is one of the best middle-wares which are formed on transparent interfaces. MTNM solution suite is one of the best standards that has been based on CORBA architecture and presented for the management of next generation networks. Finally, this standard is a set of interface definition language (IDLs) files that are defined using the CORBA architecture[3]. The Solution considered in this paper is based on this standard too. The solution is to design a proxy for data extraction using SNMP commands to be placed in data structures of MTNM solution suite.

In the second part of this article we will introduce the advantages and disadvantages of the current methods of network management. Our proposed solution will be presented in this part, too. In the third part we will introduce the proxy structure, how the SNMP protocol problems covered by the

proxy, benefiting from the advantages of CORBA middleware and MTNM solution suite and also how to cover the requirements of next-generation networks. In the forth section, we present the results of testing the proxy and will refer to its strengths. In the fifth section, we present conclusions and will refer to some of the topics that can be applied in other papers.

2. Introduce the current network management approaches

In this section, we will introduce the current network management approaches and the problems of these methods will be mentioned in the field of next generation networks management, besides the strengths of these methods. At the end of this section, our proposed solution is presented for the management of next generation networks and covering the shortcomings of current network management methods.

UDP transfer protocol is one of the fundamental problems of the SNMP protocol. Maximum size of an SNMP message is too low for bulk transfer, by using UDP. We must have exactly one PDU per SNMP message which its maximum size is limited to 64 Kbytes. Typically, it is equal to 1472 bytes for a LAN and 548 bytes for a WAN. In these circumstances, about 11 messages needs to transfer a table size of 15 KB, while about 2,000 SNMP messages required to transfer a table size 1 Mb. The large number of SNMP messages is not desirable at all because the overall latency of a data transfer increases with the number of PDU exchanges[4]. This delay has three aspects: first end-host latency increases because both the agent and the manager have to process more packets. Second, network latency augments because more SNMP messages induce more round trips and increase the overall transmission delay. Third, the higher the number of packets, the higher the number of headers to move about and the higher the network overhead. Because of this problem and many other problems, use the TCP protocol rather than UDP protocol is preferred. In fact, there are two reasons to choose TCP. First, it significantly reduce data losses. Second, TCP allows for very large application-level messages. This make bulk transfers more efficient and diminishes network and end host latency, as the number of messages exchanged over the network is significantly reduced.

IRTF Network Management Research Group proposed an SNMP-over-TCP transport mapping as an alternative to the standard SNMP-over-UDP transport mapping[5]. But just relying on TCP protocols are not sufficient to cover the requirements of next-generation networks (NGNs), although the advantages of this protocol are included in our proposed solution.

Another method of network management is Web-Based Enterprise Management (WBEM) which proposed to overcome the SNMP protocol problems. It has many problems, despite the advantage of positive features, such as the use of TCP and XML[6]. These problems are as follows: First, WBEM is designed only for use in IP networks and cannot apply to telecommunication networks, so there is no coverage requirements for next generation networks. Second, WBEM use of XML to represent management data inside an HTTP pipe, but XML is verbose, regardless of its positive characteristics, and validating an XML document takes time and consumes both CPU and memory resources[4]. Third, according to RFC 2774, it is only possible to define HTTP extension headers (domain-specific headers) in HTTP/1.1. As the result, if we abide by the HTTP/1.0 specification, we cannot use domain-specific HTTP headers, so we cannot define CIM extension headers, thus CIM operations cannot be encapsulated in HTTP. However, if we do use CIM extension headers, we use domain-specific HTTP headers, hence we break HTTP/1.0. This issue is serious because the vast majority of the embedded HTTP servers deployed to date are based on HTTP/1.0. As a result, CIM operations over HTTP do not work with most deployed equipments, unless applications deliberately break the HTTP/1.0 specification[4].

WIMA (Web-based Integrated Management Architecture) is a another management architecture for IP networks in place of SNMP. This architecture consists of two parts: a push-based architecture for regular management and notification delivery, and a pull-based architecture for ad hoc management[7]. This architecture is also have some problems, despite the relatively complete coverage of SNMP protocol problems. These problems are as follows: first, This architecture has been designed purely for IP networks so it can not be considered as a solution for next-generation networks, Just like WBEM. Second, WIMA formed based on components so the reliability of new management platforms based on COTS components and object-oriented frameworks is a serious issue. Third, need for management software integrators to test whether the components are working properly together.

Our proposed solution to overcome SNMP protocol problems and coverage the requirements for next generation networks is Multi-Technology Network Management (MTNM). MTNM is a CORBA-based network management System-to-Element management system interface suite. It supports the management of these technologies: SONET/SDH, PDH, DWDM, Ethernet, DSL, ATM,

Frame Relay, and Control Plane management[3][8]. MTNM uses a single interface infrastructure and applies the same patterns across multiple technologies, so it is a good choice for next generation networks management. In this paper, we intend to design a proxy based on this standard. The ultimate goal of this proxy is to extract data from the agents by using SNMP commands, so that this data can be sent via the MTNM data structures(IDL files) to the manager.

3. The Proxy Architecture

The proxy is designed and implemented based on MTNM solution suite and CORBA middle-ware. Obviously, the proxy has a distributed nature due to the distributed nature of CORBA middle-ware. In other words, in conducted relation between the agent and the manager in higher layer, the designed proxy can be located on the agent side (on the agent computer) or it can be located on the manager side. Put the proxy on the agent (or with the minimum distance from the agent) will result in the maximum benefit, however, neither of these two conditions is not necessary and the proxy can be located between the manager and the agent with any distance. Figure 1 shows how the proxy is placed on the agent machine.

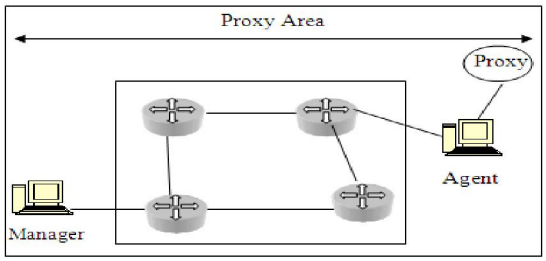

Figure 1. Communication schema and the location of the proxy

Benefit from the ability of existing management tools and protocols and overcomes their limitations is the main objective of this proxy. As a whole, three groups of advantages are obtained as a result of implementing proxy that are mentioned in the following.

The first group to benefit from the implementation of the proxy related to eliminate SNMP protocol problems. Property "being local" in the proxy, causing some of the problems in SNMP being solved. Property "being local" here means to run the proxy on a system that appeared in the role of agent and can support SNMP protocol or run it on a system that is connected to the target system through a fast and secure network. Local enforcement is necessary to cover SNMP problems, however the proxy can be on any system and with any distance from the target system. List of problems and how to eliminate them using the local properties are as follows:

A. Exchange large number of messages problem to send data

In certain conditions, using SNMP protocol bulk Commands for transferring a table require exchange of very much control messages between the manager and the agent that leading to decreased efficiency and network resources are wasted [4]. In other words, a manager uses trial and error method to recognize the correct size of a table row in some circumstances, which this trial and error causes a heavy burden on the agent, manager and the network and exchange of a large number of SNMP protocol messages. Local implementation of the proxy issues this problem to be resolved partially because it reduces the distance between the agent and proxy, as the result messages exchanged over a shorter distance and costs are paid less for the exchange of messages. The next phase of work is to send the information from the proxy to the manager. In this part, there are no particular problems to send data to the managers by proxy because the data transmission is done by using the TCP protocol. One of the key features of this protocol is to send data packets as a string of consecutive bytes. In other words, there is no restriction on the size of the packets in this protocol. If necessary, TCP protocol converts data packets into smaller packets and then the packets return to the initial state in destination. Moreover, if necessary, data packets can be merged in order to build a larger packet. As a whole, there is no problem with the CORBA architecture in the term of packet size and accuracy of reaching the packet to the destination because these operations are managed by TCP protocol automatically.

B. Poor data encoding problem

SNMP protocol uses a method called BER for data encoding. Short source code, simple algorithm, low memory consumption are the most important characteristics of this method. Unfortunately, this method has major problems. High consumption of memory during encoding/decoding the data and impose large burden are the major problems in implementing this method on managers and agents. A further problem is to increase administrative data (identifier and length) compared with payload (content), leading to decreased efficiency and increased delays in data transmission[4][7]. As in the previous section, the encoding problem has two aspects. One aspect depends on the received data from the agent which can be solved or reduced by the local property of the proxy. In fact, using of local property costs are paid less for the transfer of administrative data because it has a shorter distance to travel. CDR encoding method is used to deliver data to the

manager by proxy. In this method, binary coded representation of data in memory can be accessed directly. In other words, if a manager or agent wants to send data to the other side, binary representation in memory can be accessed directly without copying the data for encoding/decoding. In doing so, the process of encoding/decoding is done more quickly and impose additional load due to data copying could be avoided [9]. Length and type of data is not sent along with the data itself, and it's a another advantage of this method compared to BER encoding. Doing this to avoid redundant data transmission and resulting bandwidth saving.

C. Using long prefix for OID naming schema

Part of this problem is related to the connection between the proxy and the SNMP protocol and sending useless data which can reduce by using the "being local" property or close the gap between the proxy and SNMP protocol. But like the previous problems, the relationship between the proxy and the SNMP protocol is only half the work. Another part of the work is related to the relationship between the proxy and the manager. This part of the problem is covered, too. Because the connection between the proxy and higher layer managers does not have a tree structure. proxy remove the object identifiers (OIDs) when sending data to the managers and no additional data is exchanged between them.

D. Unreliable transfer protocol

This problem is related the SNMP protocol and the proxy that there are two ways to solve it. In the first method, the proxy can be run locally on the agent and fully resolve sending packets using UDP unreliable transfer protocol. By doing so, the communication between the proxy and the agent is locally and packets received by the proxy are sent to the managers using TCP reliable protocol. In the second method, the proxy can be connected to the agent in the context of a high-speed and secure network. But this method cannot solve the whole problem because depending on the distance between the proxy and the agent, messages must take over part of the network using UDP protocol. Indeed, probability of packet loss or errors in them decreases, if the distance between the proxy and SNMP protocol is much less.

E. Weak security mechanisms

TCP protocol is used to send data instead of UDP protocol to solve the weak security problem. In other words, the proxy should be run locally on the agent and using TCP protocol to send information to the managers. In terms of security, one of the advantages of TCP over UDP protocol is the method of dealing with firewalls and recognizing the Fake senders from the authorized sender [10]. This superiority related to the feature is known as state Maintenance. Indeed, TCP is stateful and this feature is achieved through sequence numbers in PDUs. Firewalls can use the sequence numbers, as one of the characteristics of the TCP protocol, to easily recognize a fake sender from a authorized sender. Moreover, Outside managers can not establish a connection with the agent behind the firewall by setting the SYN field and cleaning the ACK field in TCP protocol packet [4]. Built-in capabilities of the CORBA architecture and MTNM standard can be used to apply security mechanisms, in addition to the security benefits of TCP protocol. For example, SSL as a CORBA security protocol can be used for encryption / decryption or Some of the methods in the IDL files of MTNM standard can be used for authentication[3].

The second group of proxy benefits are rooted in its underlying architecture, CORBA middleware. These benefits, which are also included in the MTNM standard, are as follows: modularity, separating interface from implementation, using the reliable TCP protocol, programs written in different languages can interact with each other, rapid development and low cost applications, benefit from distributed systems and overcome the lack of scalability problem, increase transparency and easier to find the problems[9].

The third group of proxy benefits are rooted in TeleManagement Forum importance and the MTNM standard, Specifically. In summary, the major features of this standard are as follows: the importance of compliance with the standard, broad support of all the activities required for network management, high chance of use by service providers, full compliance of management data with TMF814 modules, Increases interoperability by increasing the supply of interchangeable components and applications, decreases service activation time, fast introduction of new technologies, increases operating efficiency and automation and reduces the OSS "integration tax", i.e., the cost of integrating and maintaining systems, enables service providers (via the production of open management interfaces) to mix and match OSSs from various suppliers based on factors such as price, quality, and functionality [3][8].

4. The results analysis

In this section we are going to compare the designed proxy and the SNMP protocol in terms of bandwidth consumption and network traffic generated. Comparison of results obtained led to emergence of some positive aspects of the proxy.

The NET-SNMP package has been used, in order to connect the proxy with the SNMP based agent, in our test environment. CORBA-based proxy implementation is also done by using the progress ORBIX 6.3 software that fully supports CORBA

services. A system that is hosting the proxy has similar features to the system that manager is installed on. Features of these systems are as follows: Windows XP PC with Pentium 4, 2 GHz CPU, 2 GB RAM. The network establishes the connection between the proxy and the manager is 100Mbps Ethernet and the distance between these two is only one hop. And, The WireShark tool is used to monitor the network traffic.

SNMPWALK command is used to navigate MIB. Traversing process is started from the root with the value "1.3.6.1.2.1.2" (SNMP MIB-2 Interfaces) and will continue until the end of MIB. OIDs retrieval process can be started from the higher levels of MIB, if we need to increase the objects retrieved from the agent.

First, for a given volume of data run SNMPWALK command directly and then this command is run for the same amount of data as a proxy. Indeed, the manager first runs the command directly on the agent via the SNMP protocol and then this command done through the proxy. The results will be compared, after running this command for various volumes of data.

In the first step, we start with a relatively small volume of management data. This volume of data is approximately equal to 4000 OIDs which are exchanged as 4000 messages by using SNMP protocol. The results for this volume of management data is shown in the first row of Table 1 directly using the SNMPWALK. The first row of Table 2 shows the result of using the proxy based approach for the same volume of data (4000 OIDs). In Table 3, hierarchical structure of the protocol is shown for 4000 OIDs using the direct method. Hierarchical structure of the protocol has been shown for the same amount of data using proxy-based approach in Table 4.

Although the number of packets in the indirect method is less than direct method, but higher traffic volumes generated, according to the first row of Tables 1 and 2 and also taking into account the structure of the protocol in Tables 3 and 4. In fact, although the number of packets is less but traffic volume has increased because of certain control mechanisms in TCP packets, due to the more flexible structure of TCP packets over UDP and handling greater volumes of data in a single packet by this protocol. According to Table 4, about 22 percent of the traffic generated by TCP protocol belongs to one of its lower layer protocols, GIOP. This protocol is the infrastructure protocol for CORBA which is used for communication between its programs [9]. However, volume of data generated by TCP is higher than UDP regardless of the GIOP protocol traffic.

Table 1. Overall evaluation of direct and proxy- based methods according to the traffic and run time

OIDs	Direct Method Traffic (Byte)	Direct Method Run Time (Second)	proxy - based Method Traffic (Byte)	proxy - based Method Run Time (Second)
4000	768101	5.741	1433041	9.102
8000	1459012	11.840	1873427	10.389
12000	2176259	16.158	2092119	14.073
16000	3020741	21.369	2252261	17.896
20000	3713570	27.059	2412154	21.444
24000	451051	34.554	2654032	27.883
28000	5194785	40.375	2843591	33.956
32000	6013756	46.475	3183589	45.410
36000	6675602	52.589	3271064	49.005
40000	7326113	59.999	3467667	49.009
44000	8217889	64.649	3744994	52.250
48000	8919880	68.111	4107998	59.750
52000	983993	73.457	4287347	62.758

Table 2. Overall evaluation of direct and proxy- based methods according to number of packets

OIDs	Direct Method	Proxy- based method
4000	8161	1767
8000	15529	2220
12000	23167	2558

Table 3. Protocol hierarchical of direct method for 4000 OIDs

Protocol	Percentage
-Frame	100
-Ethernet	100
-IP	100
-UDP	99.99
-SNMP	99.96

Table 4. Protocol hierarchical of proxy- based method for 4000 OIDs

Protocol	Percentage
-Frame	100
-Ethernet	100
-IP	100
-TCP	100
-GIOP	21.51

In another example, this time for about 8000 OIDs we repeated the experiment. The results of this sample are shown in Tables 1 and 2 in the second row using direct method (SNMPWALK) and indirect method (proxy-based).

It is clear that Traffic volumes generated by the proxy-based approach is more than direct method yet, but if you do a comparison between the first and second rows, you can see traffic volumes generated by the direct method and proxy-based method are nearing to each other. The difference between the direct method and proxy-based method on traffic generated for 4000 OIDs is approximately 664,940 bytes while this difference is equal to 414,415 bytes for 8000 OIDs which represents the distance between these two methods is reduced.

Results would be very interesting, if number of objects rises to 12,000 OIDs and repeat the test. The results are shown in the third row of Tables 1 and 2. As shown, 84140 bytes of traffic generated by the proxy-based approach is reduced compared with the direct method. Moreover, the run time required to implement the desired operations have improved significantly.

If this experiment is repeated for higher volume data, then the advantages of proxy-based approach over direct method will be revealed. Results of experiment are presented for a wide range of data, from the fourth row till the end of Table 1. The diagrams in Figure 2 for both proxy-based and direct methods is shown which clearly confirms the validity of our claim.

As shown in Figure 2, traffic generated for high volume of data in proxy-based approach grows reasonably while this claim is not true for the direct method. Diagram of the direct method grows almost exponentially with increasing data exchange, while growth chart of proxy-based approach for high volumes data is increased with constant coefficients and it has linear form. Quick startup is the main reason for the better performance of the direct method over proxy-based method in low data volume. In other words, Some additional traffic generated because the proxy-based approach must first establish a session between the manager and the agent and the services are launched, while the direct approach does not produce this additional data.

Figure 3 shows the required execution time for proxy-based method and direct method. According to this Figure, For high volume data also proxy-based approach has better performance than the direct method because the number of packets exchanged in this method are lower in comparison with the direct method which leads to a reduction in run time. In Figure 3, the running time diagram of these two methods are very close to each other because the proxy is only one node away from the manager system. The better performance of proxy will be seen more, if we increase the number of intermediate nodes. However in our tests, proxy and agent are both on the same machine which leads to improved performance in terms of time.

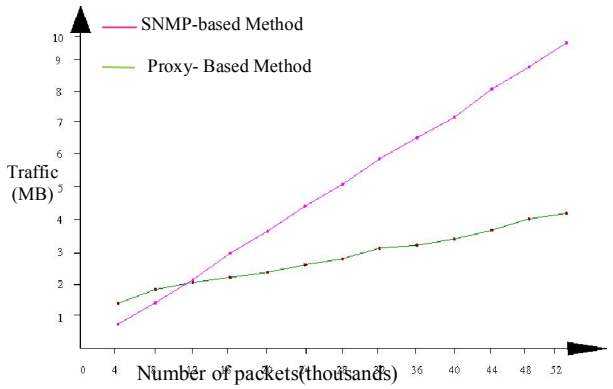

Figure 2. Evaluation diagram of the generated traffic in direct and proxy- based methods

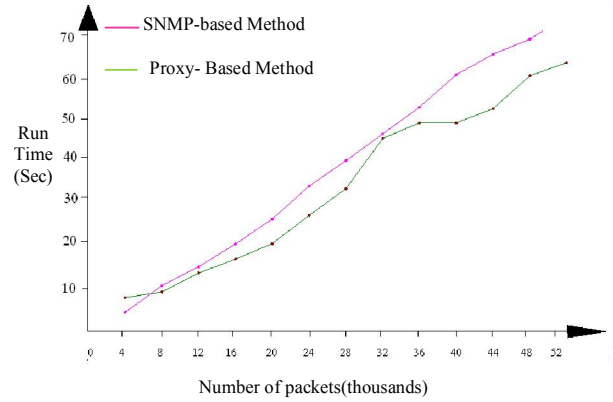

Figure 3. Evaluation diagram of run time in direct and proxy- based methods

5. Conclusion and future work

As observed, the local property of the proxy can resolve the SNMP protocol problems very well. In addition, One of the main problems with the SNMP

protocol, as one of the requirements of next generation networks, is the lack of scalability that can be covered and resolved using The distributed nature of CORBA middleware and MTNM standard.

Of the discussions, it was concluded that the current network management techniques are not appropriate for the management of next generation networks. Because these methods are limited to a specific technology. although our proposed solution is presented perfectly to cover the requirements of next-generation networks.

As the evaluation results showed, the direct method is more efficient than the proxy-based approach and produces less traffic for low volume of data generated. Advantages of proxy-based approach compared with the direct method will reveal, if the volume of data exchanged increases, something that will occur in the next generation networks certainly. On the other hand, if you upgrade the proxy from the agent level to the network level, exchanged data volume will increase significantly and the direct method actually loses its efficiency in practice and will lead to many problems including lack of scalability and bottleneck. In these circumstances, using proxy-based approach, as a scalable and suitable solution for high volumes of data, can overcome these problems.

The following items can be mentioned to improve the quality of designed proxy: upgrade the proxy from one agent mode to a network, increase the level of security, further benefiting from the advantages of distribution of the Proxy, increasing number of methods have been implemented, further reform on data structure.

Corresponding Author:
Ghasem Karami
Department of Computer Engineering
Qazvin Branch Islamic Azad University
Qazvin, Iran
E-mail: ghasemkarami87_36@yahoo.com

10/18/2017

References

1. Laxman.D, Efficient Network Management Using SNMP, Journal of Network and Systems Management, Volume 14, Number 2, 189-194, 2006.
2. Martin-Flatin J.P, Distributed Event Correlation and Self-Managed Systems, 1st International Workshop on Self- Properties in Complex Information Systems (Self-Star), pp. 61-64,2004.
3. TM Forum, MTNM Solution Suite v3.5, teleManagement standard, 2008.
4. Martin-Flatin J.P, Web-Based Management of IP Networks and Systems, New York: Wiley, 2002, pp.15-34.
5. RFC 3430: Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping http://www.ietf.org/rfc/rfc3430.txt.
6. Mishra. M, S.S. Bedi, SNMP, CMIP based Distributed Heterogeneous Network Management using WBEM Gateway Enabled Integration Approach, International Conference on Recent Advances and Future Trends in Information Technology,1012.
7. Martin-Flatin J.P, Web-Based Management, Journal of Network and Systems Management, Vol. 9, No. 1, pp. 11-13, 2001.
8. Luciani.L, Riedel.M, TMF814 Network Simulator, Ph.D. dissertation, Chalmers University of Technology, 2010.
9. Aleksy.M, Implementing Distributed Systems With CORBA and JAVA, New York: Springer, 2005, pp13-45.
10. van den Broek, J.G, Real-World Analyses of Internet Protocols, Ph.D. dissertation, University of twente, july 2012.