A Novel Technique for SNMP Bandwidth Reduction: Simulation and Evaluation

O. Said

Computer Science Department,
Faculty of Computers and Information Systems,
Taif University, KSA.

Abstract
One of the most popular and widely used Network Management Protocols (NMP) is SNMP. SNMP is extremely simple and versatile but it has scalability problems. A number of alternative NMPs such as RMI interfaces, CORBA [1] and RMON [2] have been attempted to overcome the problems with SNMP. However, these interfaces lack the standardization effort and simplicity of the usage that SNMP provides. So currently, every application develops models to achieve the desired level of scalability on its own. In this paper, we demonstrate a new technique to decrease a number of messages between manager and agent in the SNMP (i.e. traffic reduction). The base idea of our technique is to make a system manager keeps the MIB objects that are frequently required. We simulate our proposed technique using a network simulator called NS2. We compare our proposed technique with the Kwang Sik Shin attempt [3] and general polling technique [4].
Key words*: - SNMP, Network Management, Network Protocols, TCP/IP, NS2.*

1. Introduction

As the public Internet and private intranet have grown from small networks into large infrastructures, the need to more systematically manage the large number of network components within these networks has grown more important as well. In the automated network management system, there are 2 approaches in general: the centralized and distributed management approach.
The centralized approach manages the whole network in a single place (the management station). It includes the original Simple Network Management Protocol (SNMP) and Remote Network Monitoring (RMON). The SNMP management systems are based on the client/server model consisting of 3 components: a management station, agents and Management Information Bases (MIBs). The RMON setup consists of 2 components: the RMON probe and the management station. The RMON probe is an intelligent-controlled device or software agent that continually collects statistics about a LAN segment or VLAN, and transfers the information to a management workstation on request or when a pre-defined threshold is crossed. The management workstation communicates with the RMON probe and collects the statistics from it. The distributed approach is divided into the hierarchical network management and the mobile agent based management [2], [5], [6].

This paper is organized as follows: In section 2, we demonstrate the SNMP system architecture. Then, we show the drawbacks of last proposed technique [3] in section 3. We demonstrate our technique in section 4. Finally, we simulate our system, in section 5.

2. SNMP

SNMP is a popular protocol for network management. It is used for collecting information from, and configuring, network devices, such as servers, printers, hubs, switches, and routers on an Internet Protocol (IP) network. SNMP can collect information such as a server's CPU level, Server chassis Temperature… the list is nearly endless of what you can do with SNMP if configured properly.
SNMP was made with one design in mind to be simple. SNMP is a simple protocol that can be used on just about any networking device in use today. In some environments it's used heavily, in others it's scarce. Some view it as a security threat; others see it as a way to efficiently manage some of their key systems. However you decide to see it, SNMP is a easy to use, easy to set up and not very difficult to understand.
The SNMP protocol was designed to provide a "simple" method of centralizing the management of TCP/IP-based networks. If you want to manage devices from a central location, the SNMP protocol is what facilitates the transfer of data from the client portion of the equation (the device you are monitoring) to the server portion where the data is centralized in logs for centralized viewing and analysis. Many application vendors supply network management software: IBM's Tivoli, Microsoft's MOM and HP Openview are three of over 100+ applications available today to manage just about anything imaginable. The protocol is what makes this happen. The goals of the original SNMP protocols revolved around one main factor that is still in use today: Remote Management of Devices. SNMP is commonly used to manage devices on a network [2], [7].

Manuscript received February 5, 2008
Manuscript revised February 20, 2008

2.1 SNMP uses UDP

UDP stands for User Datagram Protocol and is the opposite of TCP, Transmission Control Protocol which is a very reliable and high overhead protocol.

User Datagram Protocol (UDP) is very low overhead, fast and unreliable [7]. UDP is easier to implement and use than a more complex protocol such as TCP. It does however provide plenty of functionality to allow a central manager station to communicate with a remote agent that resides on any managed device that it can communicate with. The unreliability comes in the form of checks and balances whereas if TCP sends something, it waits for an acknowledgment and if it doesn't hear back, it will resend. Since logging of devices usually happens within a time period that is cyclic in nature, then it's common sense that you missed the event and you'll catch it next time… the tradeoff being that the low overhead protocol is simple to use and doesn't eat up all your bandwidth like TCP based applications going across your WAN [2], [7].

2.2 SNMP operations

SNMP design is pretty simple. There are two main players in SNMP. The manager and the agent. The manager is generally the 'main' station such as HP Open view. The agent would be the SNMP software running on a client system you are trying to monitor.

The manager is usually a software program running on a workstation or larger computer that communicates with agent processes that run on each device being monitored. Agents can be found on switches, firewalls, servers, wireless access points, routers, hubs, and even users' workstations – the list goes on and on. As seen in the illustration, the manager polls the agents making requests for information, and the agents respond when asked with the information requested.

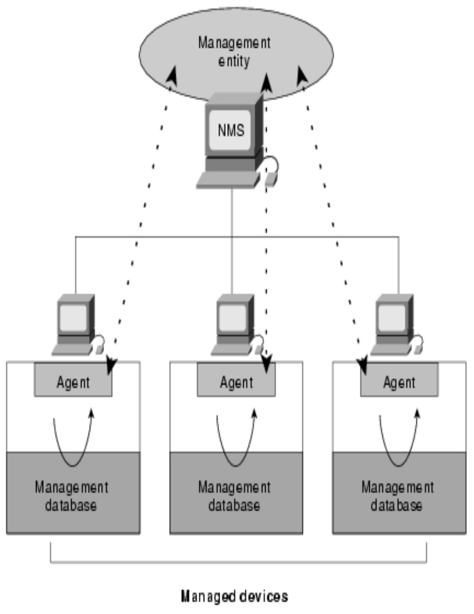

Figure (1): SNMP general view.

The manager is usually a software program running on a workstation or larger computer that communicates with agent processes that run on each device being monitored. Agents can be found on switches, firewalls, servers, wireless access points, routers, hubs, and even users' workstations – the list goes on and on. As seen in the illustration, the manager polls the agents making requests for information, and the agents respond when asked with the information requested [2].

2.3 Network Management Station (NMS)

The manager is also called a Network Management Station or NMS for short. The software used to create the NMS varies in functionality as well as expense. You can get cheaper applications with lesser functionality or pay through the nose and get the Lamborghini of NMS systems. Other functionalities of the NMS include reporting features, network topology mapping and documenting, tools to allow you to monitor the traffic on your network, and so on. Some management consoles can also produce trend analysis reports. These types of reports can help you do capacity planning and set long-range goals [2].

2.4 SNMP primitives

SNMP has three control primitives that initiate data flow from the requester which is usually the manager. These would be get, get-next and set. The manager uses the get primitive to get a single piece of information from an agent. You would use get-next if you had more than one item. When the data the manager needs to get from the agent consists of more than one item, this primitive is used to sequentially retrieve data; for example, a table of values. You can use set when you want to set a particular value. The manager can use this primitive to request that the agent running on the remote device set a particular variable to a certain value. There are two control primitives the responder (manager) uses to reply and that is get-response and trap. One is used in response to the requester's direct query (get-response) and the other is an asynchronous response to obtain the requester's attention (trap). Although SNMP exchanges are usually initiated by the manager software, this primitive can also be used when the agent needs to inform the manager of some important event. This is commonly known and heard of as a ‘trap’ sent by the agent to the NMS [2].

2.5 The Management Information Base (MIB)

We just learned what primitives were… the agent and the manager, exchanging data. The data they exchange also has a name. The types of data the agent and manager exchange are defined by a database called the management information base (MIB).The MIB is a virtual information store. Remember, it is a small database of information and it resides on the agent. Information collected by the agent is stored in the MIB. The MIB is precisely defined; the current Internet standard MIB contains more than a thousand objects. Each object in the MIB represents some specific entity on the managed device [2].

2.6 SNMPv2 and SNMPv3

With all TCP/IP related protocols, it’s a well known fact that anything dating before the creation of IPv6 (or IPng) has security weaknesses such as passwords sent in cleartext. SNMP in its original form is very susceptible to attack if not secured properly, messages sent in cleartext exposing community string passwords, or default passwords of public and private being ‘guessed’ by anyone who knew how to exploit SNMP… beyond its inherent weaknesses SNMP in its original implementation is still very simple to use and has been widely used throughout the industry. SNMP in its first version lacked encryption or authentication mechanisms. So, now that SNMP in its first version was good enough, work began to make it better with SNMPv2 in 1994. Besides for some minor enhancements, the main updates to this protocol come from the two new types of functionality, where traps can be sent from one NMS to another NMS as well as a ‘get-bulk’ operation that allows larger amounts of information to be retrieved from one request. SNMPv3 still being worked on and is incorporating the best of both versions and enhanced security as well. SNMPv3 provides secure access to devices by a combination of authenticating and encrypting packets over the network. The security features provided in SNMPv3 are message integrity which ensures that a packet has not been tampered with while in transit, authentication which is determining the message is from a valid source and encryption, which is the securing of the packet by scrambling its contents [2].

3. Related Work and its Evaluation

There has been prior work in this area [3]. Kwang Sik Shin, et al. suggested a real-time network, monitoring method for dynamic information to reduce the network traffic in SNMP. In the proposed strategy, each agent firstly decides its own monitoring period. Then, the manager collects them and approves each agent’s period without, medication or adjusts it based on the total traffic generated by monitoring messages. After receiving a response message containing the monitoring period from the management station, each agent sends a management information periodically without the management station request. This proposed technique depends on the manager selection for a group of relevant agents when the network management consumes more than the allocated bandwidth. After the studying and analysis of the last proposed technique to solve a network management traffic overloading problem, the following drawbacks have been extracted:

1- The proposed technique didn’t demonstrate a description or algorithm to determine how the manager can select the agent that will send its management messages in case of a bandwidth overloading detection.
2- The MMPs and AMPs [3] messages consume a bandwidth and are considered a system overload.
3- The proposed technique didn’t show how the agent can send an urgent message to the system manager when its sending time passed.
4- The proposed technique can’t be considered a general technique as it neglects some object changes in the agent MIBs.

4. Our Proposed Technique

As mentioned before that the main target of our paper is decreasing the number of messages between managers and agents in the SNMP system. The problem of bandwidth limitation results from a huge number of requests (SNMP manager) and responses (SNMP agent). If we try to decrease a number of connections between the manager and the agent, the number of management messages and allocated network management bandwidth will be decreased. The connection between SNMP manager and agent occurs according to the manager desire to access a variable stored in the agent MIB. This connection will be eliminated if the manager stores the urgent variables (the manager desired objects) in a simple database. We called it “Consultant Database (C.DB)”. The manager transacts the C.DB in the majority of management time. In the following subtitles, we describe a basic idea and construction of the manager C.DB.

4.1 C.DB

The suggested C.DB target is to store the objects that are frequently accessed. The C.DB consists of five fields: 1- Object Identifier (OID) which is used at the GetRequest, GetResponse, and SetRequest. 2- Object Value (OV). 3- Access Counter (AC) which determines the number of times the SNMP manager accesses a C.DB object. 4- Change Counter (CC) which shows how many times the object value is changed. 5- Change Time (CT) which calculates the time interval in which the object value is constant.

OID	OV	AC	CC	CT
1.3.6.1.2.1.4.9	10	12	7	2.51 S
1.3.6.1.2.1.4.5	5	10	4	2.31 S
-	-	-	-	-

Table (1): Simple view of C.DB

When the SNMP manager needs to get an agent object, it should search into the C.DB, before sending GetRequest or GetNextRequest. If the manager finds the target object in the C.DB, it accesses this object, otherwise it sends a request to the agent that have this object. As the manager accessed the target object from the C.DB, the AC should be increased by one. If the agent sends a message containing a new C.DB object value, the CC should be increased by one. The CT equals the difference between the previous CT and the time at which the agent message is received. Each time interval, the manager should test the AC field to detect the agent objects that hasn't been frequently accessed. These objects have low values in the AC field. This step makes the database up to date.
When the manager sends a SetRquest, it should modify the SetRequest object value provided that the same object is found in the C.DB. If the MIB is changed (i.e. any MIB's object is changed), the agent which has the changed MIB should inform the manager with the new object value. Hence, the agent should send a new type of message called Tell Message (TM). Consequently, the manager stores the new object value in the C.DB. The TM PDU (Protocol Data Unit) consists of three fields, 1- PDU type. 2- OID. 3- New value. SNMP agent can send more than one TM with a time interval determined by the network administrator.

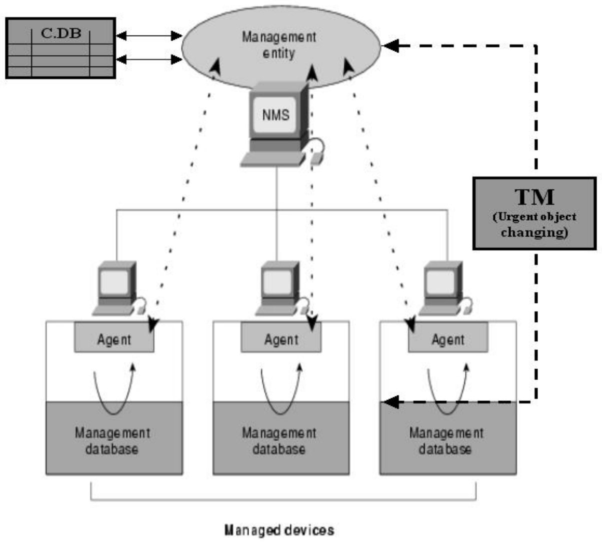

Figure (2): Our technique view.

4.2 C.DB analysis

We analyze the effect of a C.DB on the management system as regards message overloading. Assume that N is a number of management messages from the manager to the agent, therefore the total number of messages equal 2N (i.e. Requests and Responses). Assume that the maximum size of the C.DB is Z. This size (z) equals the number of objects at which the server works in a stable manner, and it depends on the administrator experience and the server specifications.
Our analysis contains six cases. The first three cases where the number of messages should be increased. The second three cases where the number of messages should be decreased. These cases are:
Case 1: If the manager doesn't use the C.DB (as the desired object is not found in the C.DB), the total number of messages will equal 2N+Z.
Case 2: If a number of desired objects equals A, and all the C.DB objects are changed, the total number of messages will equal 2N+Z-2A, Z>2A.
Case 3: If a number of desired objects in the C.DB equals A and a number of changed objects in the C.DB equals B, the total number of messages will equal 2N+B-2A, B>2A.
Case 4: If a number of desired objects in the C.DB equals A and a number of TMs equals B, the total number of messages will equal 2N+B-2A, 2A>B.
Case 5: If a number of desired objects in the C.DB equals A and there is no TMs, the total number of messages will equal 2N-2A.
Case 6: This case is the optimal one. If the manager finds all desired objects in the C.DB and no changes are done (No TMs sent), the total number of messages is reduced by 2N.

It's noted that the number of messages in the first three cases is increased. Administrator can solve these cases using a feedback strategy. This strategy determines OIDs of the C.DB objects that are frequently used by the SNMP manager. Hence; other C.DB objects should be eliminated. This will decrease the number of agent TMs. Consequently, the total number of system messages in the above three cases will be decreased. Also the objects that are frequently changed, mostly have types "Counter or Gauge" [2], [4]. So, the administrator should take care when inserting of these types of objects in the C.DB (i.e. he should only insert the highly important objects in the C.DB as can as possible).

4.3 SNMP simple infrastructure upgrading

To make our proposed technique compatible with the current SNMP, we only expand PDU, MIB, and modify the message formats corresponding to the new types.

4.3.1 MIB changing

The objects in the C.DB should be identified in the MIB. This process requires addition of a new column called consultant database column (C.DBC) to MIB. The C.DBC column has two types of entries Null or C. C is for the C.DB object and Null is for other objects that not found in the C.DB. Before an agent sends a TM, it should test the case of the object in the a C.DBC. Before the manager stores an object in a C.DB, the case of that object in the C.DBC should be modified from Null to C. The same is done in deletion from C.DB. For simplicity, manager can inform a system agent with information about the C.DB at the nearest next request that will be done after the C.DB changes

4.3.2 TM construction

The TM PDU contains the following fields:

1- PDU Type: an integer number equals 9.
2- OID: Object Identifier.
3- New Value: the new object value that changed in the agent MIB.

4.3.3 GetReques t, GetnextRequest, and Trap

The construction of GetRequest and GetnextRequest should be upgraded by adding a C.DB field. This field is one bit containing 0 or 1 value (0 for C.DB object deletion and 1 for new C.DB object insertion). The agent may use the trap message in case of urgent events in the C.DB. So, the Trap PDU construction also should be upgraded by adding a field to describe the changed object status as regards the C.DB.

5. Simulation and Results

5.1 Simulation setup

We simulate our proposed technique using NS2 [8]. To test the performance and scalability of our technique described here and to compare it to the previous techniques [2], [3], we use a part of modified version of the baseline model defined by the DARPA NMS [9]. The topology for the model that we are using can be visualized as a ring of domains, where each domain is connected to one domain preceding it and another one succeeding it. We refer to each domain as the network, as shown in Fig. 3. Each of the networks is similar to the others and consists of four sub-networks.

The sub-network labeled Net 0 consists of three routers connected by links with 5ms delay and 2 Mbps bandwidth. Sub-network 1 consists of 4 UDP servers. Sub-network 2 contains seven routers with links to the LAN networks. Each of the LAN networks has one router. The first three LAN's have 10 hosts each and the fourth LAN has 12 hosts. The other four different LAN's consisting of 40 hosts Each of the hosts is configured to run as a UDP Client. Sub-network 3 is similar to Sub-network 2.

The traffic that is being exchanged in the model is generated by all the clients in one domain choosing a server randomly from the Sub-network 1 in the domain that is a successor to the current one in the ring. We used different send-intervals of 0.1, 0.05 and 0.02 second to vary the traffic intensities, and used different numbers of domains to vary the size of the network.

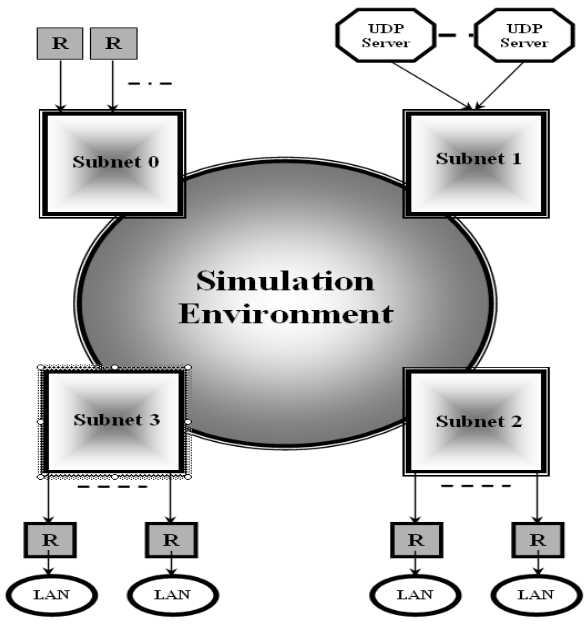

Figure (3): Simple view of our simulation setup

5.2 Simulation parameters

In our simulation, we compare our technique with the old ones (last proposed technique and general polling) using the following parameters: The bandwidth utilization, the number of management messages during the entire experiment period (traffic utilization), available memory space, and C.DB efficiency.

In the experiments, CPU utilization, available memory space, and receiving rate of UDP packets are monitored for each host. In each host, the CPU utilization has a value between 0% and 100%. The unit of an available memory space is kb and the receiving rate of UDP packets are measured by kilobits/second. The time unit is equal to 1 s in the following description and graphs. The general polling method is also carried out to compare the performance of the proposed scheme.

5.3 Results and comments

Two metrics were used to evaluate the performance and quality of network monitoring. First, the network traffic is measured to compare the amount of management messages generated by the network monitoring. Next, we evaluate the faithfulness of monitored values of an agent's information. We calculate and compare the average of an agent's monitored values from original values. In the experiments, the average to evaluate the faithfulness of monitored value is defined at [3].

Fig. 4 demonstrates the bandwidth consumption with our simulation time. Its notable that our technique keeps more bandwidth for the data transmission than the last trial and the general polling [3], [4]. This results from a traffic reduction between the manager and agent in our simulation system. The proposed scheme reduces the traffic by 20.35% when compared with the general polling scheme and 4.96 when compared with the last trial technique, fig(5). Fig. 6 reports the values of receiving rates, and available memory.

Fig. 7 demonstrates the C.DB efficiency. This efficiency is calculated by the number of objects found in the C.DB and used by the manager. It's noted that the efficiency is between 95% and 97% within our simulation time. At the second 10 and in between the second 40 and second 52, the C.DB efficiency lowers down. This is due to the huge number of changing objects that are installed randomly in the agents MIBs.

Average of the available memory space between the agent's original values and the management station's received values are 4253 2873, and 1945 Kbytes for the general polling method, the last trial, and our proposed technique respectively. The proposed scheme reduces this data type by 55.2% as regards the general polling and 32.3% as regards the last trial.

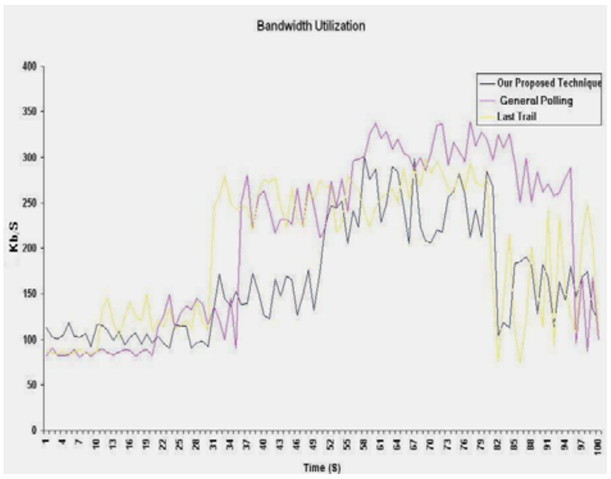

Figure (4): Bandwidth Consumption

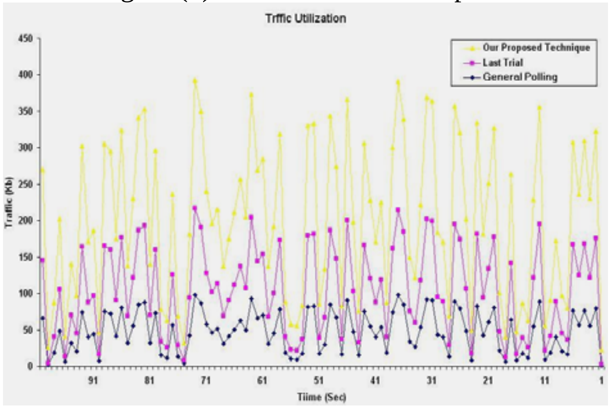

Figure (5): General Traffic Utilization

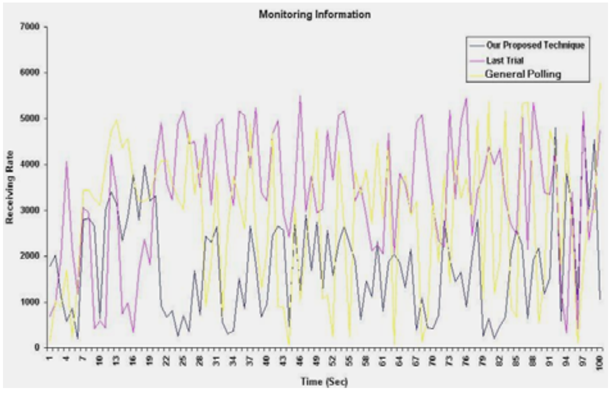

Figure (6): Monitoring Information

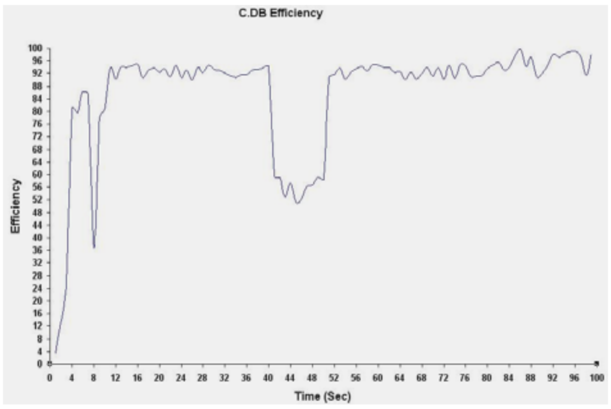

Figure (7): C.DB efficiency

6. Conclusion

In this paper, we investigated a new scheme for traffic reduction between managers and agent in the SNMP. This scheme allows the manager to keeps the high important objects in a small database called C.DB. This traffic reduction makes the bandwidth allocated to the SNMP packets low. We also simulate our new scheme using NS2 and compared it with the general polling and the last trial in this topic. Finally, we showed the results.

7. Future Work

We will try to make a standard specification for our C.DB by implementing it on different architectures. Also, we will test its efficiency as regards the real-time networks.

References:

[1] Steve Vinoski's "New Features for CORBA 3.0, Communication" ACM, October, 1998

[2] Stallings W. "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2", 2nd ed. Reading, MA: Addison-Wesley;1999.

[3] Kwang Sik Shin_, Jin Ha Jung, Jin Young Cheon Sang Bang Choi,"Real-time network monitoring scheme based on SNMP for dynamic information, Journal of Network and

[4] Computer Applications 30 (2007) 331–353.

[5] Hwang KC. A SNMP group polling for the management traffic. In: Proceedings of IEEE TENCON'99, vol. 2, September 1999. p. 797–800.

[6] Cheikhrouhou M, Labetoulle J. An efficient polling layer for SNMP. In: Network operation and measurement symposium, April 2000, p. 477–90.

[7] Case J, Fedor M, Schoffstal M, Davin J. A simple network management protocol (SNMP). RFC 1157, May 1990.

[8] J. Postel ISI "User Datagram Protocol", RFC 768, 28 August 1980

[9] McCanne, S. and Floyd, S. (McCanne, 1998), LBNL Network Simulator (NS2), 2003.

[10] NMS (Network Modeling and Simulation DARPA Program) baseline model. See web site at:

[11] http://www.darpa.mil/ipto/solicitations/closed/00-18_PIPPrint.htm

Omar Said Sayed Ahmed received his Bachelor degree in Computer Science from Menoufya University Egypt in 1997; MS.C from Menoufya University Egypt in 2002; Ph.D. degree from Menoufya University Egypt in 2005. He is currently an assistant professor in Computer Science department, Faculty of Computers and Information Systems, at Taif University, KSA. His research interests include Multimedia Communication, and Internetwork Protocols.