THE MANAGEMENT INFRASTRUCTURE OF A NETWORK MEASUREMENT SYSTEM FOR QOS PARAMETERS

Alexandru BIKFALVI, Paul PĂTRAŞ, Cristian Mihai VANCEA, Virgil DOBROTĂ
Technical University of Cluj-Napoca, Communications Department,
26-28 George Barițiu Street, 400027 Cluj-Napoca, Romania,
Tel/Fax: +40-264-597083, E-mail: Virgil.Dobrota@com.utcluj.ro

Abstract: The paper presents the designing principles of a management infrastructure for monitoring the capabilities of different network interface cards (Gigabit Ethernet, Endace), both for traffic generation and capturing at reception. This evaluation is useful to asses the goodness of captured traffic analysis and QoS performance measurements, using PC based platforms. The idea was to implement the communication of the management information between an administration console and a set of distributed SNMP-based software measurement agents for GNU/Linux platforms that enable to perform QoS measurement sessions.

Key words: *QoS, SNMP, management console, measurement agents*

I. INTRODUCTION

A measurement system capable of monitoring the capabilities of different network interface cards, both for traffic generation and capturing is useful in assessing the goodness of captured traffic analysis and QoS performance measurements. The paper presents the communication of the management information between an administration console and a set of distributed SNMP-based software measurement agents for GNU/Linux platforms. The software features a graphical user interface and a group of services that handle the management information.

The services are used to communicate with the operating system's socket interface, to perform SNMP encapsulation and decoding, a measurement session manager that has the intelligence of interpreting the measurement results. A queuing service solves the issue of asynchronous communication, by implementing a set of eight message-waiting queues. Four priority levels exist complemented by a Round-Robin servicing policy to ensure that management messages are handled in the following order: notifications, control messages, results requests and results replies.

The advantages of the proposed measurement solution versus the existing tools are the possibility of managing many test scenarios through the control of a large set of agents, no user attendance required during the experiments, customizable sessions and availability of results (numeric or plotted) both during and after the measurement is completed.

II. SCENARIOS BASED ON EXISTING TOOLS

The main goal of the Network Measurement System (NMS) software was to overcome some of the limitations of regular measurement applications available today. These limitations refer to the fact that some of the well-known and most used tools do not cope well with complex, repetitive task, especially from the results processing point-of-view. For example, *MGEN* is one of the most used free traffic generating applications. However, in a real test scenario the final data usually needs additional, time-consuming post-processing, in order to make it usable.

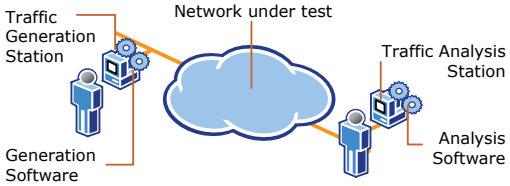

Figure 1. MGEN-based scenario

In order to perform a measurement task using MGEN the following steps need to be followed (see figure 1) [1]:

1. Install *MGEN* on the machine that generates the traffic.
2. Install a proper analysis tool on the machine analyzing the traffic.
3. Create a script to instruct *MGEN* what traffic pattern to generate.
4. Running the test and awaiting for results at the analyzing station.
5. Collecting the results data and processing it according to needs.

Suppose the objective was to measure the link behavior having the throughput as a parameter. The simplest way to do this using *MGEN* is to define a script that successively tells the program to start generating traffic with an increasing number of packets per second and after this step is completed, the traffic is generated automatically. At the receiving end a software analysis tool, such as *MGEN* or *tcpdump*, records the information about the incoming traffic. QoS parameters evaluation can be performed either by creating a dump of the incoming traffic and process it after the capturing session is completed, or by computing the required parameters in real time. The first approach has the advantage of having enough data after the dump to compute off-line a large number of parameters but it does not cope well with traffic analysis of high bit-rate flows, since the dump speed is usually limited by hard drive performance. The alternative is to compute the parameters of interest in real time and to discard all non-essential information. It takes more processing time but it can be

used for high-rates experiments, where the dump solution is not a viable choice.

MGEN perform logging of incoming traffic allowing both online and offline analysis. It is flow oriented, meaning that you can specify several different flows running simultaneously, and the program is able to make the difference between the packets belonging to different flows. *Tcpdump* however, does not have almost any measurement capabilities at all. It simply dumps incoming traffic on a given network interface (some filtering options are though available) and later one uses a third party software to extract the necessary information. Nevertheless, the overall scenario has some main disadvantages that cannot be overcome by changing the generating software, the analyzer software or tuning their parameters:

• The lack of centralized control: one cannot sit at single machine and perform all the needed measurements.

• It is difficult to combine the data from different successive flows. The next paragraph will discuss some of the scenarios where the use of the existing tools is becoming either impossible or additional tricks are needed in order to achieve the imposed goal.

III. OVERVIEW OF A NETWORK MANAGEMENT SYSTEM

The management architecture is the collection of managing and managed devices and of the communication of management information between them as in figure 2.

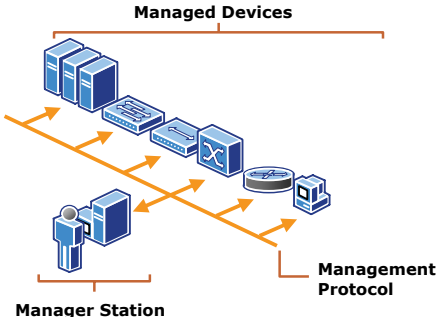

Figure 2. General architecture of a management system

When talking about the management architecture it is important to understand that, it should be related to a specific management technology. The so-called Directory Enabled Network (DEN) that keeps track of their resources by using a central repository called directory. The management applications running in such environment must identify each resource based on its information from the directory. However, in what follows it will be presented the simple, classical management architecture, which can be found in some of the most used management technologies such as the Simple Network Management Protocol (developed for IP networks) and Telecommunications Management Network (designed to comprehend a wide spectrum of telecommunication technologies) [2].

A. Managers

The managers are the nerve center of a management infrastructure that collect information about the state of the network and send configuration messages to the network elements.

Their key functions are to receive notification data for the managed devices, also known as traps, to request information from these devices or to send parameters. A manager must also allow a configuration to be set up or specify the number of retries, timeout duration or polling intervals that will be used as thresholds in order to determine that a device is not responding. The messages transmitted between a managed device or network element, on one side, and a manager on the other side, is related to any aspect regarding fault, configuration, accounting, performance and security.

B. Agents

Agents are pieces of software or hardware implementing functions that run on the managed devices. The first task (of the hardware or software routines within the managed device) is to respond to the managers' inquiries. They allow inspection and changing of the managed device parameters. They also detect any abnormal condition and report it to the manager.

Examples of agents include a specific service of a Windows® based computer, a daemon running on a Linux machine or routing software from the Cisco's IOS® operating system used on Cisco devices [3].

C. Manager-Agent Communication

This process is used by the agent to inform about the status of the managed device or by the manager to request some information. The communication is usually based on the transport provided by a management protocol specific to both the platform and managing technologies. Regardless of the implementation, two mechanisms are possible: polling-based and event-driven [4].

C1. Polling-Based Mechanism

In this case, managers request information periodically from the network elements. The advantage of this approach is that the complexity of the agents is very low.

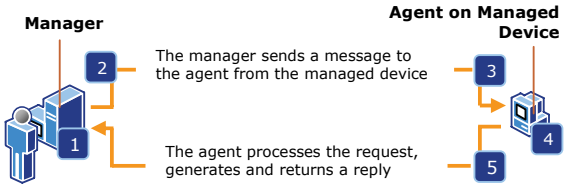

Figure 3. Polling-based communication mechanism

A debate is made on what should be an appropriate polling interval. A too small value results on having the most up-to-date information regarding the network, but wasting valuable bandwidth. With a high value, the bandwidth is saved and performance improved but the probability of missing key events or being aware of them too late increases. The communication between the manager and agent of the managed device in the example above proceeds in the following fashion (see figure 3):

1. The manager forms a query message that contains an information request.
2. The manager sends the information request to the managed device.
3. The managed device receives the message, checks the authenticity of the message and evaluates the request.
4. If the authentication data or access permission is

incorrect, the agent could send a notification.

5. The agent calls the appropriate local service to retrieve and reply the requested information.

Note that from the list of operations performed by the managed device and the agent at the third step, the actual activity depends on the management technology involved.

C2. Event-Driven Mechanism

For this mechanism, the network elements have the managing intelligence of informing the manager that something has happened. This overcomes the problem of bandwidth but requires the network element to be still operational in order to report a problem, increasing the processing and resource usage at the managed device in order to evaluate each operation parameter and to determine if a message should be sent to the manager.

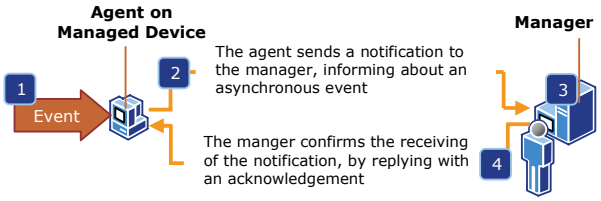

Figure 4. Event-driven communication mechanism

The communication between the manager and agent of the managed device in the example above proceeds in the following fashion (see figure 4):

1. An event occurs on the managed device.
2. The agent from the managed device creates a notification message to be sent to the manager containing the event data.
3. The manager receives the notification message and takes the appropriate action, such as a visual or audio alarm or by performing a predefined or determined operation.
4. The manager station can return an acknowledgment message.

The most used option is however the mixed one, when the event-driven messages could be sent for extraordinary events, while polling can be used at larger intervals, during normal operation.

D. Management Information

The management information is about the managed nodes and devices. In most of the implementations it is considered as a collection of physical/logical resources that can be managed, named managed objects. This abstraction is useful in hiding from the management system the specific details of that resource, others than the ones that have management importance, such as implementation, vendor or device specific extensions and access methods [5].

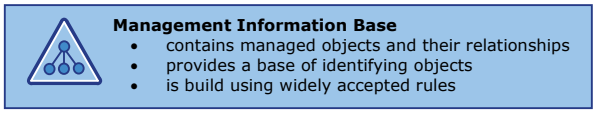

Figure 5. MIB characteristics

On the other hand, due to the wide variety of manufacturers, equipments, applications and parameters that could be managed a repository of the possible manageable information is needed, such that both agents and managers understand each other.

IV. DESIGN AND EXPERIMENTAL RESULTS

The goal of the management infrastructure is to provide an interpretation of the user's commands and then to assure the reliable transport of it to the processing agent. The first objective is done by the management console only. The user-interactive part of the software is designed to allow the setup of a measurement test within a session or session group, to schedule it, and care nothing more about it. Special services within the management infrastructure will perform the task, collect the results and make them available for display when the test is over.

The functionality of the management infrastructure is divided into several operational units called services. All components (except Session Manager and Service Control Manager) are found on the manager and the agent platforms, the overall structure being symmetrical (see figure 6). Therefore, it does not matter what functionality (management/ measurement) the application has.

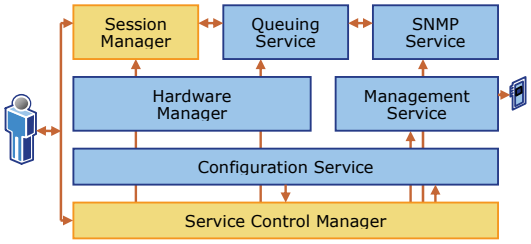

Figure 6. The architecture of management infrastructure

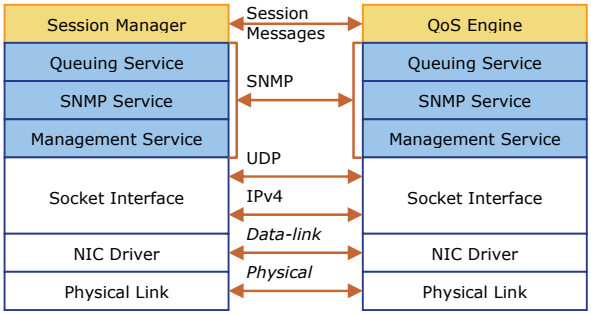

Figure 7. Communication between the management console and the measurement agent

The same architecture, but from a layered perspective, is presented in figure 7. Observe that the console features the Session Manager, whilst the agent has instead a QoS Measurement Engine.

A. Management Service

The Management Service handles networking functions i.e. transmission and reception of management packets for any selected local interface. Outbound, the destination address and port are given by the Session Manager that knows which agents are used within the test. The Management Service performs bi-directional multiplexing and de-multiplexing of the messages to be added or retrieved from the message queue. It also implements security functions by filtering the IP addresses of inbound messages, according to a user-defined list.

B. SNMP Service

The management service relies on the SNMP service in performing message encapsulation. Its two main functions

are the following: a) to create a new SNMP message, given the parameters; b) to identify, verify and extract the SNMP parameters from an incoming message.

C. Queuing Service

The task of the Queuing Service is to ensure that multiple simultaneous incoming or outgoing management messages can be processed. In addition, this service establishes priorities on the messages placed in the queue and handles retransmissions. It also ensures that duplicate messages arriving within the duplicate discarding interval are eliminated. Queuing Service performs the recycling, i.e. the messages waiting more than the permitted time are removed from the queue.

D. Session Manager

The Session Manager is the highest-level software routine of the infrastructure, implemented at the management console only and performing functions related to measurement. The user interaction means setting up a test by creating sessions (see table I), session groups (see table II) and scheduled tasks.

From the management's point of view the measurement tasks scheduled by the user are translated into appropriate SNMP messages, to be sent to the agents. When the first reply is received, a session is created. The replies could be redirected to the task that waits for them. The results for each task are stored in order to make them available for later inspection.

TABLE I. SESSIONS

Session Type	Agents Used	Description
Generation	1	It uses one agent to generate the network traffic.
Analysis	1	It uses one agent to analyze the network traffic.
Generation and Analysis	2	It uses two agents, one for the generation and one for the analysis. The session is flow-based, meaning that only the traffic generated by the first agent will be analyzed by the second one.

TABLE II. SESSION GROUPS

Session Group Type	Sessions Used	Relationship
Group of Independent Sessions	Multiple	Based on time
Group of Parameter Dependent Session	Single	Based on a user-selected parameter

From the management's point of view, the measurement tasks scheduled by the user are translated into appropriate SNMP messages, to be sent to the agents. When the first reply is received, a session is created. The results for each task are stored in order to make them available for later inspection.

E. Configuration Service

The role of this service is to store (onto magnetic media) the configuration data for all other service components of the management infrastructure. The information, available after the console restarts, is related to user-configured data but also to machine-specific data, such as the networking devices.

F. Service Control Manager

Implemented only at the management console, the Service Control Manager handles all services operation. For each service it maintains the status (started or stopped) and for some of them allows the user to change the status. The SCM also handles service recovery in the situation of a service failure, several recovery actions being possible to define.

V. EXPERIMENTAL RESULTS

This paragraph describes the experiments of testing the gigabit network interface cards, to see how NMS can be used to accomplish the task instead of using the classical approach.

In the scenario from figure 8, Chenin workstation has a NMS management console installed, and NMS agents run on both Enologa and Macabeu. On Enologa there is a SysKonnect SK9843 NIC connected via a 1000BaseSX link to the Endace DAG card installed on Macabeu. The network traffic was generated with Enologa agent while the analysis is done with the Endace card on Macabeu.

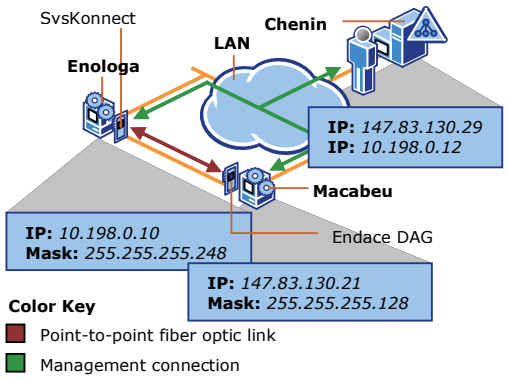

Figure 8. Testing NICs with NMS scenario

The network traffic was generated with Enologa agent while the analysis is done, of course, with the Endace card on Macabeu. Prior to testing, the configuration of the management console, measurement agents and agents' registration was completed. The flow-based session used Ethernet encapsulation for data-link test.

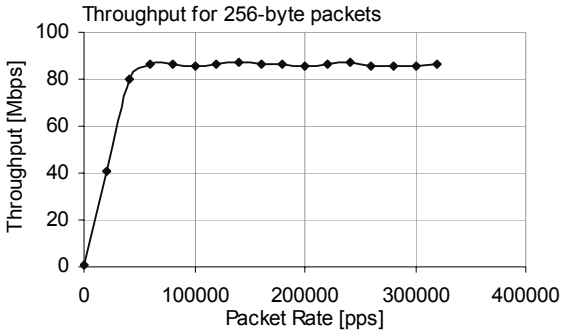

Figure 9. Average throughput for 256-byte packets

Three types of scenarios were involved: periodic, Poisson and link flooding traffic, with packet size as a parameter (256, 512, 768, 1024 and MTU – 1500 bytes). For each packet size, a session group was created to run a series of tests with packet rates going up to 1 Gbps for the

selected packet size. Let us discuss the results obtained for 256 and 1500 byte frames in the case of the periodic distribution. The interesting parameters were the average throughput and number of packets transmitted and received.

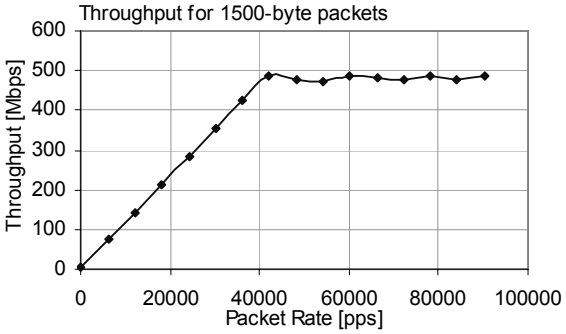

Figure 10. Average throughput for 1500-byte packets

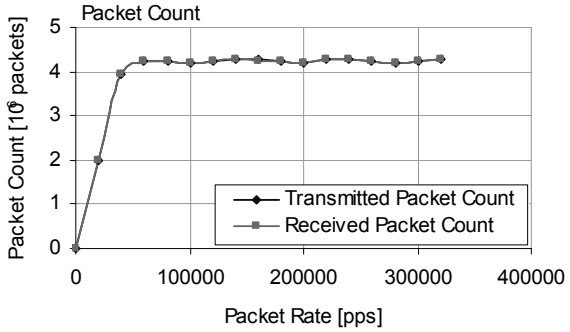

Figure 11. Number of packets transmitted and received for 256-byte frame

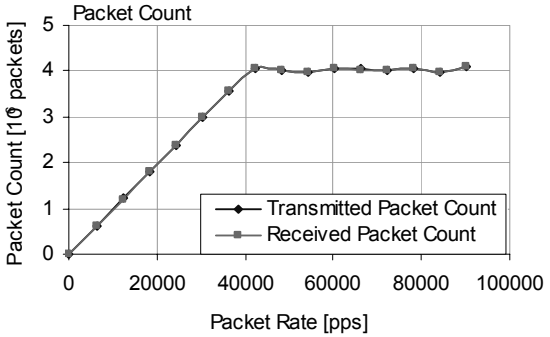

Figure 12. Number of packets transmitted and received for 1500-byte frame

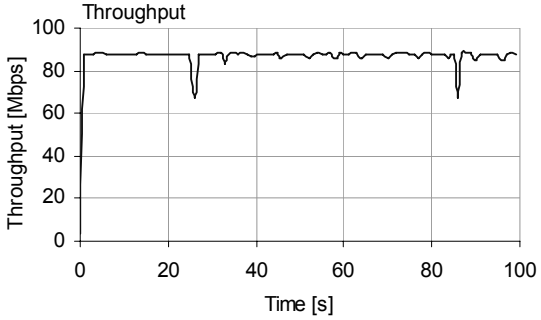

Figure 13. Average throughput for 256-byte packets and a packet rate of 160200 pps

The average throughput obtained depends largely on the packet size (figures 9 and 10), whilst the software limitation is rather related to packet rate (about 50 kpps, as in figure 11 and figure12). All four figures contain the data from the session group with the packet rate as a parameter.

In addition to the group data, session data is also available. Figures 13 and 14 show similar parameters as in the previous ones, i.e. the throughput and the number of packets transmitted, for 256-byte packets, versus time (i.e. during one session of 100 seconds).

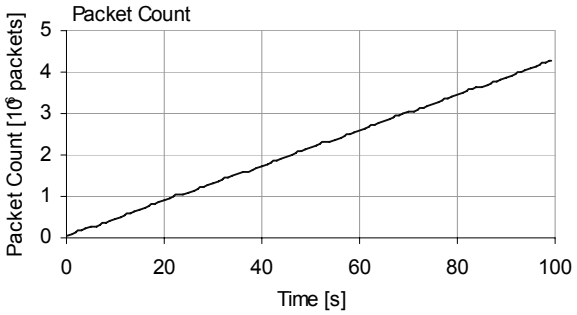

Figure 14. Number of packets transmitted for 256-byte packets and a packet rate of 160200 pps

VI. CONCLUSIONS AND FUTURE WORK

The Network Measurement System realized has several advantages compared to the existing measurement solutions. First, we can implement very quickly a wide range of measurements scenarios. Once the machines and the software are properly installed, all operations can be performed from the management console only. Since the start of a session takes just few steps in a wizard, it greatly saves a lot of time, when doing complex measurements. NMS does not require user attendance during the execution of the tests.

The management infrastructure of the Network Measurement System offers better performance for traffic generation. The user can focus on the objectives of the test, rather on how to implement them. It offers a high level of functionality. In most situations, there is no need to use third party software. However the system is not perfect, because is not completely portable across all GNU/Linux platforms. Nevertheless, since the source code is available with slight changes it can be ported to almost all major distributions. For the moment, it is not optimized for local resource usage. In addition, it does not implement a fined-grained analysis system in order to save the limited bandwidth of the management link. Further work is obviously needed to eliminate all these drawbacks and to add new features.

REFERENCES

[1] ***: “MGEN User’s and Reference Guide”, U.S. Naval Research Laboratory, 2006

[2] ***: “ISP Network Management”, Ericsson, 1999

[3] ***: “Internetworking Technologies Handbook”, Cisco Systems, 2002

[4] Douglas Mauro, Kevin Schmidt: “Essential SNMP”, O'Reilly & Associates, 2001

[5] M. Rose, K. McCloghrie: “RFC 1155 - Structure and Identification of Management Information for TCP/IP based internets”, IETF, 1988