

Survey of SNMP performance analysis studies

Laurent Andrey¹, Olivier Festor^{1,*†}, Abdelkader Lahmadi¹, Aiko Pras² and Jürgen Schönwälder³

¹LORIA-INRIA, Lorraine, France

²University of Twente, The Netherlands

³Jacobs University Bremen, Germany

SUMMARY

This paper provides a survey of Simple Network Management Protocol (SNMP)-related performance studies. Over the last 10 years, a variety of such studies have been published. Performance benchmarking of SNMP, like all benchmarking studies, is a non-trivial task that requires substantial effort to be performed well and achieve its purpose. In many cases, existing studies have employed different techniques, metrics, scenarios and parameters. The reason for this diversity is the absence of a common framework for SNMP performance analysis. Without such a framework, results of SNMP-related performance studies cannot easily be compared, extended or reused. It is therefore important to start a research activity to define such a framework. Such research activity should start from analysing previous studies on this topic to reveal their employed methods. In this survey we examine these studies by classifying and discussing them. We present techniques, approaches and metrics employed by these studies to quantify the performance of SNMP-based applications. Copyright © 2009 John Wiley & Sons, Ltd.

1. INTRODUCTION

The SNMP protocol was defined at the end of the 1980s [43]. Since then, it has been widely deployed and used for IP-based network management. In recent years various studies on SNMP performance have been conducted and many papers have been published. One of the first attempts to make a dedicated performance study of SNMP was undertaken by Pattinson [15]. This study attempts to answer simple performance questions on SNMP activities, such as health checking and fault finding. The main performance metrics used in that study were response time and the amount of SNMP traffic that needed to be exchanged.

In addition to this study, several others have compared SNMP performance to that of other management approaches. Unfortunately, several of these studies tend to be impartial in their comparison. This can be understood from the fact that these studies often aimed at proposing new, emerging approaches which, from a performance point of view, were supposed to outperform SNMP. Given the challenge of developing accurate and scalable management applications, there is therefore a need for a common, objective analysis framework that provides deeper insight into the performance of SNMP and related protocols.

In this survey we investigate the main papers that have been published to assess the performance of SNMP with regard to their techniques, metrics, scenarios and parameters. Based on this survey, we outline the following flaws that can occur in performance analysis studies:

- Results of different performance studies may not be *comparable*. For example, one study may have measured the delay of retrieving a certain MIB table using `GetNext` PDUs, whereas the other study uses `GetBulk` PDUs.

*Correspondence to: Olivier Festor, INRIA, 615 rue du jardin botanique, Villers les Nancy 54602, France

†E-mail: Olivier.Festor@inria.fr

- Experiments may not be *reproducible*, which means that other researchers trying to perform the same experiment may not be able to achieve similar results. This problem may be caused by the fact that the experiments have not been described in sufficient detail, leaving the researcher trying to reproduce the experiments with too many options. This problem may also be caused, however, by some events interfering with the experiment, which basically means that the results are not correct.
- Performance studies may not be *representative*, which means that SNMP usage in real networks may be largely different from the one assumed in a study.

We find that these flaws originate from common mistakes and lack of knowledge about performance evaluation. To avoid repetition of the same mistakes but also to allow new researchers to learn from others, this paper identifies the main literature on SNMP performance analysis and discusses the approaches taken thus far. In addition, the paper may serve as a starting point for developing a common performance analysis framework for SNMP or other management approaches.

In this survey, we have investigated the existing literature in this field. To determine whether different performance studies are comparable, we have studied the techniques that have been used as well as the metrics that have been investigated. To determine whether different performance studies are representative, we have studied the scenarios that were analysed as well as the parameters that were varied, and compared these to traces we have captured from real networks. Finally, we have studied how detailed the various tools have been described to get an idea whether a certain study would be reproducible.

The structure of this paper is as follows. Section 2 discusses the process used to find existing literature on SNMP performance analysis. Section 3 discusses the techniques (analysis, simulation or measurements) that have been used to evaluate performance. Section 4 discusses which performance metrics exist, and how these metrics were used in previous studies. Section 5 discusses the measurement scenarios, and Section 6 focuses on performance parameters. Tools are discussed in Section 7. Each of these sections is based on the well-defined terminology of performance evaluation, as described in Jain [28]. Finally, Section 8 provides the conclusions.

2. OVERVIEW OF EXISTING LITERATURE

This section first describes the method used to identify papers discussing SNMP performance. In the second part, we discuss some high-level observations concerning the motivation behind the identified performance studies.

2.1 Paper search and selection method

In order to find existing literature, we focused on two electronic databases: ACM and IEEE. We used a list of Boolean conditional keyword phrases, such as 'SNMP', 'SNMP and performance', 'SNMP and benchmarking', 'SNMP accuracy' and 'SNMP cost'. For major networking conferences, we used the keyword *SNMP* to search all SNMP-related papers and looked for papers containing some analysis of this protocol. A set of more than 1300 papers was found. We looked through the titles of the articles to eliminate the ones that were unrelated; for the rest, we browsed the abstracts to gauge their relevance. We have also 'manually' checked the online titles and abstracts of some major conferences and journals of the network and networks and systems management domain. Some papers were also known a priori by the authors or come from the transitive closure of the bibliography of the studied papers.

Most of articles were retrieved from online paper sources such as IEEE Xplore,¹ the ACM Digital Library,² and the DBLP Computer Science Bibliography.³ We have focused on two main kinds of publications:

¹<http://ieeexplore.ieee.org>.

²<http://portal.acm.org>.

³<http://dblp.uni-trier.de/>.

	Total	97	98	99	00	01	02	03	04	05	06
ECCM	1	0	0	0	1	0	0	0	0	0	0
CM	1	0	0	0	0	0	0	0	1	0	0
DSOM	2	0	0	0	0	0	1	0	0	0	1
ETNSM	1	0	0	0	0	0	0	0	1	0	0
ICM	1	0	0	0	0	0	0	0	1	0	0
ICSE	1	0	1	0	0	0	0	0	0	0	0
IJNM	1	0	0	0	0	0	0	0	1	0	0
IM	4	1	0	1	0	1	0	0	0	1	0
JNSM	1	0	0	0	0	0	1	0	0	0	0
JSAC	1	0	0	0	0	0	1	0	0	0	0
MMNS	1	0	0	0	1	0	0	0	0	0	0
NOMS	5	0	0	0	1	0	0	0	3	0	1
SPC	1	0	0	0	1	0	0	0	0	0	0
TSE	1	0	1	0	0	0	0	0	0	0	0

ECCM, *Elsevier Communication Journal*; CM, *IEEE Communication Magazine*; DSOM, *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*; ICM, *IEEE Internet Computing Magazine*; ICSE, *IEEE/ACM International Conference on Software Engineering*; IJNM, *International Journal of Network Management*; JNSM, *Journal of Network and Systems Management*; JSAC, *IEEE Journal on Selected Areas in Communications*; MMNS, *IFIP/IEEE Management of Multimedia Networks and Services*; NOMS, *IEEE/IFIP Network Operations and Management Symposium*; SPC, *ACM/IEEE conference on Supercomputing*; TSE, *IEEE Transactions on Software Engineering*.

Table 1. SNMP performance analysis papers

- Publications in conferences and workshops devoted to network management (IM, NOMS, DSOM, MMNS, IPOM) and other major networking conferences such as INFOCOM, SIGCOMM, SIGMETRICS, GLOBECOM, ICC, which include tracks on the network operations and management research or computer performance analysis.
- Publications in journals devoted to network management such as *IEEE Transactions on Network and Service Management* (TNSM), *Journal on Network and Systems Management* (JNSM), *International Journal on Network Management* (IJNM) and some major networking journals such as *IEEE Journal on Selected Areas in Communications* (JSAC), *ACM Transactions on Networking* (TON), *IEEE Computing Magazine*, *Elsevier Computer Communications Journal*, and *IEEE Communications Magazine* series on Network and Service Management.

We also surveyed articles in the *Simple Time* newsletter.⁴ *Simple Time* offers a good overview of SNMP concerns and changes over time.

After eliminating duplicated papers, 22 articles were selected for a full text review. The articles in our survey should be regarded as representative of the work related to the topic, but not in any sense as a ranking. While writing this survey, we also read many other articles in addition to the 22 selected papers and we cite some of them through our work. A number of articles in the list of selected papers ranges across the boundaries of different management protocols, either because they compare SNMP performance to other protocols or because they discuss a hybrid mixture of SNMP with other protocols (e.g. SNMP and XML-based management, SNMP and Web Services).

Table 1 summarizes the sources of reviewed papers and their distribution over years.

2.2 Motivation and goals

In this section we analyse the selected papers according to: (1) the research objective behind each study; (2) the considered performance evaluation approach; (3) the related SNMP features under test (the

⁴<http://www.simple-time.org>.

distribution model, data model used, management data access model and SNMP operations models); and finally (4) the options in considering management nodes communications.

SNMP performance inefficiency as a motivation behind management algorithms optimization research

In major networking conferences (INFOCOM, GLOBECOM, SIGCOMM, IMC), the performance of SNMP was stated, without an in-depth analysis, since it is a protocol readily available for network measurement and anomaly detection [40]. Most of the papers published in conferences denote the inefficiency of SNMP to collect fine-grained measurements due to its overhead on routers and links. This assumption has been one of the main motivations behind many research papers [41,42] targeted toward optimizations of network measurement algorithms to infer and diagnose network properties. However, these papers do not really assess and justify SNMP weaknesses.

Some papers discuss issues linked solely to SNMP features, improvements of SNMP features, or SNMP implementation specifics. Usually some kind of performance evaluation supports the discussion. An old and much discussed matter is tabular data and large data retrieval [52,53]. The case of Chan and Chan [3] is typical: there a new `get-rows` SNMP operation is proposed and a performance evaluation section supports the proposition. Zhang *et al.* [23] also target performance aspects of SNMP, but focus on agent's implementation. This paper proposes the use of multi-path tree and AVL-trees to improve object lookup from object identifiers (OIDs). A small performance comparison with hash tables, which is supposed to be the usual way to accelerate object lookups in agents, is done.

Comparing SNMP performance with other management protocols

The majority of articles compare SNMP to other management protocols (e.g., CORBA, XML-based protocols) or they contain a performance analysis of SNMP for managing specific environments such as grids, clusters, databases, CDMA PCS networks, satellite networks and even earth observer system networks [36].

Baldi and Picco [2] compare the overall management traffic generated for information retrieval by SNMP against a variety of mobile code or mobile agents approaches. The comparison is a function of affecting parameters, including the dimensions of the managed network (number of managed devices), complexity of the task (number of so-called *micro-management* queries needed to be performed per node and per management task), transmission overhead, frequency of code invocation and data size. The network was assumed to have 50 devices, with 30 queries on each. They state that traffic needed for management using mobile agents is only one-third of that of the SNMP implementation, which confirms that the mobile agent paradigm can be cost effective for the management of networks. In Gu and Marshall [21] the performance of SNMP and CORBA is studied and compared with a single agent and a single manager. Their results show that SNMP costs less network bandwidth and has a lower latency than CORBA when manipulating small amounts of data (e.g. single object, small table). Manipulating large tables is another typical management task. In this type of task, CORBA consumes much less bandwidth and experiences less delay than SNMP. Holt *et al.* [24] also compare mobile agents for retrieving managed objects over many nodes from one single manager. This paper gives some analytical formulas for overall delay for this retrieval at the manager side. It also studies the impact of *clustering*: the use of several parallel mobile agents.

Pavlou *et al.* [11] and Pras *et al.* [12] compare SNMP protocol usage based on polling to web service-based management. Their results show that web service protocols affect bandwidth consumption much more than SNMP. However, when using compression and when a large number of objects is retrieved, Web Services demand less bandwidth than SNMP. Generally, the three studies state that when a single object is retrieved SNMP is more efficient than Web Services. In cases where many objects are retrieved, Web Services have been shown to be more efficient than SNMP. Bandwidth consumption and delivery delay are compared between Web Services and SNMP-based notifications in de Lima *et al.* [6]. Their results are inconsistent with those of Pavlou *et al.* [11] and Pras *et al.* [12], where the experiment demonstrates that Web Service notifications perform better than SNMP for a large number of objects. Therefore, we point out that this situation needs to be verified from real management traces to know how many

objects are usually carried by SNMP messages. There are more papers about Web Services or XML-based protocols for network management, which typically feature a performance evaluation section using standard SNMP interactions as references. Good examples of such papers are Soldatos and Alexopoulos [25] and Alexopoulos and Soldatos [26], but their performance analysis sections are not necessarily good examples of best performances analysis practices. Their performance evaluation study is partial, lacks detail on their experiments, and their results stay limited. Their evaluation scenario uses a single operation without any traffic injector behind. They just repeated the experiments with multiple individual requests which is not in any way representative of a management application. They also used a single metric to compare the performance of the different management approaches. This usually limits their results since a single metric is not representative of the performance of a system.

In an effort to investigate the security overhead in network management, Corrente and Tura [5] and Du *et al.* [16] assess the performance of SNMP with security support. In Du *et al.* [16] the authors implement a prototype of SNMP on a TLS/TCP base. In Corrente and Tura [5] the authors compare the built-in security support of SNMPv3 to unsecure SNMPv2c. The two studies showed that security is introducing a clear but not excessive degradation of performance. Similar to the idea of securing SNMP with a third-party security protocol as expressed in Du *et al.* [16], Marinov and Schönwälder [10] quantify the performance of SNMP over SSH. Although the two studies used a very different setup, some key results are similar. Their results show that SNMP over TLS or over SSH is more efficient in terms of latency than SNMPv3/USM for longer sessions. The study of bandwidth consumption shows that SNMPv2/SSH requires less bandwidth than SNMPv3/USM/UDP.

There are no standard criteria to evaluate the effectiveness of the SNMP protocol

Given that the SNMP protocol is well specified [43] and many commercial as well as open source implementations exist (net-snmp),⁵ it was surprising that we did not see any agreement on conventions for evaluating its performance. Even the definitions of performance metrics used by the surveyed articles are often inconsistent and confusing. For example, network overhead due to SNMP activities is called *bandwidth utilization, usage, traffic volume, management traffic* or *network usage* in different studies.

As a result, no common foundation has been established to evaluate the performance of the SNMP protocol so far. In 2002, Henrick E. Holland [27,46] performed a study on the evaluation of MIB-II implementations of the SNMP protocol on some routers (Cisco AGS+, Cisco C2600 and Cabletron ssr2000). The study is interesting from a methodology point of view, since the author identifies some common scenarios to evaluate the performance of SNMP, but still lacks more performance evaluation features. Mainly, it lacks a well-defined set of metrics and more representative management scenarios.

In 2004, Pras *et al.* [12] studied the performance of the SNMP protocol within a polling scheme. They mainly assessed resources usage (CPU and memory), bandwidth usage and round-trip delays at the packet level between managers and agents. In 2006, de Lima *et al.* [6] studied the performance of SNMP within a notification scheme. They followed closely the same metrics and measurement methodology as in Pras *et al.* [12] to assess the performance of SNMP's notification mechanism. These two articles can be seen as a first attempt for a common evaluation scenario of the SNMP protocol performance. However, we believe that the management community still needs a unified framework for assessing measurements of SNMP protocol performance.

3. TECHNIQUES

The three main techniques for performance evaluation are analytical analysis, simulation and measurements (benchmarking) [28]. Most of the reviewed papers use a combination of measurements and quantitative analysis (see Table 2). We also found articles dealing with SNMP modeling using some well-known modeling tools such as Petri Nets, UML diagrams [37] and queueing theory [44], or even an SDL

⁵<http://net-snmp.sourceforge.net>.

Technique	Used times
Analytical	7
Simulation	3
Measurement	15

Table 2. SNMP performance evaluation techniques used

specification [38] of the SNMP protocol. We have also found a Tcl/Tk-based multi-platform SNMP agent simulator [39].

We have observed that the majority of papers use synthetic workloads for measurement-based SNMP performance evaluation. One reason for this is that no real-world workload publicly exists for SNMP applications, despite some recent attempts to collect such workloads [51]. Thus, to generate their workload, most authors use their own injectors with their own workload models because no standard benchmarking suites exist for SNMP performance assessment.

3.1 SNMP simulation models

Few studies use simulations for SNMP performance analysis. Rubinstein *et al.* [18] investigate the effect of a large number of nodes on the performance of an SNMP-based management system using simulations. Pattinson [15] investigates the performance of SNMPv1 with simulation using the OPNET package. His results shed some light on understanding the behaviour of SNMP in health check and fault-finding applications. However, despite the fact that his paper is very early work in the field, we raise some questions over its simulation model compared to simulations given in Kantorovitch and Mahonen [50]. Normally, SNMP follows the manager-agent pattern that looks like a client-server model, where the client is the manager and the server is the agent. However from a data flow view, the manager-agent mode of operation has a data flow different from the typical client-server process. In particular, it is common to have a small number of clients (managers) and a large number of servers (agents). However, in the OPNET package, the model requires a single server and many clients. To resolve this issue, Pattinson in his simulation work based on the OPNET simulator inverses the roles, and the manager was the server and the agents were the clients. Thus his model produces SNMPv1 responses before the associated request, since agents take on the role of clients. Kantorovitch and Mahonen [50] also use the OPNET simulator; however, they respect the traditional roles of the manager-agent pattern mapped as a client-server model.

4. METRICS

This section first introduces and classifies some possible SNMP performance analysis metrics before analysing the metrics used in published papers.

4.1 Possible metrics

For a performance study, a set of performance criteria or metrics must be chosen. The IP Performance Metrics (IPPM) framework [49] proposes that these metrics must comprise a small set, well defined and reported by a single numeric quantity. In the SNMP protocol, there are several quantities related to its performance and reliability that we would like to know the value of. We classify the set of these metrics into three categories, as shown in Table 3.

The *speed* metrics deal with management data access either locally or remotely. These metrics capture the quantity and related timing of management data under some management activities (for example, issuing SNMP queries or updating operational statistics counters on a router). More specifically, for an

Metric	Speed	Cost	Quality
Number of collected/adjusted variables	X		
Number of traps	X		
Delays	X		
CPU usage		X	
Memory usage		X	
Bandwidth usage		X	
Estimate accuracy			X
Timing accuracy			X
Loss			X

Table 3. SNMP performance metrics classification

SNMP-based statistical monitoring algorithm, speed metrics measure the number of collected statistics and their transfer delays to a monitoring station. Other metrics might be derived from these speed metrics according to the measurement objectives and the management application requirements. For example, we derive from the delay metric the one-way delay for unconfirmed notifications and a round trip delay for confirmed polling operations.

As depicted in Figure 1, the end-to-end delay between an SNMP manager and its agent is divided into many components. We enumerate delay components with numbers from 1 to 7. The end-to-end delay is defined by the number 1. It represents the delay that management operations experience when called by a management algorithm running on the manager. The components numbered 2, 3 and 4 represent respectively PDU processing delay, message processing delay and security processing delay. The processing delay is the sum of the three delays. A measurement process has to define which delay component it is measuring.

The *cost* metrics are divided into three categories. The storage cost refers to the amount of management data stored at the agent. The communication cost refers to the total amount of information that needs to be transmitted through the network between two or many SNMP entities. The computational cost reflects the processing activity at an SNMP entity.

The *quality* metrics deal especially with the spatial and temporal errors of management data. By spatial error, we mean that we obtain an error metric per management attribute (OID) that summarizes its deviation from the real value of the manageable attribute. The temporal error gives error information per time slot (a monitoring round, for example) summarizing the time deviation from that instant while acting over a management attribute (collect, notify, update). We note that a fundamental trade-off exists between the three categories of metrics.

In practice, we often find that an SNMP performance metric yields an important variability. Therefore, although the average provides a useful indication, it is more informative to characterize the measurements by means of an *empirical cumulative distribution function* (ECDF) [49]. We also point out that when a metric is specified a given measurement method might be noted and discussed.

4.2. Metrics in published papers

The most commonly considered performance metrics of SNMP-based management systems are bandwidth usage and delays, called response times; the former is calculated in more studies than the latter (see Table 5).

In Gu and Marshall [21], delays are divided into two parts: delays caused by network communication and local delays caused by computer computation. This is helpful for isolating and locating performance bottlenecks. As stated in Pras *et al.* [12], we point out that local data retrieval delays from within the managed resource on agents are usually more expensive than encoding and decoding delays. Pras *et al.* [12] found that the amount of CPU time for coding a BER-based SNMP message is three to seven times less than an XML-based encoding. Generally, the burden placed by SNMP on the processors of network

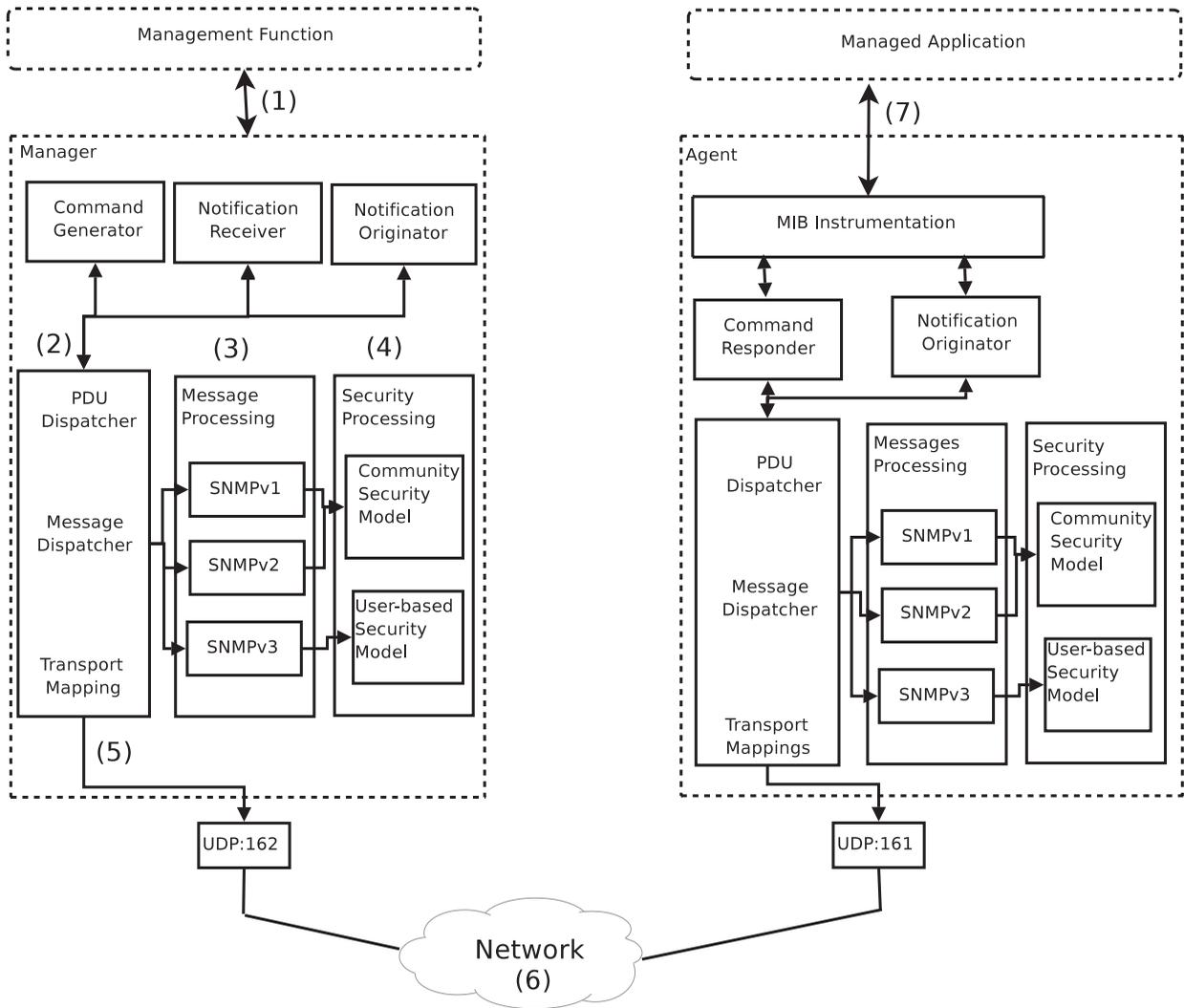


Figure 1. Different components contributing to the end-to-end delay within the SNMP architectural model

equipments and their links mainly depends on the parameters used by the management application (polling versus notification, request frequencies, number of retrieved objects).

Barford *et al.* [1] consider accuracy for certain measurement techniques based on SNMP as a primary performance metric. This study compares probe-based and router-based methods for measuring packet loss and examines the accuracy of SNMP loss measurements in both a controlled laboratory experiment and in a live network. Accuracy is perceived by management algorithms to react in case of problems and hence is considered as a *quality* performance criterion.

An important step in every performance study is the performance metrics measurement approach. To measure the delay of SNMP operations at the application level, most studies [10,12,16] used the POSIX *gettimeofday()* function. The accuracy of *gettimeofday()* appears to be good in the microsecond range. However, the resolution of this function depends on the hardware architecture. Intel processors as well as SPARC machines offer high-resolution timers that measure microseconds. Other hardware architecture falls back to system timers, which are typically 100 Hz (equals 10 ms). Corrente *et al.* [5] used the Time Stamp Counter of the Intel Pentium processor for time measurement. The value of the counter can

be read using the RDTSC assembler (Intel) instruction [47]. This method is more accurate than *gettimeofday()* and has very low overhead. At the network level, the *tcpdump*, *ethereal*, *snoop* tools were used to calculate the number of bytes exchanged and packet timing data. The *tcptrace* tool was used in Kantorovitch and Mahonen[50] to analyse SNMP captured traces. Memory usage was measured using *ps* utility, *dmalloc* library or the *pmap* tool.

5. SCENARIOS

This section introduces possible organizational scenarios and afterwards discusses which scenarios have been used by performance analysis papers and how the selection of scenarios may relate to operational networks.

5.1 Possible scenarios

In the following discussion, *M* stands for an SNMP manager, *A* stands for an SNMP agent and *I* stands for some kind of interworking device. Figure 2 shows some possible fundamental organizational scenarios.

- (a) *M-A*: In the simplest scenario, a single manager communicates to a single agent.
- (b) *M-I-A*: This scenario consists of a single manager and a single agent with an interworking device between them. The interworking device might be an intermediate-level manager (in the case of a distributed management scenario) or a gateway (used for example to convert between SNMP and other technologies like Web Services).
- (c) *M < A*: This scenario assumes a single manager communicating to a larger number of agents. An example is a management system collecting performance statistics from all routers in a backbone network.
- (d) *M > A*: This scenario assumes many managers communicating to a single agent. An example is an agent configured to deliver event notifications to several different management systems or a shared router monitored by different customers.

In operational networks, combinations of these four fundamental scenarios are likely to be found.

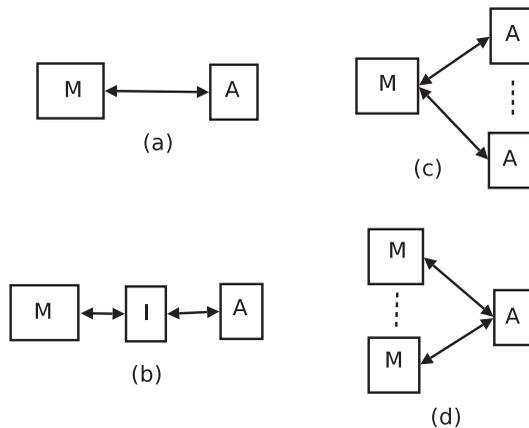


Figure 2. Fundamental scenarios of SNMP-based management (*M* denotes a manager, *A* denotes an agent, and *I* denotes an interworking device)

5.2 Scenarios in published papers

Table 4 shows how often each scenario is used in the studied literature.

SNMP performance analysis studies commonly use the centralized $M-A$ pattern as a management distribution model. The performance analysis of this model raises the question of how we can infer the performance of a $M < A$ scenario from the performance of the $M-A$ scenario. If the underlying protocol is UDP, the inference might be possible as long as no congestion appears; however, for TCP, the inference might be more difficult since TCP incurs more communication overhead than UDP and needs more resources while opening TCP sessions to a large number of agents. Some studies [6,22] use the $M-I-A$ scenario for comparing the Web Services to the SNMP management approach, where the gateway acts at the protocol level that directly maps SNMP primitives to Web Services operations, or at the object level that offers operations that reflect the structure of the management information. From their testing, the authors concluded that object-level gateways are more interesting than SNMP when the amount of retrieved objects is high. The hierarchical SNMP distribution model was analysed in Subramanyan *et al.* [14], where the performance of a centralized and distributed SNMP are compared in terms of delays involved in performing a designated monitoring task. The results show that the SNMP distributed model performs better than the centralized model. This result is rather obvious and can be easily explained [4].

It is well illustrated that the number of managed devices and queries, the request frequency and the dimension of replies can drastically alter the results. From the studies, we note that SNMP with or without security features performs better compared to other approaches when the managed network dimensions (MIB object size and network size) are *small*. As the network dimensions grow larger, other approaches (Web Services with compression, SSH or TLS security versus *built-in* mechanisms) will perform better in terms of bandwidth consumption and even latency. Therefore, we argue that it would be fairer if two larger managed networks (a grid, for example) and architectures, one using SNMP and the other with another approach (e.g., Web Services or CORBA) respectively, were compared. But this is never done.

The degree of remoteness [15] of an agent node from the manager node appears as to be an important factor influencing the performance of the SNMP protocol. SNMP response times in dynamic networks topologies (P2P, wireless, mobile IP) depend on the distance between the manager and the particular agent [50]. Formally, the degree of remoteness can be denoted by a parameter $r \in \{1, 2, \dots, n\}$, describing the distance between the manager and a particular agent, where n is the number of hops separating a manager and an agent nodes. A more sophisticated definition of remoteness can be found in Kwak *et al.* [45].

5.3 Scenarios in operational networks

Schönwälder *et al.* [51] analyse collected SNMP traffic to derive typical SNMP usage scenarios in real networks. The study reveals that the most frequently used scenario is the manager to multiple agents ($M < A$) scenario. The number of agents varies between two to hundreds of managed elements. The number of messages exchanged between a single manager and a single agent can vary widely and the distribution of the messages per minute over many agents seems to be long tailed, i.e. relatively

Scenario	Used times
$M-A$	9
$M-I-A$	2
$M < A$	6
$M > A$	3

Table 4. SNMP-based management scenarios used by reviewed papers

few agents contribute to most of the SNMP traffic, while there is a large number of agents that communicate with low frequency. Some cases were also found where a single agent is attached to multiple managers.

6. PARAMETERS

This section discusses parameters used in simulation or measurement studies and summarizes recent work to identify parameters used in operational networks.

6.1 *Parameters in Published Papers*

Performance studies usually make some assumptions or simplifications. These assumptions can be explicit or not. Our analysis of the papers reveals that performance studies often fail to cover the whole SNMP parameter space that can impact the results, and authors often fail to assess how much of this parameter space is covered. The parameters used by the surveyed paper are listed in Table 6.

In comparison studies, authors often conclude that the trade-off between the performance of SNMP and the performance of another management approach depends on the characteristics of the network being managed (costs, number of nodes, protocols) and the management task (expected frequency, complexity of task, dimension of management data). The performance of SNMP depends on its usage pattern and on the managed network characteristics. Ongoing work in the Network Management Research Group (NMRG) of the Internet Research Task Force (IRTF) to collect SNMP traffic traces from different operational networks [54,55] might lead to insights about SNMP usage patterns (polling frequencies, trap-polling interactions, dimensions, etc).

The most frequently used management data is the interfaces table from the IF-MIB module [12,17,18]. The choice of management data is often related to the SNMP operation being analysed. MIB tables are chosen when `GetBulk` or `GetNext` operations are analysed. A single variable is used [7,13,19] from a specific MIB when a single SNMP operation (`Get`) is analysed to retrieve one object at once. Few studies used proprietary MIB modules in their SNMP performance analysis. Corrente and Tura [5] use a synthetic proprietary MIB module, which is a mixture of scalars and tables that allow the use of the most frequent SNMP primitives (`Get`, `GetNext`, `GetBulk` and `Set`). We point out that the approach of using a synthetic MIB rather than a standardized one will be of benefit to analyse accurately the impact of management data on SNMP performance and allows a large coverage of the most frequent MIB data types. The content of such a synthetic MIB module can be derived from Schönwälder [48]. On the other hand, it must be noted that the time spent in instrumentation code, namely (1) in Figure 1, can be significant and may be a good reason for using real objects instead of synthetic objects.

Most SNMP performance studies use an IPv4 network running over 10/100 Mbits Ethernet local area networks. Most of the time, the network is considered to behave uniformly with no background load and is supposed to be error free. The network therefore introduces a constant delay for each message. However, large wide area, wireless and mobile SNMP applications response times will suffer from low bandwidth. In these networks, the communication network may indeed become a bottleneck, especially when a large amount of management data is gathered or when a large number of agents are involved within a management task and contending for bandwidth. Unfortunately, very few attempts have been made to assess the performance of SNMP in constrained environments. The impact of packet loss on the SNMP latencies was partially addressed in Marinov and Schönwälder [10]. Their testing shows a significant latency increase for lossy networks either for TCP or UDP. The authors also state that TCP transport outperforms UDP in lossy networks. However, this statement is based on the NET-SNMP implementation and should be verified with other SNMP implementations that might use more elaborate retransmission algorithms.

Kantorovitch and Mahonen [50] analyse the performance of the SNMP protocol in wireless networks in different scenarios, specifying particular wireless channel state conditions (good/bad) and background

Basic choices					
Study	Technique	SNMP approach	Heterogeneity	Security	Performance metrics
[9]	Measurement	SNMPv2, manager/agent, compared to JAVA-based agent	CMU-SNMP	None	Mean response time measured at the manager
[2]	Analytical (quantitative)	SNMPv12, manager/50 agents, compared to mobile agents	None	None	Overall management traffic
[8]	Analytical	SNMPv12, manager/N agents, compared to mobile agents	None	None	Overall management traffic
[13]	Measurement	SNMPv1, manager/100 agents, compared to mobile agents	None	None	Overall management traffic
[14]	Measurement	SNMPv2, hierarchical static, 1–3 manager levels, 4–64 ILM, 1024 agents	Hardware	None	Delay, memory, cpu, number of packets/s
[4]	Analytical	SNMPv1, centralized, hierarchical	None	Message digest and encryption	Management traffic, memory on agent, cpu utilization on manager and agent, delay to detect variable change on the manager
[15]	Analytical and simulation (OPNET)	SNMPv1, centralized	None	None	Management traffic, response times at the manager
[16]	Measurement	SNMPv1,v2c,v3 (UCD-SNMP), one manager/one agent	None	TLS/TCP and SNMPv3/TCP with USM	Mean session times, session setup time
[17]	Analytical and measurement	SNMPv1, centralized, compared to mobile agents	None	None	Bandwidth usage (kbytes)
[18]	Measurement and simulation	SNMPv1, centralized, compared to mobile agents	AdventNet SNMPv1, UCD-SNMP	None	Mean bandwidth utilization (bytes), mean response time
[19]	Measurement	SNMPv1,v3, manager/APSSNMP agent, compared to mobile agents	None	Authentication (HMAC-MD5-96), encryption (DES)	Mean processing time
[20]	Measurement and simulation (SSFNnet)	SNMPv1, compared to mobile agents	SNMP's CISCO router	None	Bandwidth usage, dropped requests, mean response times

[21]	Measurement	SNMPv1/v2, one manager/one agent, compared to CORBA-based management	SNMPv1 compared to SNMPv2	None	Traffic volume (bytes), delays (local and remote)
[5]	Measurement	SNMPv3/v2c (AGENT++ API), manager/one agent	None	Authentication (HMAC-MD5-96), encryption(CBC-DES)	Primitives elaboration time, number of transactions per minute, CPU usage, protocol overhead, network capacity consumed
[1]	Measurement	SNMPv1 compared to probe-based measurement, one manager/many agents in live network	None	None	Measurement accuracy
[22]	Measurement	SNMPv1/v2, WS gateway/SNMP agent, compared to web service-based management	None	None	Bandwidth consumption (kbytes)
[11]	Measurement	SNMPv1/v2, manager/agent, compared to web services and Corba based managements	Implementation: Net-SNMP and Advent-Net	None	Mean response time, management traffic (bytes)
[12]	Measurement and quantitative	SNMPv2c/v3, manager/23 agent, compared to web services based management	Implementation	None	Bandwidth usage (octets), cpu time, memory usage (kbytes), round trip delay
[7]	Measurement	SNMP, decentralized (Mbd) TLM/MLM/agent, compared to web services-based management	Implementation (NET-SNMP, UCDD-SNMP)	None	Bandwidth consumption (kbytes), execution time
[6]	Measurement and quantitative	SNMPv1,v2c manager/agent, notification compared to web services-based management	None	None	Network usage (octets), delivery delay (network level)

Table 5. Basic choices of performance benchmarking on SNMP protocol

Performance parameters										
Study	# sites	MIB modules			Polling frequency	Workload		Communication	Management function	Interaction mode
		# objects	# objects/request	# objects/request		Operations type				
[9]	4 nodes and 1 manager	system and ip groups	1 oid/request	1 oid/request	N/A	Single	Ethernet LAN (10 Mbits)	Data collection	Polling-driven	
[2]	50 nodes * 30 interfaces and 1 manager	Interface load level: variable	1 oid/query	1 oid/query	N/A	Single	15 LANs	Data collection	Polling-driven	
[8]	1 manager/N agents	Load on every network interface	1 oid/query	1 oid/query	N/A	Logical	Ethernet LAN	Data collection	Polling-driven	
[13]	1 manager/100 nodes	HP-UNIX MIB: computerSystemFreeMemory.0, computerSystemUserCPU.0, processNum.0, computerSystemCPU.0	1 oid/query	1 oid/query	5 s	Single: get-request	10 Mbits Ethernet	Data collection	Pull-driven	
[14]	128 nodes: 1024 emulated agents	cpu load variable	1 oid/query	1 oid/query	2-3 s	Single	Ethernet LAN	Data collection	Pull-driven	
[4]	1 manager/N nodes	1 variable	1 oid/query	1 oid/query	Dynamic	Single	Non-uniform network	Data collection, data searching	Pull-driven	
[15]	1 manager/120 agents (2 interfaces * 5 servers)	Interface utilization, IP throughput, packet loss and error statistics at IP and TCP levels	1 oid/query	1 oid/query	60-450 s	Logical	4 Ethernet LANs 10 Mbits interconnected with 1.544 Mbits	Data collection (health check, fault finding)	Pull-driven	
[16]	1 manager/1 agent	System.sysName, SYSTEM object	1 oid/query	1 oid/query	None	Single (Snmpget, Snmpwalk)	Ethernet 10 Mbit	Data collection	Pull-driven	
[17]	1 manager/50 agents	MIB-II interfaces table	21 oid/query	21 oid/query	10-40 s	Single (get-next)	Ethernet LAN: 10 Mbits, WAN: 10 Mbits and 64 kbits	Data collection	Pull-driven	
[18]	1 manager/250 agents	ifnErrors from MIB-II	1 oid/query	1 oid/query	N/A	Single	Ethernet LAN: 10 Mbits, Internet topology: 2 Mbits	Data collection	Pull-driven	

[19]	1 manager/ 1 agent	sysContact.0	1 oid/query	N/A	Single (get-request and set-request)	Ethernet LAN: 10Mbits	Data collection and update	Pull-driven
[20]	10 managers/ 1 agent	N/A	1 oid/query	Variable: 1-5-10 s	Single	Ethernet LAN	Data collection	Pull-driven
[21]	1 manager/ 1 agent	1 object, 3 objects and synthetic BGP table (1-2000 rows)	1 oid/query, 3 oids/query, 25 oids/query	N/A	Single (Get, Get-next, Get-Bulk)	Ethernet LAN	Data collection	Pull-driven
[5]	1-2 managers/ 1 agent	Synthetic MIB: mixture of scalar and table values	1 oid/query, 10 objects/query (getbulk)	N/A	single (getRequest, getbulkRequest, setRequest, Walk)	Ethernet LAN 10 Mbits	Data collection	Pull-driven
[1]	One measurement station per router, 12 routers, 30 interfaces per router	MIB-II ifTable nad ifXTable (Ingress and egress packet counts, interface drop counts and error counts), Cisco-specific MIB	N/A	30 s	Single	Abilene/Inernet2 backbone (USA)	Data collection	Pull-driven
[22]	1 manager/ 1 SNMP device	Synthetic MIB (wsTable with one column, variable rows)	1 oid/query	N/A	Single (Get-Next)	Ethernet LAN	Data collection	Pull-driven
[11]	1 manager/ 1 agent	TCP MIB (TCP connections) with 40 rows	1 oid/query (get), 8 oids/query (get), 1 MO/query (get-Next), N MO/query (GetBulk)	N/A	Single and logical (multiple)	Ethernet LAN	Data collection	Pull-driven
[12]	1 manager/ 23 agents	MIB-II (ifTable) with 10 rows	1 oid/query (get), 6 oids/query (get), N MO/query (getBulk)	N/A	Single and multiple	Ethernet LAN 100 Mbits	Data collection	Pull-driven
[7]	1 TLM/1 MLM/1 SNMP device	one MIB-II object (sysUpTime)	1 oid/query (Get)	0.5 s	Single (get)	Ethernet LAN 10 Mbits, no background traffic	Data collection	Pull-driven
[6]	1 manager/ 1 WS gateway/ 1 SNMP device	Linkdown traps (ifIndex, ifAdminStatus, ifOperStatus), routing table MIB (ipCidrRouteDest, ipCidrRouteMask, ipCidrRouteIifIndex, ipCidrRouteNextHop)	N objects/trap, $N \in [1:100:600]$, 1 object/trap and m objects/query $m \in [5:25:200]$	N/A	Single traps, single trap/multiple requests	Ethernet LAN 100 Mbits, no background traffic	Data collection	Trap-driven, trap-pull driven

Table 6. Parameters of performance benchmarking on SNMP protocol

traffic load. From their testing, SNMP response times become longer under bad wireless link quality and an increasing network load.

6.2 Parameters in operational networks

Schönwälder *et al.* [48,51] have analysed SNMP parameters used in real networks. In Schönwälder [48], SNMP MIB modules defined by standardization bodies like the Internet Engineering Task Force (IETF) or the ATM Forum (ATMF) and MIB modules defined by manufacturers like Cisco and Juniper have been analysed. The study reveals that the most frequently used data types in all MIB modules are the Integer32 and the Counter32 types. The Enumeration and Unsigned32 data types are also heavily used. This shows that counters play an important role in MIB modules. The study also reveals that the maximum level of access of 60% of MIB objects is read-only. The tables in 50% of the analysed modules are indexed by a single column and less than 20% are indexed by three or more columns. These are useful characteristics of SNMP MIB modules that can serve as parameters in performance simulations.

In Schönwälder *et al.* [51] the authors analyse SNMP MIB pattern usage in real networks. They find that SNMPv1 and SNMPv2c are widely deployed versions, despite the fact that SNMPv3 is the current full standard and SNMPv1 and SNMPv2c are both historic IETF standards. They find that the overall SNMP traffic is dominated by `Get`, `GetNext` and `GetBulk` operations and their responses. The study confirms that the polling interaction mode contributes to more traffic than the notification mode. The dominant operations often contain only one varbind in their varbind list. Thus, in a real network, table retrieval is surprisingly often realized in column-by-column mode. Furthermore, it was found that response messages fit well the Ethernet maximum transmission unit (MTU).

7. WORKLOAD INJECTORS AND SIMULATION TOOLS

This section first discusses the need for appropriate measurement and simulation tools and then reviews how published papers have dealt with this issue.

7.1 Possible tools

According to the chosen evaluation technique, tools are needed to inject or to play the measurement workload against the system under test.

For a simulation-based measurement technique the most widely used tools are the open source Network Simulator (NS-2) [30] and the commercial package OPNET [31]. The main differences between them are the good documentation of OPNET and its well-designed graphical interface, which is lacking in NS-2. In addition, OPNET is an expensive commercial product with a very steep learning curve, while NS-2 is free and with much less steep learning curve.

A comparative study of the accuracy of the results of these simulators was performed by Lucio *et al.* [29]. They compared the simulation results with measurements from a live network testbed. They deployed both constant bit rate data traffic and an FTP session. Their results show the necessity of fine-tuning the parameters (TCP window size, enable new reno, max segment size) within a simulator so that it closely reflects the behaviour of a real network. When dealing with stochastic simulation, an important point is the credibility of obtained results. Major studies neglect this aspect and have total belief in their results. However, according to the study of Pawlikowski *et al.* [32], a credible stochastic simulation needs the use of appropriate pseudo-random generators of independent uniformly distributed numbers and appropriate analysis of simulation output data. Other guidelines are formulated to assure a basic level of credibility of simulation studies.

A measurement-based performance evaluation process needs load injectors to play a workload against the system under test. The design and development of such injectors is not a trivial task [33]. The most important test factor supported by a load injector is the injection rate. This test factor is obviously the

base to obtain measures where the throughput metric applies. One must be careful that the desired number of requests per second is really injected into the system under test (e.g. the SNMP agent). The load injector must have enough resources (cpu, memory, network bandwidth) to complete the injection. It is also up to the user to set up a testbed where the network does not introduce a bottleneck between injector(s) and the system under test. To achieve a desired injection rate without the use of too many injectors, we need to code them in a proper way. One way to develop such injectors is to use threads. Each thread has an injection rate of one operation per second. Therefore, the number of instantiated threads is the desired injection rate. The advantage of this way of coding is that there is no need for synchronization between injectors located on separated machines. The main drawback, however, comes from the resolution of the *sleep* method used between each call of an SNMP operation within an injection thread.

A load injection process is split into three phases, as explained in Cecchet *et al.* [34]:

1. A ramp-up or warm-up period: this phase allows one to start injection until it reaches a steady-state rate (desired injection rate).
2. A measurement phase, where the injection rate is maintained at its target value and where all measurements are taken.
3. A ramp-down phase, where the injection rate is still maintained at its target value without further measurement logging. The idea is to avoid the measurement of unachieved requests due to system under test shutdown.

Therefore we have to start one injection thread for each ramp-up duration divided by the injection rate in seconds. Figure 3 depicts the effect of ramp-up on the injection rate during a test. The duration of the ramp-up should be long enough to allow thread creation and initialization, and stabilization of adaptive resource allocation algorithms in response to system load or network traffic.

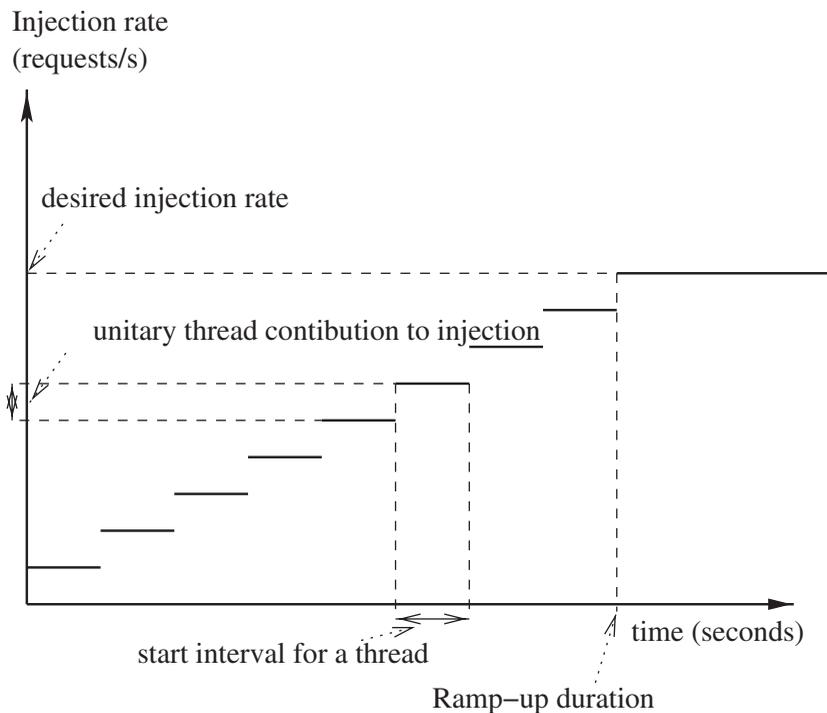


Figure 3. Ramp-up effect on injection rate

7.2 Tool usage in published papers

As stated in Section 3, only few publications use simulations to assess the performance of SNMP. The OPNET simulator is used in Pattinson [15], while the Network Simulator (NS-2) is used in Rubinstein *et al.* [18].

Although an SNMP application is a networking application, its model within the simulator needs special care. A management application has its proper semantics, which differ from classical networking models. In particular, the regular polling traffic to a large number of agents is different from many traditional server-centric traffic models. It is thus not obvious whether SNMP agent should be considered to act like a server or as a client. The same holds for the manager. Simulation models also need to specify the distribution of the SNMP parameters and protocol operations. The choices made by the authors have to be motivated and documented in any performance study.

Lahmadi *et al.* [35] have developed a workload injector tool to assess the performance of JMX-based monitoring applications. The output measurement results show that the monitoring delays fit well within a Weibull distribution in the case of JMX-based management. It is unclear whether such a model is also valid for SNMP delays. This is another example where questions in the context of simulating management approaches like SNMP need answers.

In measurement-based papers, authors typically use their own load injectors. These injectors are often partially developed only in the scope of the papers. Herein, we outline the lack of common and standard measurement load injectors dedicated to management approaches. In other research fields (e.g. web servers, data bases), standard load injectors are available to measure and compare the performance of applications.

8. CONCLUSIONS

SNMP was specified some 20 years ago and is still widely deployed. Despite this success, SNMP performance has never been fully and properly studied in the literature. This lack can be explained by the fact that there is no well-defined and commonly agreed upon measurement methodology to assess its performance. In addition, little is known about SNMP usage patterns in operational networks, which makes it hard to come up with realistic scenarios for the analysis.

In this paper, we have surveyed the main papers that deal with the measurement and analysis of SNMP performance. We find that most papers try to compare SNMP with other management protocols. The comparison is usually partial and does not cover all protocol parameters. Thus the published results of existing studies cannot be generalized. However, this is a fundamental limitation of experimental approaches. Despite this, experimental research remains a good starting point in generating knowledge and validating proposed management approaches. Another popular reason to assess the performance of SNMP is to motivate optimizations of management algorithms (e.g. new protocol operations, table retrieval algorithm optimizations).

Even after many years of SNMP usage, an in-depth analysis of SNMP performance with complete coverage of its features and parameters (security, bulk data retrieval, mixing polling and notifications, etc.) is missing. A first step towards such an analysis is to collect and analyse SNMP traces to identify the most used patterns in real networks [55]. Once suitable patterns are identified, they can be used to identify common management practices that can be evaluated and compared to other management technologies. This first step has been successfully started recently. In a second phase, it will be necessary to redo some of the published studies with a more appropriate parameter set and tool set. This represents a huge effort, which can be undertaken only at the community level.

ACKNOWLEDGEMENT

The work reported in this paper was supported in part by the EC IST-EMANICS Network of Excellence (#26854).

REFERENCES

Surveyed papers

1. Barford P, Sommers J. Comparing probe- and router-based packet-loss measurement. *IEEE Internet Computing Magazine* 2004; **8**(5): 50–56.
2. Baldi M Picco GP. Evaluating the tradeoffs of mobile code design paradigms in network management applications. In *Proceedings of the 20th International Conference on Software Engineering*, Kyoto, 1998; 146–155.
3. Chen Y-C, Chan I-K. SNMP GetRows: an effective scheme for retrieving management information from MIB tables. *International Journal on Network Management* 2007; **17**(1): 51–67.
4. Chen TM, Liu SS. A model and evaluation of distributed network management approaches. *IEEE Journal on Selected Areas in Communications* 2002; **20**(4): 850–857.
5. Corrente A, Tura L. Security performance analysis of SNMPv3 with respect to SNMPv2c. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, Seoul, 2004; 729–742.
6. de Lima WD, Alves RS, Vianna RL, Almeida MJ, Tarouco LMR, Granville LZ. Evaluating the performance of SNMP and Web Services notifications. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, 2006; 546–556.
7. Fioreze T, Granville LZ, Almeida MJ, Tarouco L. Comparing Web Services with SNMP in a management by delegation environment. In *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005)*, 2005; 601–614.
8. Fuggetta A, Picco GP, Vigna G. Understanding code mobility. *IEEE Transactions on Software Engineering* 1998; **24**(5): 342–361.
9. Luderer GWR, Ku H, Subbiah B, Narayanan A. Network management agents supported by a Java environment. In *Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management*, Vol. 86, San Diego, CA, 1997; 790–790.
10. Marinov V, Schönwälder J. Performance analysis of SNMP over SSH. In *Proceedings of the 17th IFIP/IEEE International Workshop on Distributed Operation and Management (DSOM 2006)*. LNCS no. 4269. Springer: Berlin, 2006; 26–36.
11. Pavlou G, Flegkas P, Gouveris S, Liotta A. On management technologies and the potential of Web Services. *IEEE Communications Magazine* 2004; **42**(7): 58–66.
12. Pras A, Drevers T, van de Meent R, Quartel D. Comparing the performance of SNMP and Web Services-Based management. *IEEE Transactions on Network and Service Management* 2004; **1**(2).
13. Zapf M, Herrmann K, Geihs K. Decentralized SNMP management with mobile agents. In *Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management*, 1999; 623–635.
14. Subramanyan R, Miguel-Alonso J, Fortes JAB. A scalable SNMP-based distributed monitoring system for heterogeneous network computing. In *Proceedings of the ACM/IEEE Conference on Supercomputing*. IEEE Computer Society: Dallas, TX, 2000.
15. Pattinson C. A Study of the behaviour of the Simple Network Management Protocol. In *Proceedings of the 12th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM 2001)*, 2001.
16. Du X, Shayman M, Rozenblit M. Implementation and performances analysis of SNMP on a TLS/TCP base. In *Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management*, 2001; 453–466.
17. Gavalas D, Greenwood D, Ghanbari M, O'Mahony M. Advanced network monitoring applications based on mobile/intelligent agent technology. *Computer Communications* 2000; **23**(8): 720–730.
18. Rubinstein MG, Duarte OCMB, Pujolle G. Reducing the response time in network management by using multiple mobile agents. In *Proceedings of the IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, 2000; 253–265.
19. Wee CM, Beg MS. Performance evaluation of APSSNMP: an alternative security algorithm for SNMP. *Journal of Network and Systems Management* 2002; **10**(4): 411–415.
20. Bivens A, Gupta R, McLean I, Szymanski B, White J. Scalability and performance of an agent-based network management middleware. *International Journal of Network Management* 2004; **14**(2): 131–146.
21. Gu Q, Marshall A. Network management performance analysis and scalability tests: SNMP vs. CORBA. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, 2004.
22. Neisse R, Vianna RL, Granville LZ, Almeida MJB, Tarouco LMR. Implementation and bandwidth consumption evaluation of SNMP to Web Services gateways. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, Seoul, 2004; **9**(1): 715–728.

23. Zhang XT, Zhang L, Zhou J. A high performance strategy for constructing dynamic MIB. In *Fifth International Conference on Information Technology: New Generations*, 2008; 1164–1167.
24. Holt A, Huang C-Y, Monk J. Performance analysis of mobile agents. *Communications, IET* 2007; **1**(3): 532–538.
25. Soldatos J, Alexopoulos D. Web services-based network management: approaches and the WSNET system. *International Journal of Network Management* 2007; **17**(1): 33–50.
26. Alexopoulos D, Soldatos J. XMLNET: an architecture for cost effective network management based on XML technologies. *Journal of Network and Systems Management* 2005; **13**(4): 451–477.
27. Holland HE, van de Meent R, Pras A. Evaluating MIB II (RFC1213) implementations. *Simple Times* 2002; **10**(1): 8–15. <http://www.simple-times.org/pub/simple-times/pdf/vol10-num1.pdf> [30 June 2009].

Complementary papers

28. Jain R. *The Art of Computer Systems Performance Analysis*. Wiley: New York.
29. Lucio GF, Paredes-Farrera M, Jammeh E, Fleury M, Reed MJ. OPNET Modeler and Ns-2: comparing the accuracy of network simulators for packet-level analysis using a network testbed. *WSEAS Transactions on Computers* 2003; **2**(3): 700–707.
30. Fall K, Varadhan K. *NS Notes and Documentation*. Technical report, VINT Project, 1999.
31. OPNET Technologies. The OPNET modeler. http://www.opnet.com/solutions/network_rd/modeler.html [13 July 2009].
32. Pawlikowski K, Jeong H-DJ, Lee J-SR. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine* 2002; January: 132–139.
33. Buble A, Bulej L, Tuma P. CORBA benchmarking: a course with hidden obstacles. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*. IEEE Computer Society: Washington, DC, 2003.
34. Cecchet E, Marguerite J, Zwaenepoel W. Performance and scalability of EJB applications. *SIGPLAN Notices* 2002; **37**(11): 246–261.
35. Lahmadi A, Andrey L, Festor O. On delays in management frameworks: metrics, models and analysis. *17th IFIP/IEEE Distributed Systems: Operations and Management, DSOM 2006*, Dublin. Springer: Berlin, 2006.
36. Loiacono J, Germain A, Smith J. Network performance measurements for NASA's earth observation system. *Computer Networks* 2004; **46**(3): 299–320.
37. Lee J-S, Hsu P-L. Design and implementation of the SNMP agents for remote monitoring and control via UML and Petri nets. *IEEE Transactions on Control Systems Technology* 2004; **12**(2): 293–302.
38. Henniger O, Barbeau M, Sarikaya B. Specification and testing of the behavior of network management agents using SDL-92. *IEEE/ACM Transactions on Networking* 1996; **4**(6): 951–962.
39. Gani-Naor I, Margolin E, Rafaeli R. GeNMSim: the agent simulator. In *Fifth Annual Tcl/Tk Workshop*, Boston, MA, USENIX, 1997.
40. Mai J, Chuah C-N, Sridharan A, Ye T, Zang H. Is sampled data sufficient for anomaly detection? In *Proceedings of the 6th ACM SIGCOMM on Internet Measurement (IMC 06)*, Rio de Janeiro. ACM Press: New York, 2006; 165–176.
41. Zhao Q, Ge Z, Wang J, Xu J. Robust traffic matrix estimation with imperfect information: making use of multiple data sources. In *SIGMETRICS '06/Performance '06: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, St Malo. ACM Press: New York 2006; 133–144.
42. Zhao Y, Chen Y, Bindel D. Towards unbiased end-to-end network diagnosis. In *SIGCOMM '06: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pisa. ACM Press: New York, 2006; 219–230.
43. Case JD, Fedor M, Lee Schoffstall ML, Davin JR. Simple Network Management Protocol (SNMP). *RFC 1157*, 1990.
44. Subramanyan R. *Scalable SNMP-based monitoring systems for network computing*. PhD dissertation, Purdue University, 2002.
45. Kwak BJ, Song NO, Miller LE. A mobility measure for mobile ad-hoc networks. *IEEE Communications Letters* 2003; **7**: 379–381.
46. Holland HE. *Evaluating MIB-II implementations: a comparison of management functionality within routers*. Technical report, Centre for Telematics and Information Technology, University of Twente, 2002.
47. Intel. *Using the RDTSc Instruction for Performance Monitoring*. Intel Corp., 1997.
48. Schönwälder J. Characterization of SNMP MIB modules. In *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005)*, Nice, 2005; 615–628.

49. Paxson V, Almes G, Mahdavi J, Mathis M. Framework for IP performance metrics. *RFC 2330*, 1998.
50. Kantorovitch J, Mahonen P. Case studies and experiments of SNMP in wireless networks. In *IPOM 2002: IEEE Workshop on IP Operations and Management*, Dallas, TX, 2002; 179–183.
51. Schönwälder J, Pras A, Matus H, Schippers J, van de Meent R. SNMP traffic analysis: approaches, tools, and first results. In *Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007)*, Munich, 2007; 323–332.
52. Malowidzki M. GetBulk Worth Fixing. *Simple Times* 2002; **10**(1): 3–6.
53. Sprenkels R, Martin-Flatin J-P. Bulk transfers of MIB data. *Simple Times* 1999; **7**(1): 1–7.
54. van den Broek JG, Schönwälder J, Pras A, Harvan M. SNMP trace analysis definitions. In *Proceedings of the 2nd International Conference on Autonomous Infrastructure, Management and Security (AIMS 2008)*, Bremen. LNCS 5127, Springer: Berlin, 2008.
55. Schönwälder J. SNMP traffic measurements and trace exchange formats, Jacobs University Bremen. *RFC 5345*, 2008.

AUTHORS' BIOGRAPHIES

Laurent Andrey (Laurent.Andrey@loria.fr) is an associate professor in the Department of Computer Science at Nancy 2 University and member of the MADYNES research team. He received a Ph.D. in Computer Science from Nancy University in 1991. His research interests are in network and service management architectures, performance evaluation and networked applications security assessment models and techniques.

Olivier Festor (Olivier.Festor@inria.fr) is a research director at INRIA Nancy—Grand Est where he leads the MADYNES research team. He has a Ph.D. degree (1994) and an Habilitation degree (2001) from Henri-Poincaré University, Nancy, France. He spent 3 years at the IBM European Networking Center in Heidelberg, Germany and one year at the EURECOM Institute in Nice, France. His research interests are in the design of algorithms and models for automated security management of large scale networks and services. This includes monitoring, fuzzing and vulnerability assessment. Application domains are IPv6, Voice over IP services and dynamic ad-hoc networks.

He has published more than 70 papers in network and service management and serves in the technical program and organization committees as well as in the editorial boards of several international conferences and journals. He was the TPC Co-chair of the IFIP/IEEE IM'2005 event. Since 2006, he leading the EMANICS European Network of Excellence dedicated to Management Solutions for the Future Internet and was named co-chair of the IFIP TC6 Working Group 6.6 co-chair in 2008.

Abdelkader Lahmadi (Abdelkader.Lahmadi@loria.fr) is a research engineer at Nancy Université, France and a member of the MADYNES research team. He obtained a Ph.D degree in computer science in 2007 from Henri Poincaré University—Nancy for his thesis entitled «Performance of network and service monitoring frameworks». His research interests include management benchmarking, IP multicast where he developed the reference implementation of the IGMP proxy, IPv6 and network security. Currently, he is working on algorithms and models for automated security management for SIP-based voice over IP networks.

Aiko Pras (a.pras@utwente.nl) is an associate professor in the Department of Electrical Engineering and Computer Science at the University of Twente, the Netherlands, and a member of the Design and Analysis of Communication Systems Group. He received a Ph.D. degree from the same university for his thesis titled “Network Management Architectures”. His research interests include network management technologies, network monitoring and measurements, and web services. He has participated in many European and Dutch research projects, and is currently a research leader in the European Network of Excellence on Management of the Internet and Complex Services (EMANICS). He is chairing the IFIP Working Group 6.6 on “Management of Networks and Distributed Systems”, is editor of the IEEE Communications Magazine series on “Network & Service Management”, associate editor of the International Journal of Network Management (IJNM), and Editorial Advisory Board member for the Journal of Network and Systems Management (JNSM). He is Steering Committee member of the IFIP/IEEE NOMS and IM Symposia (NISC), Management Week (Manweek), AIMS, E2EMon, as well as the EUNICE Consortium. He was/is (Technical Program) Co-Chair of several conferences, including IM'05, EUNICE'07, TMA'09, AIMS'09 and Manweek'09.

Jürgen Schönwälder (j.schoenwaelder@jacobs-university.de) is an associate professor of computer science at Jacobs University Bremen, leading the research group on computer networks and distributed systems. He received his diploma in computer science in 1990 and his doctoral degree in 1996 from the Technical University of Braunschweig, Germany. His research interests are network management, distributed systems, wireless sensor networks, and network security. He is an active member of the Internet Engineering Task Force (IETF) where he has edited more than 20 specifications and standards, and chair of the Network Management Research Group (NMRG) of the Internet Research Task Force (IRTF).