

Design and Implementation of an ANTS-based test bed for Collecting Data in Active Framework

Victória Damasceno Matos¹, Jorge Luiz de Castro e Silva², Javam C. Machado¹,
Rossana de Castro Andrade¹, José Neuman de Souza¹

¹Universidade Federal do Ceará (UFC)
{victoria, rossana}@lia.ufc.br, {javam, neuman}@ufc.br
²Universidade Federal de Pernambuco (UFPE)
jlcs@cin.ufpe.br

Abstract. The active networks approach has been presented as an alternative technology for solving several problems in conventional networks, mainly in the network management area. This work describes a data retrieval mechanism test bed, including design and implementation, which can be used for managing SNMP enabled networks, based on the ANTS (Active Node Transfer System) toolkit framework developed at the Massachusetts Institute of Technology (MIT). The results obtained from the experiments have shown that the chosen approach is a good alternative for data retrieving operations.

1 Introduction

The motivation for the proposed model came from the fact that traditional management approaches like SNMP (Simple Network Management Protocol) and CMIP (Common Management Information Protocol) generate excessive network traffic, as they have mechanisms based on the client-server model, where the manager centralizes the information and provides scheduling for execution of corrective actions, while the agent interacts with the MIB (Management Information Base) and executes the requests issued by the manager.

Another important concern is the network management applications that may benefit from the use of the active network paradigm. Regarding active network studies [1, 2, 3, 4, 5], it has been verified that one of the biggest platforms embedding this technology is the ANTS (Active Node Transfer System).

In the proposed model, some procedures were implemented within the ANTS platform, enabling data acquisition from an initialization process within an active node. In the process, an active node sends capsules (corresponding to traditional network packets) that execute locally the collection task.

With this model, which provides dynamic network traffic parameters, it is possible to assist performance management methods, mainly those that use active technologies.

2 Active Networks

Traditional networks guarantee data sharing in the sense that packets can be efficiently carried among connected systems. They only make the processing needed to forward the packets to the destination. This kind of network is insensitive to the packets it carries. The packets are transferred without modification [6]. Computation role in these kinds of networks is extremely limited.

Active networks are a new approach to network architecture that allow their users to inject customized programs into the nodes, enabling these devices to perform customized processing on the messages flowing through them [1, 6]. This approach can be applied both to users and applications.

The results obtained from the active networks in network management applications are of great interest to this work.

2.1. Active Node Transfer System (ANTS)

Developed by the Massachusetts Institute of Technology (MIT), ANTS (Active Node Transfer System) is a toolkit, written in Java where new protocols are automatically deployed at both intermediate nodes and end systems by using mobile code techniques [2].

ANTS views the network as a distributed programming system and provides a programming language-like model for expressing new protocols in terms of operations at nodes. The tool combines the flexibility of a programming language with the convenience of dynamic deployment.

In the ANTS architecture the capsules which refer to the processing to be performed on their behalf replace the packets in traditional networks include a reference to the forwarding routines to be used for processing at each active node. There are forwarding routines that will be found in all the nodes. Others are specific to the application and will not reside at every node, but must be transferred to a node by means of code distribution before the capsules of that type can be processed for the first time. The code group is a collection of related capsule types whose forwarding routines are transferred as a unit by the code distribution system. Finally, the protocol is a collection of related code groups that are treated as a single unit of protection by an active node. The active nodes have the responsibility to execute protocols within restricted environments that limits their access to shared resources.

3 Proposed Retrieval Mechanism

The module, which was designed and implemented, is composed of a few scripts that are responsible for the configuration of the notes to be monitored. The module also contains forwarding application capsules and Java classes that are responsible for structuring the capsules and the data collection process.

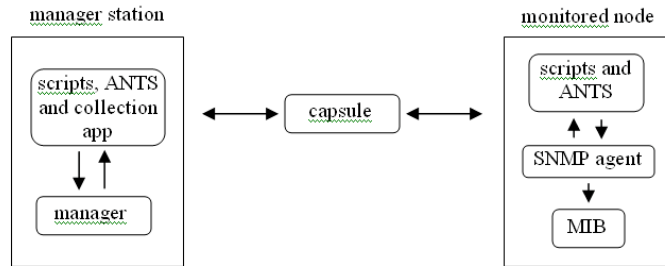


Fig. 1. Data retrieval process

In other words there is an interaction between the application and the SNMP agents in order to get MIB information from the analyzed nodes. Figure 1 shows how the mechanism works. The manager station sends capsules with routines that must be executed in the monitored node, thereby obtaining the desired information.

The first step towards the implementation was to establish ANTS active environment where an active application is due to execute the routines desired in a node. To activate a node it is necessary to associate a virtual IP and an entry to a determined active application in the node, where such configuration is passed as a parameter through a script (*coleta.config*). Thus, when the capsule of a certain application arrives at an active node, it executes there the desired routines.

Another important factor is the routes to be followed by the capsules. The forwarding must be pre-established in one script called *coleta.routes*, which can be generated by the ANTS.

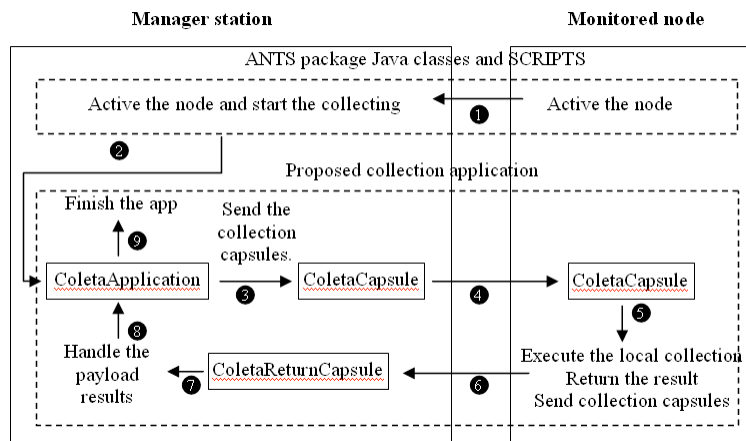


Fig. 2. Execution model of proposed retrieval mechanism.

After that (scripts and ANTS package), the application is ready to work. It is composed essentially of three distinct classes: *ColetaApplication*, *ColetaCapsule* and *ColetaReturnCapsule*.

The *ColetaApplication* class starts the data capture, manages the sending and receiving of capsules and finishes all the process in the manager node. The

ColetaCapsule creates the capsules that embed the local collection code that will be sent to the monitored nodes. This way, it is possible to send new collection capsules to the monitored nodes, which implies a distributed model. Finally, the *ColetaReturnCapsule* creates the capsules that will return the results and treats the values returned to the manager node. Figure 2 shows the steps of this execution model.

3.2. Advantages of the Active Retrieval Mechanism

An analysis was made that compares the proposed mechanism with the native model of the management protocol using SNMP (Figure 3) in order to show the performance of both approaches.

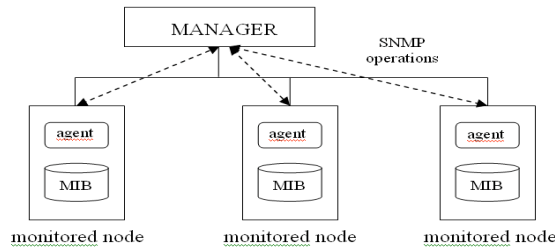


Fig. 3. Traditional network framework with SNMP protocol.

In this environment, which was based on the client-server model, there is a great number of messages exchange. The manager, thus, is interacting with all the managed devices to get the management information from the MIB agents. This can result in decreasing performance, as the management centralization will overload the manager and the tremendous number of messages exchanges between the manager and the agent will eventually increase the network traffic.

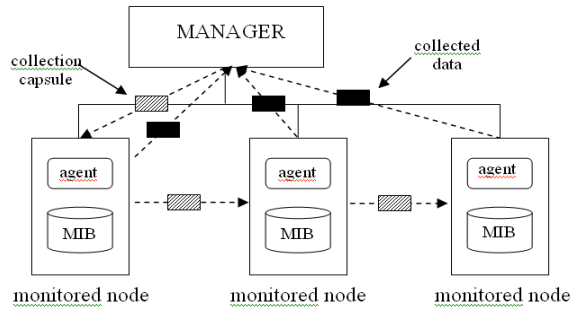


Fig. 4. Distributed framework of data collection

As previously seen, the traditional form of getting MIB data is centralized, thereby forcing the manager to capture information from all the devices that it manages. In the proposed method, any active node, not necessarily the manager, can send a capsule to

run the collection in another active node and one single capsule can run the collection operation in some devices, as represented in Figure 4.

When the process is started, only one capsule that is sent to the agents can acquire information from all monitored node. This is possible because an agent is able to send collection capsules to other agents. The obtained results will be returned to the management station through capsules too.

4 Results

A comparison was made between the time spent by the proposed collection model and the traditional model of SNMP, using the ucd-SNMP tool [7]. It was discovered that, by employing the proposed mechanism, the capture of the entailed values of some OIDs, in only one node, results in a longer time. But when some nodes were added to the collection process and the distribution mechanism was used, the performance was significantly improved, as explained in Figure 4. This conclusion was arrived to by looking at the average of the obtained results that are graphically presented in Figure 5. In one experiment, two nodes (the manager and the agent) were used. In another experiment, three nodes (one manager and two agents) were employed. The benchmarks were made with and without distributed forwarding and the varied collected OIDs numbers.

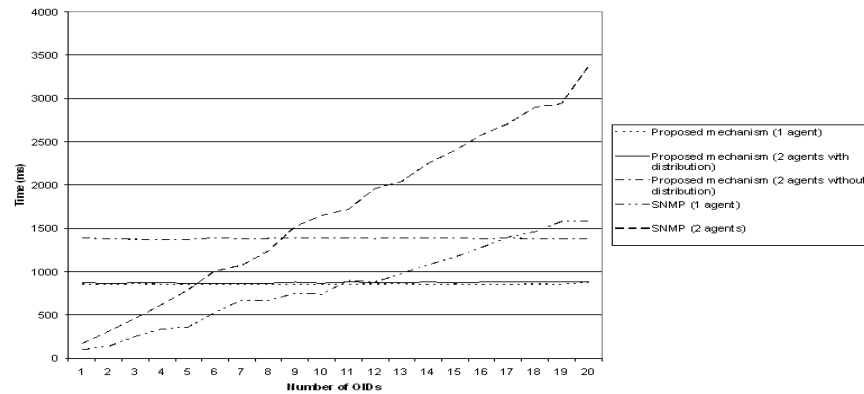


Fig. 5. Benchmarks.

A more significant improvement has been found when the number of OIDs to be consulted was modified. Using the experiment with three machines, one can notice that the increase on the number of OIDs to be obtained has more impact on the elapsed time when the environment is the traditional SNMP. This is explained by the fact that, in the proposed method, the manager sends only one capsule with all the OIDs to be collected; while in the traditional SNMP, some PDUs (Protocol Data Unit) are sent to get data.

In the practical study, it was verified that the configuration of the routes is also an excellent factor to obtain better times. This way, in an environment with many machines, it is not enough to have a distributed management without evaluating that

points would be advantageous to make the decentralization. It was noted that in a monitoring of few machines, the collection system must be totally distributed, or either, a monitored node to pass the collection capsule one another node to be monitored. However, as the number of nodes increases, there can be a delay in the collection with this decentralization model. Therefore, it would be of great value the study and the implementation of an optimal solution in the establishment of the routes.

5 Conclusions

The motivation of this work is the deficiencies presented in the traditional management protocols, based on the client-server model. They are centralized management models that may generate a great number of exchanges of messages between an agent and a manager, wasting bandwidth and resulting in delays in the network monitoring.

In the analysis that were performed, it was noted that one of the most important active platforms is the ANTS, a toolkit written in Java developed by the Massachusetts Institute of Technology (MIT) that allows a dynamic construction and implementation of network protocols. For this reason, this framework was used in the development of the proposed module.

The aim of optimizing the use of the broadband and diminishing the time spent on collecting data has been reached, because the proposed model is a data collection mechanism based on distributed management and by the fact that only the capsules that will indicate the start of the local collection processing are sent through the network. This is achieved through local SNMP operations, preventing the excess of messages exchange between the client and the server. Benchmarks were conducted that prove the benefits on active data collection claimed in this work.

6 References

1. TENNENHOUSE, D. L. A Survey of Active Network Research. *IEEE Communications Magazine*, v.35, n.1, p.80-86, Jan., 1997.
2. WETHERALL, D.J.; GUTTAG, J.V.; TENNENHOUSE, D.L. ANTS: "A Toolkit for Building and Dynamically Deploying Network Protocols. In: IEEE OPENARCH'98, San Francisco, CA, Apr. 1998.
3. SCHWARTZ, B.; ZHOU, W.; JACKSON, A.W. Smart Packets for Active Networks. *BBN Technologies*, Jan. 1998.
4. WETHERALL, D.J.; TENNENHOUSE, D.L. The ACTIVE_IP option. In: THE 7TH ACM SIGOPS EUROPEAN WORKSHOP, 7, Sep., 1996.
5. GUNTER, C.A.; NETTLES, S.M.; SMITH, J.M. The Switch Ware Active Network Architecture. *IEEE Network*, special issue on Active and Programmable Networks, v.12, n.3, , May/Jun, 1998.
6. TENNENHOUSE, D. L.; WETHERALL D.J. Towards an Active Network Architecture. *Computer Communication Review*, v.26, n.2, p.5-18, Apr., 1996.
7. Net-SNMP website. Available in: <<http://www.net-snmp.org>>. Access in: 20 out. 2002.