Automated and Distributed Network Service Monitoring

Giovan Germoglio, Bruno Dias, Pedro Sousa

Centro de Ciências e Tecnologias da Computação, Departamento de Informática,
Universidade do Minho, Campus de Gualtar, 4710-057 Braga, Portugal
{gicage, bruno.dias, pns}@di.uminho.pt

Abstract. Decentralized architectures have been proposed using functional hierarchies of several middle-level managers with delegation of management activities. However, no management applications have been implemented and widely used that could sanction this approach for real use. Also, current frameworks or mechanisms in network management do not directly address automation. More efficient network management systems with higher levels of availability and automation are needed. In this paper we propose a design and specify an automated and distributed network services monitoring system for Internet services that should be capable to effectively gather and calculate relevant operational parameters that will be used to determine the availability level of a network or application service. Based on this availability level, the system recommends or automatically triggers a service reconfiguration or a total or partial network service replication.

Keywords: Network Services Management, Automation, Distributed Monitoring.

1 Introduction

The last twenty years have been prolific on standards, protocols, technologies and assorted mechanisms on the computer communications field, resulting from an intensive and increasing research activity and driven by user demands for better devices, services and applications. The same applies for the specific area of network management.

In general, the technologies used to implement management systems are based on an overly centralized agent-manager architecture, which puts too much effort on the manager, or application, side. This classic client-server paradigm uses manager's invocations of remote procedures that implement a management activity or function. Within these frameworks, management functions should be classified using the old and classic Open Systems Interconnection Network Management Framework (OSI/NMF) [13] taxonomy: fault, configuration, accounting, performance and security management activities. More decentralized architectures have been proposed using functional hierarchies of several middle-level managers or delegation of

management activities (or management code) to the agents (or management servers), but no real management applications have been implemented and widely used. Furthermore, automation is still an open issue in network management, which is not efficiently addressed by current frameworks or mechanisms. The only form of automation pursued by standards are the policy management approaches defined by the Internet Engineering Task Force's (IETF) Policy Framework working group, that has defined the Policy Core Information Model [14] extensions to the Distributed Management Task Force (DMTF) Common Information Model (CIM) [15] specification for exclusive usage on policies definition, and the IETF's work on policy provisioning specified as the Common Open Policy Service Protocol (COPS) [16]. Again, these approaches had, until now, no significant impact on available network management products in respect to automation or distribution.

Internet Services availability is of great importance for many user-distributed applications and is completely dependent on the correct operation and availability of all major standard network services. Our overall purpose is to design and specify an automated and distributed network services monitoring and configuration system for Internet services focused on the calculation of availability levels. In this article we discuss the architecture of the monitoring sub-system that should be able to effectively calculate the availability level of a network or application service and instruct the configuration sub-system on further actions (backup, replication, etc).

2 Related Work

Several research and development projects already exist for Internet services monitoring, although the most relevant were deployed for DNS monitoring, either using active or passive techniques.

The RIPE NCC DNS Monitoring Service project (DNSMON) [23] uses a direct approach and tries to actively test the availability of the DNS system using a globally distributed active measurement service. This is, probably, the only network services project deployed on a global scale and uses injected DNS test traffic to measure QoS information. On the other hand, the dnslogger [24] is a Weimer's DNS passive logging software. In this approach, sensors capture DNS packets on the network and forward them to an analyzer module integrated on its architecture. This solution is bounded to a specific network service and does not use a real-time monitoring mechanism, although non real-time availability studies could be performed from the logged data.

The mobile agent platform for Customer-based IP Service Monitoring (CSM) that exploits the unique capabilities of mobile agents is implemented by Günter et. al [6]. Agents are deployed by ISPs and by customers to perform measurements tailored to individual monitoring needs and use filter objects to describe protocol packets in to be analysed. This is a more generic concept since it could permit, depending on the specification of the filter objects, monitoring of several network services.

The MonALISA project [25] defines a distributed architecture for service monitoring and is based on a scalable dynamic and distributed architecture. This model used a higher-level component to decide which monitoring mechanisms and

tools to use depending on the overall monitoring goal, which is also one of the requisites of our monitoring system.

Choi and Hwang [26] focused their approach on in-service end-to-end flow monitoring by utilizing packet flows to gather information about QoS parameters. Test packets are generated periodically and circulate on an application flow so to allow collection of important statistical information such as network throughput, packet loss, delay and jitter. References [1] to [3] also provide mechanisms for calculation of network or end-to-end application performance parameters.

It is clear that the monitoring concepts behind these projects are not universal and they are overly dependent on the specific needs of the monitoring process of the specific network service availability they intent to monitor. These tools concentrate their efforts on monitoring some direct or indirect connectivity or functional parameters of a specific network service.

3 Automated Network Services Management

In the context of our project, we are interested in define a generic mechanism for measuring and calculate availability levels for any network service. Furthermore, the monitoring sub-system should integrate an automated decision process and the capability to interact with a configuration sub-system, as depicted on Figure 1.

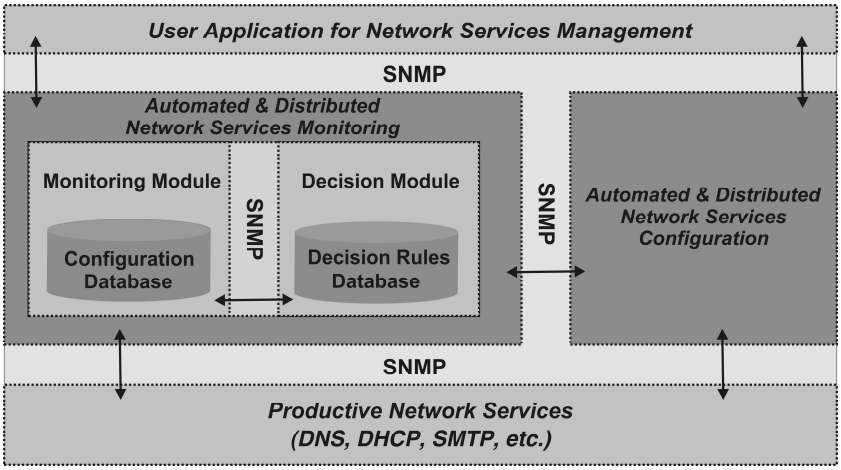

Fig. 1. The Automated Network Management Framework.

The internal design of the monitoring sub-system comprises two types of modules: the Monitoring and the Decision Modules. The deployment of these modules in separate network devices/hosts enhances the resilience capabilities of this sub-system. On a typical management domain there will be a much larger number of monitoring modules than decision modules.

3.1 Decision Module

An Active Decision (AD) module , which could be implemented like a Lightweight Policy Management Server (LPMS), must derive decisions from a pre-defined set of strategic, administrative and operational management policy rules when applied to a set of network services QoS metric values that should be continuously evaluated by the monitoring modules. When this metrics imply reduced availability levels an Automated and Distributed Network Services Configuration sub-system (ADNSC), such as depicted on Figure 1, should be automatically notified for an immediate network services reconfiguration procedure. The ADNSC sub-system will be responsible for the reconfiguration process. In the present project we are only interested in implementing major reconfiguration mechanisms like an entire configuration backup, configuration reposition or service replication. In this respect, the decision modules may need complex artificial intelligence algorithms for evaluation of availability metrics but should also use simpler estimation techniques like the ones referred on [6].

Since the two types of modules will be deployed separately, the decision modules can be implemented as part of a global policy management network already deployed on the network services management domain. Also, the high-level language for definition of the policy management rules could be the same as the language used for configuration of monitoring modules.

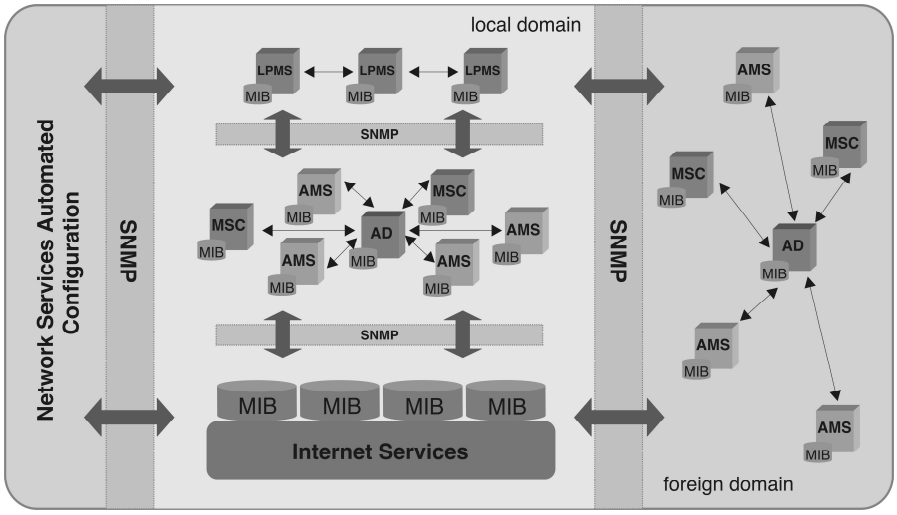

Fig. 2. Distributed Monitoring Architecture

3.2 Monitoring Module

The monitoring process is distributed through a set of monitoring servers, which are named Active Monitoring Servers (AMS) or Monitoring Server Candidates (MSC),

according to whether the monitoring state is active or in standby mode respectively. Both types of monitoring modules should be deployed on several locations inside the monitoring domain and, if possible, on some strategic locations outside the monitoring domain, as showed on Figure 2. This distribution is mandatory since the reliability of the continuous measurement of the QoS metrics for a network service depends on the number and location of the AMSs. The group of MSCs is continuously and dynamically maintained through manual or automatic manipulation of configuration parameters, that is, the methodology to determine which servers become active for monitoring of a specific network service could be human based or automated through a decision module server. In the last case, two possible mechanisms, which complement each other, will be described in Section 4.

The monitoring sub-system should support the use of independent databases technologies for storage of the monitoring configuration parameters. It could also support the use of database systems external to the monitoring framework. This added flexibility could add resilience and speed up implementation and deployment times, since already existing distributed or local database systems could be used.

3.3 Automated and Distributed Network Services Configuration

The framework also should support a network services configuration sub-system. The aim of this part of the management system is to support network services configuration, re-configuration, backup or replication procedures taking into account the notified decisions made by AD modules on the monitoring sub-system or as a result of higher level user management applications. The internal design and specification of this configuration sub-system is out of the scope of this text.

3.4 Module's Interaction and Management Information Model

The definition of such management system requires the specification of management communication protocols and information models for management data representation. This includes management data structures or objects, their encoding rules and the protocol operations that must be implemented to ensure a correct and efficient interaction between all modules in the system and between these modules and external lower level monitoring probes or higher level user management applications. Shorter development/implementation periods and universal support should be obtained if well known and widely deployed management frameworks and protocols are used.

The management framework proposed in [8], which adopts the eXtensible Markup Language (XML) as an option for representation of management objects in network management, is composed by XML-based Manager, a XML/SNMP gateway for legacy SNMP (Simple Network Management Protocol) agents and a XML-based agent. The author indicates some useful features of XML for the transfer of a large amount of management data using HTTP, claming similar efficiency when compared to pure SNMP frameworks and considering resource utilization and processing time. Nevertheless, this information model and communication paradigm presents some

drawbacks as reviewed on [7], mainly scalability problems introduced by the XML/SNMP gateway. The monitoring performance of SNMP-based solutions was also compared to Web Services-based frameworks by Aiko Prass *et al.* [9] and for many relevant performance aspects (like bandwidth usage, round trip delays, CPU time and memory requirements) the SNMP framework was considered a much better solution. It becomes obvious that the use of a protocol and an information model that was not originally specified for network management systems should be avoided as much as possible.

At this moment we have chosen the SNMP framework as the supporting deployment technology for the lower level management data representation and transfer between component modules, as depicted on Figure 1 and 2. The Internet Network Management Framework is the most dominant and widely available framework for network management.

For representation of higher level configuration parameters and monitoring metrics we can use the Structure of Management Information (SMI) [10] language, creating special configuration and decision rules databases. These can be implemented as special Management Information Bases (MIBs) [11]. Alternatively, this higher level management data could be represented by XML Schemas, which provide better features to represent data with a higher level of functionality. The tests realized in [12] showed that the XML encoding process can be faster than the ASN.1 Basic Encoding Rules (BER) encoding process if small data structures are used, although, BER decoding is always faster than XML decoding, independently of the data size.

4 Distributed Monitoring Architecture

Some existing distributed monitoring frameworks use the concept mobile agent as a means to distribute the monitoring process. Two types of agent mobility are characterize in Fuggetta *et al.* [4]: strong mobility, which has the ability to migrate both the code and the execution state of one network element; and weak mobility, which allows only code transfer across different network elements. The performance of both approaches is assessed in [5] and weak mobility presents a favorable performance factor, capable of minimizing the detection time for changes on the network state. But in our model, even the weak mobility technique could be problematic since we are after very short network services reconfiguration or replication periods. We use a mechanism that can be seen equivalent to a weak mobility technique but is much more efficient in terms of system response in case of relocation of a monitoring module, that is, in reality, implemented as a simple change in the sate of a standby module to an active module (and the replaced active module simple change of its state to inactive or to standby).

As depicted on Figure 2, we could consider two layers on our monitoring process: an higher layer composed by the monitoring servers and a lower layer composed by all the monitoring probes (the ones already implemented by standard monitoring agents on the network). The pseudo mobility mechanism, implemented through AMSs and MSCs, is only supported and explicitly effective on the higher layer. Its algorithm must guarantee that the network services availability metrics evaluated by

the monitoring process are not disguised by an incorrect operation of one or a group of active monitoring modules.

The correct network location of the monitoring servers is of extreme importance when asserting the validity of its evaluated monitoring metrics. This requisite implies a complex algorithm to calculate the monitoring servers' location. Several existing techniques could help and some were studied. Jacobson's traceroute [17] was the first network tool that is widely used to determine the route taken by packets to diagnose network problems by building network maps. The inaccuracy of traceroute can occur when the path measurements go through a load balance router. They can distribute their traffic across multiple paths by using routing policies and circle, loops and diamonds [18] are common anomalies found when calculating the path from one source to multiple destinations. Paris traceroute [18] is an attempt to avoid the load balance problem. The reference works on [19] and [20] present an extension to Paris traceroute with the capability to construct a multipath map between a source and a destination by implementing an adaptive stochastic probing algorithm. Several probe methods were compared in different scenarios by Luckie *et al.* [21 and the method used by ICMP-Paris demonstrate the best performance. ICMP-Paris and UDP were also were compared with CAIDA's Archipelago (Ark) [22] dataset, which is the evolution of the Skitter infrastructure, where data is continuously sent by active probes to randomly select network destinations

Taking into account the more accurate measurements of Paris traceroute and the concepts behind projects like Skitter and Archipelago, a methodology for distribution of the monitoring servers will be derived. This mechanism should be implemented on the AD modules and should calculate an efficient and resilient monitoring topology of AMSs and MSCs.

6 Conclusion

In this paper, we presented a distributed and automated management system with focus on the monitoring sub-system. The framework should be capable of monitoring all Internet services and make automated decisions based on the evaluated high level availability metrics. The system specification was based on other state-of-the-art network-services monitoring projects or technologies, that, in some cases, represent standard, well known and widely used mechanisms on the Internet, and, in other cases, represent new state-of-the-art approaches to network services monitoring. This article presents an ongoing research and development work that will enter the experimentation in the next months, so, we expect that experimental results could be presented in the near future.

References

1. Matthews, W., Cottrell, L.: The PingER project: active Internet performance monitoring for the HENP community. IEEE Communications Magazine, 38, pp. 130--136, (2000)

2. Fu, Y., Cherkasova, L., Tang, W., Vahdat, A.: EtE: Passive End-to-End Internet Service Performance Monitoring. In: USENIX Annual Technical Conference, (2002)
3. Chen, T.M., Hu, L.: Internet performance monitoring. In: IEEE, pp. 1592--1603, (2002)
4. Fuggetta, A., Picco, G.P., Vigna, G.: Understanding code mobility. In: IEEE Trans. Soft. Engineering, 24, pp. 342--361, (1998)
5. Chen, T.M., Liu, S.S.: A model and evaluation of distributed network management approaches. IEEE J. on Selec. Areas in Comm. 20, 850--857 (2002)
6. Gunter, M. Braun, T.: Internet service monitoring with mobile agents. IEEE Network. 16, 22--29, (2002)
7. Choi, M., Jong J.: Performance Evaluation of XML-based Network Management. In: 16th IRTF-NMRG meeting, (2004)
8. Choi, M., Hong, J., Ju, H.: JXML-Based Network Management for IP Networks. ETRI J. 25, 445--463 (2003)
9. Pras, A., Drevers, T., van de Meent, R., Quartel, D.: Comparing the Performance of SNMP and Web Services based Management. In: IEEE electronic Transactions on Network and Service Management, 1, (2004)
10. Rose, M., McCloghrie, K.: Structure and Identification of Management Information for TCP/IP-based Internets. RFC 1155, May (1990)
11. McCloghrie, K., Rose, M.: Management Information Base for Network Management of TCP/IP-based internets. RFC 1156, May (1990)
12. Chadwick, D. Mundy, D.: Comparing the Performance of Abstract Syntax Notation One (ASN.1) vs eXtensible Markup Language (XML). In: TERENA Networking Conference, (2003)
13. ITU-T Rec. X.701: Information Technology – Open Systems Interconnection. Systems Management Overview, (1992)
14. Moore, B., Ellesson, E., Strassner, J., Westerinen, A.: Policy Core Information Model. RFC 3060, February (2001)
15. Distributed Management Task Force (DMTF): Common Information Model (CIM) Standards, http://www.dmtf.org/standards/cim/
16. Durham, D., Ed., Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A: The Common Open Policy Service Protocol. RFC 2748, January (2000)
17. Jacobson, V.: traceroute. (1989)
18. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C. Teixeira, R.: Avoiding traceroute anomalies with Paris traceroute. In: ACM SIGCOMM Internet Measurement Conference, pp. 153--158, (2006)
19. Augustin, B., Friedman, T., Teixeira, R.: Multipath tracing with Paris traceroute. In: Workshop on End-to-End Monitoring Techniques and Services, pp. 1--8, (2007)
20. Augustin, B., Friedman, T., Teixeira, R.: Measuring load-balanced paths in the Internet. In: 7th ACM SIGCOMM conference on Internet measurement, pp. 149--160, (2007)
21. Luckie, M., Hyun, Y., Huffaker, B.: Traceroute probe method and forward IP path inference. In: 8th ACM SIGCOMM conference on Internet measurement, pp. 311--324, (2008)
22. Archipelago measurement infrastructure, http://www.caida.org/projects/ark/
23. RIPE NCC DNS Monitoring Services, http://dnsmon.ripe.net/dns-servmon/about.html
24. Weimer, F.: Passive DNS replication. In: FIRST Conference on Computer Security Incident Handling, (2005)
25. Newman, H.B., Legrand, I.C., Galvez, P., Cirstoiu, C.: MonALISA: A distribute monitoring service architecture. In: 2003 Conference for Computing in High-Enery and Nuclear Physics, (2003)
26. Choi, Y., Hwang, I.: In-service QoS monitoring of real-time applications using SM MIB. Int. J. Net. Man. 15, 31-- 42 (2005)