Automatic SLA Management in SLA-aware architecture

M. D'Arienzo, M.Esposito*, S.P. Romano, G. Ventre

Università degli Studi "Federico II" di Napoli
DIS, Dipartimento di Informatica e Sistemistica
Via Claudio, 21
80125, Napoli
fax. +39 081 7683816
{maudarie, mesposit, spromano, giorgio}@unina.it

* ITEM - Laboratorio Nazionale CINI per l'Informatica e la Telematica Multimediali
Via Diocleziano, 328 - 80125, Napoli

Abstract

While proposals for improvements to be introduced in the network in order to support a real differentiation of services follow each other, network operators seem to face these requirements by making overprovisioning on their IP infrastructures. However, no mechanism is offered to request different end-to-end services at different prices.

At the moment the interconnections among IP networks are established by means of SLAs which require high manual overhead and a high associated cost which limits their wide-spread use. This causes scarce providers' revenues and a non-optimised resource allocation of resources inside the network. Hence, it seems now clear that such agreements cannot be managed statically and on a long term timescale, and that the issue of an automatic SLA management has to be addressed.

This paper introduces a new entity based on emerging Elastic Networks paradigm for dynamic management of SLAs, and describes the interaction of such an entity with an architecture capable to provide SLA enforcement on top of QoS-enabled networks.

Keywords: Service Level Agreements, QoS, Elastic Networks

1. Introduction

Although research has been debating for several years about the introduction of assured services inside the network, none of the proposed models seems to offer by itself a definitive solution to the problem. The most advanced models for QoS deployment like *DiffServ* or MPLS, still have to define in more detail the interaction with other mechanisms needed in future networks. In particular, it should be defined how the services are requested (i.e. whether or not to use signalling), how the network can be configured (explicit admission control, either *in-band* or *out-of-band*), and how and where the packets have to be marked.

In the meanwhile, network operators keep on investing a lot of money in order to increase the capacity of their networks and they still support just one kind of service, that is simple network connectivity. Currently, end-to-end connections are carried out by interconnecting IP networks owned by different operators. Such interconnections are regulated by *Service Level Agreements* (SLA). An SLA is a contract between two parties: one is the customer and the other is the provider of a specific service. In the following, we will consider "customer" of a service either a single user or a network operator. Of course, entities which establish an SLA as customers can in turn be providers for other entities. In any case, the SLA contains the rules for the service required.

Nonetheless, nowadays there are no automatic processes for the implementation of the negotiated SLAs which thus have to be instantiated by manual intervention, and of course at a high cost. That makes this process suitable only for large services, while the current requirements seem clearly to indicate the need of *smaller services over smaller timescales* [1]. Besides this, network operators should be interested both in the ability of distributing traffic efficiently in their networks and in the possibility of making a diversification of their offer. To this purpose, they might offer services either per-flow or per-user, with different ranges of price. Finally, they should equip their networks with mechanisms that allow them to be able to re-negotiate SLAs in case of change in current requirements with respect to QoS offered, established prices or altered network conditions.

Therefore, the need arises for an automatic SLA management process. The aims of this work are the definition of an architecture for the implementation of SLAs and the design of an entity capable to operate an automatic SLA request.

Next section discusses the state of the art in SLA definition and introduces an architecture for the dynamic implementation of SLAs. In section 3 a short description of the emerging Elastic Network paradigm is presented, and

0-7803-7661-7/03/$17.00 2003© IEEE

the design of a node for dynamic operation inside the network is depicted. Section 4 describes a model useful for SLA management and, finally, section 5 reports our conclusions and future work.

2. Framework for implementation of SLAs

The interactions among network entities are regulated by means of SLAs. To ease the processing of a service request, the SLA contains a high level description of a specific service. Only when an SLA is finally subscribed, a translation into a more technical document is operated. Such a new document is called *Service Level Specification* (SLS) and contains the real network parameters needed to drive the network configuration process.

Basically, there are two kinds of agreements: *transit agreements* and *peering agreements*. The former concerns the interaction between a small provider and a bigger one, so the first provider becomes customer of the services offered by the second one. The latter fixes the rules between peering network operators.

Besides, SLAs can be classified in *retail*-SLA and *wholesale*-SLA. The retail SLA refers to the agreements between an end-user and a service provider. The end-user might be either a single person or a users' organisation (e.g. a corporate network). Such an end-user could be induced to establish an SLA with his provider in order to support different kinds of applications. Conversely, a wholesale-SLA is an agreement between network operators, and takes into account traffic aggregates flowing from one domain to another. In general there is no direct connection between r-SLA and w-SLA. In particular, w-SLAs might not be based on parameters related to a single service but rather focus on statistical indicators related to the entire resource bundle provided by one provider to one of its neighbours [2].

In this context, we rely on a new proposal for an SLA-aware architecture capable to provide dynamic creation and provisioning of QoS based communication services on top of so called *Premium IP* (PIP) networks [3]. PIP networks are networks composed of heterogeneous QoS-aware network elements. The final aim of this architecture is to deliver services according to SLA requests. To the purpose, it is based on key functional blocks needed to operate the entire process of a mediation between customers' requirements and the appropriate network configuration and management of the QoS-aware network elements available in the underlying network infrastructure. In particular, the combined role of these blocks accomplishes the task of the formal translation from the SLA to the SLS. The functional blocks are:

- AM – Access Mediator
- SM – Service Mediator
- RM – Resource Mediator.

The AM is the entity that receives requests from customers. It is responsible for forwarding a query to one or more SMs, which are the service providers, in order to start the trading process.

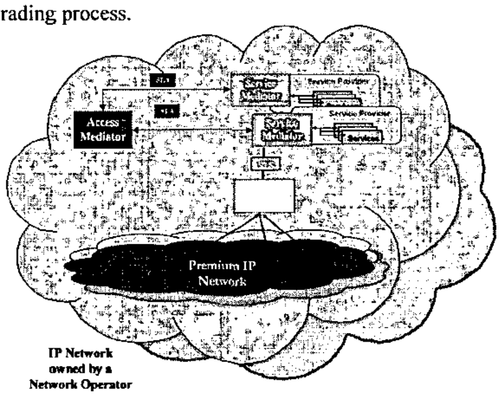

Fig. 1 – Architecture for SLA implementation

The SMs verify if there are available resources on the underlying network for required service and prepare an offer that is sent back to the AM. Finally, the AM presents the SLA to the customer. In case the customer agrees with the terms of the contract, the SM will translate this SLA into the corresponding SLS, while the RM will be in charge of operating the device configuration.

The main role of the AM thus consists in assisting and easing the service selection process. This functionality may be under the control of a trusted third party and appears to offer excellent novel opportunities for a value-added service provider. The Access Mediator may form associations with one or more Service Mediators to which requests are issued.

The Service Mediator has an important role, as this is the place where services are created and from where the impacts of service reconfigurations are communicated to the network resource management. It also has to check that the addition of a new service, or invocation of an existing one, will not affect the services that are currently operational. It is the task of the Service Mediator to prepare the SLA for the user to sign, and subsequently map the SLA from the Access Mediator into the associated SLS(s) to be instantiated in co-operation with the Resource Mediator(s).

In this scenario, a policy based approach is a possible solution to ensure the correct operation of the network. Subsequent to the service creation, a policy extension could be applied to the network to ensure that all services can be managed correctly. The system would have a global view of the configuration of the devices (including an accounting system) and of the policy rules to be applied. In such a case, it would be the function of the Service Mediator to update the service level management system with new rules and configuration as required, in conjunction with the Resource Mediators.

The communication between the Service Mediator and the Resource Mediator should be generic (i.e. independent of the technology employed by the underlying network). According to our design, it is the Resource Mediator that will hold the current view of the complete network of its competence, by communicating with all the appropriate underlying network management systems. A network provider wishing to offer its resources should support an interface capable of handling messages defining an SLS, from its network management system to one or more Resource Mediator(s).

Since there can be more than one Resource Mediator, a Service Mediator can issue identical requests for information about network resource availability to several Resource Mediators. The Resource Mediators will either act on their own image of the network, or explicitly enquire to the individual network management systems, before returning an answer to the Service Mediator. The Service Mediator will accept the best offer, on the basis of the current policy decisions.

In order for the Resource Mediator to maintain and update *the edge-to-edge network view of the current QoS* availability, it may use a set of policy rules that are agreed with the underlying network management systems. A common feature of the communication process surrounding the Access Mediator, Service Mediator and Resource Mediator components is a "one-to-many, search-and-selection" mechanism. In particular:

- the Access Mediator is responsible for selecting the appropriate Service Mediator(s), according to the user's request;
- the Service Mediator is responsible for finding - and, in some cases, building from individual elements - the service, requesting information from (and then selecting) the appropriate Resource Mediator(s);
- the Resource Mediators are responsible for selecting the appropriate network capabilities, given several available options.

3. Elastic Networks Model

Because of new flexibility requirements inside networks, during last years a new technology has been proposed: the Programmable Network technology. While this represents a quite recent proposal (1995), since then a part of the network was experiencing an evolution towards this direction. Nodes like Firewalls, NAT or Implicit Proxies are the first examples of pseudo-programmable nodes, and they face new requirements like security issues, IP addresses saturation, traffic restriction.

Basically, a programmable network is composed of nodes whose behaviour can be modified. In this context, two different schools of thought have emerged [4]:

- The Open Control Network (OCN or OpenSig);
- The Active Network technology (AN).

The former argues that intermediate nodes, like routers and switches, should have to be modified in order to allow open control on their network interfaces. Usually this approach operates on control and management planes, and no computation is performed on data path. The AN technology, instead, represents a more extreme approach and aims at introducing programmability at all levels inside the network. By "injecting" code in the active nodes, they can dynamically deploy new services and perform computations on data flowing through them. The code can be either carried in packets (*capsules*) or downloaded from a secure server. From this perspective, the AN could be seen as the infrastructure for deployment of Mobile Agents (MA) [5]. Although this is the most complete approach, the introduction of computation on data path imposes a big overhead on intermediate nodes.

After several proposals of AN platforms supporting different features and the demonstration of some possible applications of this technology, the new idea of a fusion of the previous approaches has emerged. The Elastic Networks [6] aim at getting over the limitations of the traditional models by proposing an OCN platform which has the flexibility proper of the AN architecture, in particular it has to provide support for the code installation. However, it is not supposed to perform computations on data path, thus, the whole node performances are not compromised.

In this context, we designed an Elastic Network Node. In figure 2 our prototype is depicted. It is composed of two parts: a "fixed" part and a "relocatable" part. The former is composed of a resource Traffic Controller (TC) operating on the network interfaces and managed by a Routing Control Module (RCM), plus a separate Packet Capture Module (PCM). This PCM simply acts as a listener of traffic by making copies of packets flowing through the node, it doesn't make any divert operation on packets. Hence, the subsequent computations are made "off-line". The "relocatable" part is composed of three modules: a Traffic Analyser (TA), a Supervisor (S) and an Interaction Module (IM) for co-ordination with other Elastic Modules.

By implementing special purpose modules and protocols for exchanging information among Elastic Nodes in the network, this kind of architecture might be useful in several situations: network management, security, remote configuration, traffic measurement and monitoring.

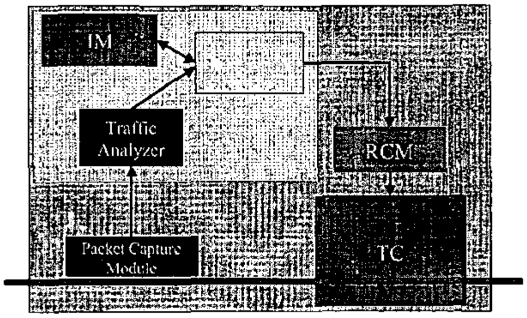

Fig. 2 – Elastic Network Node

4. Dynamic management of SLA

To the purpose of automating the SLA implementation, new functionality has to be introduced in the network management plane. In particular, according to [7], network monitoring is the most fundamental aspect of automated network management. This is the reason that suggested us the use of the proposed Elastic Network Node for the dynamic management of SLAs. As we have discussed in the Section 1, there are two different situations in which an SLA can be negotiated: between an user and an Access Network Provider (*retail-SLA*), and between Network Providers (*wholesale-SLA*). In both cases, currently these agreements cause a manual intervention for the correct network configuration and thus require a long time to be instantiated. This means that only long term services can be positively arranged. However, the current situation clearly seems to indicate an interest in short term services. Being able to provide such services in a timely fashion definitely represents a big opportunity of innovation for network providers or third-party entities.

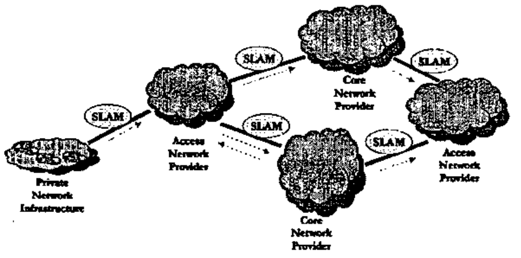

Fig.3 – SLA Management

By introducing the Elastic Network Node both between each pair of Network Providers and between corporate networks and Access Network Providers, it becomes possible to operate some statistical evaluations on traffic flowing among them in order to produce an automatic request of the most appropriate SLAs among various network operators. We call SLAM (SLA Manager) this kind of application of the Elastic Network Node.

The need of an entity which aggregates requests coming from the users or from providers is also motivated by scalability reasons. It is self-evident that when the number of users increases, many different requirements come into play. Considering a separate SLA for each different requirement coming from each user would cause a big SLS jam to be accommodated in each network domain.

In case of a corporate network, for instance, the SLAM would be an entity pertaining to this organisation and could opportunely evaluate the current requirements in order to make an SLA request on behalf of its internal users.

To make automatic request for SLAs, we envisage the necessity of three different statistical evaluations on three different timescales.

Time scale	Action
Short term	Immediate SLA
Medium term	Re-negotiation of an existing SLA
Long term	Generate a profile of a customer and find his ideal SLA

In case of occasional consumer of a particular service or occasional consumers of that area (e.g. mobile users), the statistical evaluations have to be made in a short time and consequently an immediate SLA for that consumer will be requested. The task of the SLAM doesn't conclude after a successful SLA subscription, but the flows belonging to the accommodated SLA are continuously monitored, and in case of modified conditions with respect to either the quality offered for that particular service, or the price agreed, the SLA in question can be either re-negotiated or replaced by a brand new one. Finally, one of the most important features is the long term evaluation. The statistical evaluation on long term time scales made on traffic generated by usual users of the domain can give results useful to understand their real needs. This can bring to the definition of an ideal profile of users and further to the arrangement of the corresponding SLA.

With respect to the implementation, we did some experiences in configuration of services using AN platforms [8]. Our aim is to integrate some functionality coming from different tools in an Elastic Node. What we need is simple active network functionality like remote installation and execution code plus a resource control module. The Darwin architecture [9] seems to be a good candidate for that. Besides, we need a traffic analyser toolkit. There are many open source proposals from the CAIDA organisation [10]; most of them are based on the packet capture library libpcap [11]. Among them, the ntop [12] toolkit provides several functionality for statistical traffic measurements. Finally, a special purpose supervisor should be implemented in order to forward the requests to the AM.

5. Conclusions and Future work

This paper covered the aspects related to the management of network resources via SLA subscription. Usually these contracts are negotiated by network operators to arrange connections among their networks. Actually the SLA management is currently carried out by manual intervention and only long term services can be established. The introduction of automatic mechanisms for SLA management can lead to a cost reduction and enable the creation of short term services. Here we proposed a model for automatic management of SLA subscription, based on Elastic Network technology and capable of interacting with an architecture for the dynamic implementation of SLAs. By introducing this module among SLA-aware networks, we might perform statistical evaluations of traffic flowing between each pair of networks managed by two different operators and eventually subscribe the most appropriate SLA. Besides the real implementation and experimentation of such an SLA Manager, new studies can be carried out on the nature of traffic flowing through the networks in order to better understand its characteristics, especially with respect to the portion of packets flowing at the edge of the networks.

References

[1] T. Roscoe, and B. Lyles. "Distributing Computing without DPEs: Design Consideration for Public Computing Platforms". In *9th ACM SIGOPS European Workshop, Kolding, Denmark*, September 2000.

[2] S.P. Romano, M. Esposito, G. Ventre, G. Cortese. "Service Level Agreements for Premium IP Networks" IETF draft `draft-cadenus-sla-00.txt`, November 2000, http://www.cadenus.org/papers - work in progreess.

[3] S. D'Antonio, M.D'Arienzo, M. Esposito, S.P. Romano, and G.Ventre. "Beyond Quality of Service: a Framework for Creating and Trading Multimedia Service with Quality". Submitted to *ACM Multimedia System Journal*. http://www.cadenus.org

[4] A.T. Campbell, H.G. DeMeer, M.E. Kouvanis, K. Miki, J.B. Vicente, and D. Villela. "A Survey of Programmable Networks". *Computer communication Review, 29(2):7-23*, April 1999.

[5] R.Boutaba, A. Polyrakis "Projecting Advance Enterprise Network and Service Management to Active Networks". *IEEE Network*, Janauary/February 2002.

[6] H. Bos, R. Isaacs, R. Mortier, and I. Leslie, "Elastic Network Control: An Alternative to Active Networks". *Journal of Communications and Networks, vol. XX, no. Y*, March 2001.

[7] W. Stallings "SNMP, SNMPv2, SNMPv3 and RMON 1 and 2". *Addison Wesley*, September 1999.

[8] R.Maresca, M.D'Arienzo, M.Esposito, S.P.Romano, G.Ventre "An Active Network approach to Virtual Private Networks". In *proceedings of ISC2002*, July 2002.

[9] Jun Gao, P.Steenkiste,E.Takahashi, and A.Fisher. "A Programmable Router Architecture Supporting Control Plane Extensibility". IEEE Communications Magazine, March 2000.

[10] www.caida.org

[11] www.tcpdump.org

[12] L. Deri, S. Buin "Effective Traffic Measurements using Ntop". *IEEE Communications Magazine*, May 2000.