Architecture for the Creation of Service Level Agreements and Activation of IP Added Value Services

Dimitrios Kagklis, Nicolas Liampotis, Christos Tsakiris

Telecommunication Laboratory - Department of the Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
{kaglis, nliam, chrtsak}@telecom.ntua.gr

***Abstract*— The pervasive influence of the internet as long as the insatiable desire for bandwidth have led the researchers to promote massive and disruptive changes within the Internet technology. The end goal is to enable the rapid development of a wide variety of services and support them in a unified and consistent manner. As far as the customer view is concerned, means for specifying dynamically the services each customer wishes to be subscribed offered by an ISP are needed. Moreover an IP QoS signaling protocol is required in order for the customer to activate added value services. We propose system architecture for dynamic service subscription based on a negotiation strategy and an approach for explicit activation of services. These models offer the customer an automated manner to specify, select, negotiate, activate and make use of added value services with specific characteristics. The novelty of the proposed approach is the reduction of the operational cost of an ISP and the time needed for a customer to request and access services.**

***Keywords*- IP; QoS; services; SLS/SLA; Signaling**

I. INTRODUCTION

The simplicity of the IP protocol has led to the fact "IP over everything". However the need for deployment of value added IP services has yielded a thorough research on how to provide Quality of Service (QoS) for network applications. QoS means providing consistent predictable data delivery service, in simple terms, satisfying customer application requirements. But "best effort" can make no guarantees about when data will deliver, or how much it can deliver. Increasing bandwidth, which is the obvious solution, is inevitable but the need for efficient protocols, providing differentiated high quality services, still remains.

The Differentiated service (DiffServ) architecture was proposed by IETF for providing end-to-end guaranteed IP services. The DiffServ domain separates the network elements into two groups: a) the boundary routers, which mark, police, meter and classify the packets and b) the core routers, which classify the aggregated flows. The classification of the aggregated flows is performed according to the Service Level Agreement (SLA) signed between each customer and the Internet Service Provider (ISP) that supports the DiffServ model [1]. SLA is a legal service contract between a customer and a service provider that specifies the forwarding service a customer should receive [2]. This contract includes levels of availability, serviceability, performance, operation and other attributes of the service.

There is still no standard method for creating a SLA although a number of research projects have tackled with that problem. Until now, the communication between the customer and the ISP in order to result in a specific SLA is carried out manually.

The dynamic creation of a SLA is in simple terms the procedure through which a customer can electronically specify the characteristics of the service he wishes to have and the network will decide whether it can accommodate the customer or not. The interaction between ISP and customer must be done through a standard form based on the SLS-template philosophy [4]. We call this standard form "*service template*" [5]. The resulted dynamic SLA gives the customer the potentiality to have access to registered services. A customer may be an organization or another service provider or a residential user. The customer includes a set of sites. With the term sites we define the set of users, which are located in the same network area.

Nowadays, the model for providing added value services is that ISPs need only to sign a SLA with the customer. This kind of services is called *implicit* services [6]. The SLA gives each customer the potentiality to access the service according to the terms of the contract. In the typical scenario where the contract has a time limit and the customer is constrained by the SLA and wishes to make use of a service, he must make an *explicit* service request for activation.

However, the *explicit* activation of services specified without signing a SLA must be done with means of signaling and admission control mechanisms. The Resource reSerVation Protocol (RSVP) has been proposed as an IP QoS signaling protocol and with the enhancements supporting Multi Protocol Label Switching (MPLS) is a feasible and effective solution. We propose a signaling protocol that is based on RSVP.

This paper proposes a dynamic algorithm through which both customer and ISP can result in a SLA, as it has been adopted by the TEMPO[1] project, an ongoing Greek research

[1] TEMPO: TEchnologies and applications for MultiPrOtocol networks

and development project, funded by the Research Laboratories of Hellenic Telecommunications Organizations S.A (OTE). Moreover, a signaling mechanism through which the customer can activate a service without the need of a SLA is defined.

The structure of this paper is the following: In the second section the algorithm for the SLA creation is proposed. The algorithm consists of two phases, the filling of the SLA form and the negotiation phase between ISP and customer. In the third section the architecture for activating added value services is illustrated according to the status of already existing networks and protocols. In the final section, a summary of the proposed architecture, as well as the conclusion are stated.

II. SLA CREATION

In this section we describe the main functions that are performed, as well as the interfaces and the interoperability among them and the data structures that hold information needed for the SLA creation procedure. This procedure is performed in two phases, the service template completion phase and the negotiation one. Moreover, the network requirements and the assumptions needed to be taken into account are stated.

A. Assumptions

There are a number of assumptions and a new terminology related to the dynamic service subscription model.

Firstly, both the Customer and the ISP have a pre-determined common filled-form document for their interaction, the *service template* [5], which contains a number of information entities. There are specific constraints regarding the data type of the information entities as well as their position and role in the document.

Secondly, a customer may be an organization or another service provider or a residential user. The customer includes a set of sites. With the term sites we define the set of users who are located in the same network area.

Thirdly, the ISP has an automated system for SLS manipulation.

Fourthly, each time a customer connects to the ISP, he receives all the information for the available customer identifications sites and users that are available to have access to. Also it supports an automated procedure through which it informs each customer for the available services.

B. Service Template Completion

The functional model and the interactions between the different blocks for the service template completion are shown in Fig. 1.

Primary, for the SLA creation the **Customer Repository** stores the information, such as the service template, the customer identification, the subscribed user and sites and the ISP's available servers, retrieved from the ISP. The main roles of the **Customer Repository** are to manipulate the stored data and provide other blocks with means in order to retrieve, modify and delete the stored data.

Then, this block feeds the **SLS Parsing** block with the predetermined service template and the customer information. The functionality of the **SLS Parsing** block is to process all of the above info and create a dynamic algorithm through which the customer is guided to fill the free attributes of the supplied service template. The dynamic algorithm, which comprises a set of rules, is created according to the structure of the service template.

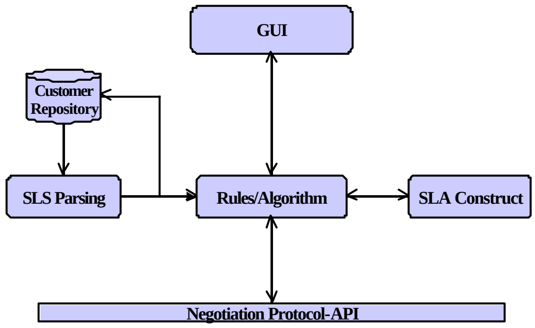

Figure 1. SLA Creation Functional Model

The set of rules specifies exactly the free fields of the service template that have to be completed mandatory or optionally and presupposes a concrete sequence of steps taken for a successful completion. Furthermore, the **Rules and Algorithm** block performs the mentioned sequence of steps and, in collaboration with the **GUI** block; it fills the fields for completion of the service template. The **GUI** block provides the customer with a user-friendly graphical interface in order to fill the contract in an automated way and subscribe for a service of the corresponding platform of the ISP. The customer can specify his demands about a service in a high level logic, without having special technical knowledge. After a successful verification of the produced service template by the **Rules and Algorithm** block, the given values of the filled template are passed to the **SLA Construct** block for the final step of the SLA creation procedure.

In conclusion, the **SLA Construct** block uses the supplied values to create a service instance, which is conformed to the template and the rules of the dynamic algorithm. The created service instance is the contract, which will be negotiated with the ISP through the **Negotiation Protocol API**.

C. Negotiation

The **Negotiation Protocol API** is a software component that accepts requirements, offers an input, negotiates with its peer and delivers a deal or conflict deal. This protocol should allow clients and servers in distributed object systems to negotiate for QoS agreements and pursue an agreement on the content of a specific service proposal. Several protocols exist that allow a client to attain from a distributed system a service with specific characteristics and guaranteed performance. There is still no standard way for service negotiation allowing each ISP using each own technique.

A protocol that fulfills at least the following conditions is required:

- The client can define and request a service level

- The ISP can accept or reject a request submitted by the customer
- The ISP is also able to propose an alternative service level to the one requested
- The customer is capable of accepting or rejecting the ISP's offer and creating alternative requests

A protocol that meets the above requirements and is appropriate for the implementation of the proposed architecture is COPS-SLS [7]. This is an extension of the Common Open Policy Service (COPS [8]) protocol, which utilizes its distinctive features, in particular for the negotiation of service levels. Using COPS-SLS for the definition and the negotiation of SLAs, enables automatic and dynamic QoS provisioning. It is TCP-based and uses a client/server model. The **PEP** (Policy Enforcement Point) and the **PDP** (Policy Decision Point) entities of the protocol hold the place of the customer and the ISP respectively.

The **PEP**, the customer in our architecture, requests a service level sending a Request (REQ) message to the **PDP**, the ISP of our architecture, and the latter responds by sending a Decision (DEC) message which denotes if the submitted request is accepted or not. Alternative proposals by either the customer or the ISP are sent including them in the same messages of the COPS-SLS protocol, REQ for the customer and DEC for the ISP.

However, the COPS-SLS protocol does not only implement the basic operations of a negotiation process but it is also defined by such attributes that suit the specific needs of the dynamic service subscription architecture. It encompasses two phases, the configuration one, where the **PEP** initiates the procedure and the **PDP** replies by providing information which is essential for its progress, and the main negotiation one. Thus, in the former phase the ISP, or the **PDP** in the protocol's terms, can supply the information that we have defined as essential to be sent to the customer, whereas the second phase, provides us with a manageable and effective mechanism for the realization of the negotiation procedure.

The SLSs, which are converted to XML files and are exchanged between the customer and the ISP, are sent as a ClientSI[7] object of a COPS-SLS message, either Report (RPT) or Decision (DEC).

Below we describe the operation for the activation of services without the existence of a SLA.

III. Service Activation

In the following paragraphs we describe the approach of the model we propose, the technologies and protocols that we have adopted and finally the machinery of the proposed explicit service activation architecture.

A. Approach

So far many signaling protocols have been proposed for providing IP QoS. The current status indicates that the network architecture for providing added value services to customers is the combination of the IntServ model for the customer side and the DiffServ model for the network side.

The RSVP, which was introduced by the Integrated Service (IntServ) model, provides the mechanism to apportion network resources according to an application QoS request and subject to bandwidth management policy. The reservation of resources for aggregated flows has been proposed in the context of DiffServ/MPLS architecture as means for reducing significantly the signaling load and the state information stored at routers, while still providing the same QoS for real time flows. An extension to RSVP the RSVP-TE [9] was recently defined to support aggregation.

We adopted this approach and based on [10] we propose a signaling protocol that uses the RSVP as a pure transport mechanism. The architecture for the explicit activation of services is highlighted in Fig. 2.

We use the RSVP protocol for exploiting encapsulated signaling messages describing in XML format the characteristics of the requested service. The customer by using the enhanced GUI block for explicit services of the already described SLA creation functional model selects the characteristics of the service and when finishes the GUI block passes to the customer side signaling layer the XML document that describes the service's characteristics.

The manipulation of the encapsulated XML document by the ISP, describing the policy of the service is performed by admission control functions, which are out of scope of this article.

The signaling component manages the activation signaling state machine, encapsulates the service activation requests to the appropriate RSVP messages and uses the RSVP machinery to communicate with its peers. The RSVP sub-component implements enhancements to the standard RSVP, so as to transport the signaling messages. Upon the reception of an RSVP message containing signaling information the appropriate notification is launched to the signaling sub-component.

B. Machinery

At the signaling zero phases all edge nodes initialize two RSVP sessions, one for communication with the customer and one for communication between network edges. These sessions are active throughout the operational phase of the system. RSVP messages belonging to other sessions are ignored by the signaling protocol.

In order to overcome the lack of control in the RSVP messages on commercial routers the COPS outsourcing facilities are used. The COPS protocol regulates the treatment of the received RSVP messages but does not initiate RSVP sessions nor generates RSVP messages itself. Therefore, the COPS is used only to capture and respond to the RSVP PATH messages while the RSVP will be still the mechanism to request and receive replies from peer edges. Since the interface with the signaling sub-component must be common with both types of routers (commercial and Linux), a common RSVP part must be build.

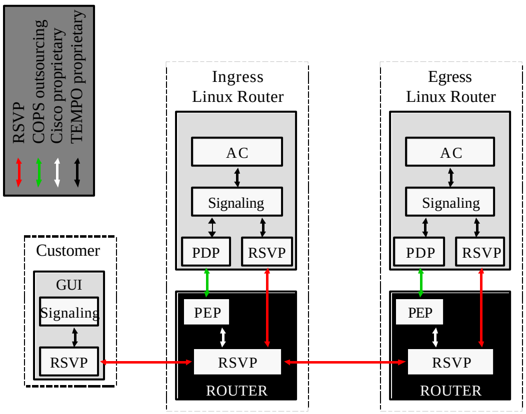

Figure 2. Service Activation Architecture

The machinery of the protocol of the service activation is better shown in Fig. 2 and described by the following scenario.

The RSVP messages received in a router are handled by the PEP of the COPS protocol (white arrow). The PEP forwards the received RSVP message to the PDP (green arrow). The PDP will extract the RSVP information and pass it as an event up-call to the signaling sub-component, which in turn will appropriately notify the AC. In order to communicate with the egress the signaling layer will trigger the RSVP protocol to send a new PATH message to the egress having the Router as the first hop. Upon confirmation of the request admission, the PDP will be instructed to respond with a RESV message to the originally received PATH message.

C. *Novelty Of the Proposed Model*

The advantages of the proposed model are:

1. The XML document that specifies the service characteristics and is propagated to the ISP provides the ISP with means for Authentication Authorization and Accounting. If the customer used only the RSVP protocol for service activation the ISP could never know at any time the identity of customers and the utilization of its network resources.
2. The enhancements and the protocols we have implemented were designed in order to be compliant with existing network status and IP network protocols and standards (Cisco routers, IETF standards).
3. Our model has been tested successfully in the testbed of the Multimedia Laboratories of the Research Department of OTE. The implemented code was running in Linux PCs and we have used two Cisco routers of the 3640 series attached to the customer and ingress Linux PC and a 7200 series Cisco router attached to the egress Linux PC. We have performed integration of our software subcomponents and interoperability among them and the Cisco routers. The test-bed supported MPLS and the Cisco routers communicate with the Open Short Path First (OSPF) routing protocol.

IV. CONCLUSION

In this article, architecture for creating and activating a Service Level Agreement is described. This architecture is defined in the particular context of networks supporting mechanisms for SLS manipulation. We described an efficient and simple way for the customer to dynamically create and activate a Service Level Agreement.

The proposed model supplies with means for enabling automated service subscription, which for ISPs reduces the operational cost and contributes to an integrated and fully automated service provisioning process and for customers, reduces the time to request and access services. Finally, this architecture allows a customer to receive from a distributed IP network system added value services with specific characteristics and guaranteed performance with means of signaling mechanisms.

In future work we intend to evaluate the performance of our model in cases of multiple service activation requests and analyze the results especially when our model fails to accommodate the customer's request.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated services," RFC 2475, December 1998.

[2] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog et al., "Terminology for policy-based management," RFC 3198, November 2001.

[3] A. Prieto and M. Brunner, "SLS to DiffServ configuration mapping," Distributed Systems: Operations And Management, DSOM, 2001.

[4] Danny Goderis, Yves T'joens, Christian Jacquenet, George Memenios, George Pavlou, Richard Egan et al., "Service Level Specification semantics, parameters and negotiation requirements," Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress, June 2001.

[5] IST-CADENUS Homepage - http://www.cadenus.org.

[6] IST-TEQUILA Homepage - http://www.ist-tequila.org.

[7] T.M.T. Nguyen, N. Boukhatem, Y. Ghami Doudane, G. Pujolle, "COPS SLS: A service level negotiation protocol for the Internet," IEEE Communications Magazines, May 2002.

[8] S. Herzog, R. Cohen, D. Durham, S. Herzog, R. Rajan and A. Sastry, "COPS usage for RSVP," RFC 2749, January 2000.

[9] D.Awduche, et al, "RSVP-TE: Extension to RSVP for LSP Tunnels" RFC 3209, December 2001.

[10] RSVP-TE daemon for DiffServ over MPLS under Linux http://dsmpls.atlantis.rug.ac.be.