

# Policy-based service level negotiation with COPS-SLS

Nguyen Thi Mai Trang, Boukhatem Nadia

*ENST, 46 rue Barrault, 75013 Paris, France*

*e-mail: {trnguyen, nadia.boukhatem}@enst.fr*

*LIP6, 8 rue du Capitaine Scott, 75015 Paris, France*

*e-mail: Thi-Mai-Trang.Nguyen@lip6.fr*

**Abstract:** Data streams in end-to-end communications can cross multiple network domains in which different QoS solutions are deployed. The provision of end-to-end quality of service over heterogeneous environment assumes the negotiation of mutually acceptable SLA. The negotiation process gives the user the ability to specify the desired level of service, along with the permissible pricing of the service. This paper presents the use of COPS-SLS protocol for the negotiation of intra-, inter-domain and end-to-end QoS.

**Key words:** policy-based networking, COPS protocol, quality of service.

## 1. INTRODUCTION

To date, most of service level negotiation process has been carried out manually. When modification is needed, the customer may require the assistance of the network provider. The provider, in turn, may need to reengineer part of the system to ensure the mandated service levels. In this case, the SLA (Service Level Agreement) is considered static. A dynamic SLA, on the other hand, allows a subset of parameters to be modified over time more easily.

In order for the service level negotiation to be dynamic on a large scale, a negotiation protocol is mandatory. We believe that a standardized negotiation protocol can make the Internet more flexible in service level offering and management. Provider, customer, or any software developer

can rely on the protocol to develop different applications for different negotiation purposes and different types of customers. Service level negotiation will become a popular network service like DHCP (Dynamic Host Configuration Protocol) or DNS (Domain Name Server). A client connecting to a network can automatically request his desired service level. Two domains can negotiate different service levels for different traffic streams independently of the underlying QoS technology deployed by the provider.

We have proposed COPS-SLS (COPS usage for SLS negotiation) as a generic service level negotiation protocol [1, 2] providing all needed operations to negotiate and establish a contract. This protocol is defined in the particular context of policy-based networks [3] and uses the flexibility of COPS (Common Open Policy Service) protocol to transport negotiation information between a customer and a provider.

There are recently some projects working on the automation of service deployment in which the client can customize his services through Web portals of his Internet Service Provider (ISP) [4] or use a protocol to negotiate services with the provider [5, 6, 7].

There are two main characteristics differentiating COPS-SLS from the above-mentioned works. First, COPS-SLS uses a PIB (Policy Information Base) to represent SLS (Service Level Specification) information enabling a high level of independence and reusability. Second, the organization of the negotiation process in two phases (Configuration phase and Negotiation phase) allows the automatic configuration of the negotiation process.

The specification and operation of this protocol can be found in [1, 2]. This paper aims to present the use of this protocol for inter-domain, intra-domain, and end-to-end service level negotiation. The organization of the rest of this paper is as follows. Section 2 provides an overview on COPS-SLS protocol and framework. Section 3 presents the use of this protocol in intra- and inter-domain negotiations. Section 4 discusses the end-to-end service level negotiation using COPS-SLS. Finally, section 5 provides the conclusion and future works.

## **2. COPS-SLS OVERVIEW**

COPS-SLS is an extension of COPS protocol [8] to negotiate a service level between a customer and a network. The customer can be an end-user, a subscriber, a local network or an ISP. The network can be a local network or an ISP. Conceptually, this protocol is designed to communicate between a resource consumer representative entity and a resource provider

representative entity. The model of COPS-SLS protocol is illustrated in figure 1.

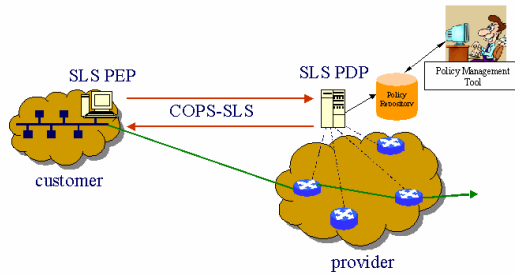


Figure 1. The COPS-SLS model

According to the COPS protocol model, the resource consumer representative entity corresponds to the Policy Enforcement Point (SLS-PEP). The resource provider representative entity corresponds to the Policy Decision Point (SLS-PDP). This mapping fits into the philosophy in which the provider always takes the final decision on service levels of data streams consuming network resources in its domain. Policy Management Tool is a software which assists the administrator to create service level policy in a human friendly manner. These policies are then stocked in Policy Repository and retrieved by the PDP in order to make a decision.

COPS-SLS comprises two phases, configuration and negotiation. In the configuration phase, the PDP uses the Provisioning model [9] to tell the client how to request a level of service. For example, it supplies the client with information about the negotiation mode and the time interval to renegotiate. After successfully installing the configuration applied by the PDP, the PEP can start the negotiation phase using the Outsourcing model [9] and indicating the desired level of service. The PDP can accept the request, reject the request, or propose another level of service to the client.

SLS information exchanged between the PEP and the PDP is represented by a named data structure, also known as Policy Information Base (PIB). Conceptually, a PIB can be described as a set of classes. Each class has a class name and a set of attributes [9].

COPS-SLS allows the provider two SLS negotiation modes: predefined SLS mode and non-predefined SLS mode [1, 2]. A non-predefined SLS mode does not place constraints on the SLS parameter values requested by the client. Predefined SLS constrains the SLS through a subset of parameters with predefined values or value ranges. The predefined SLS mode allows the provider to provision service templates. The client should

verify the constraints of the provider before sending its request for a desired SLS in the negotiation phase.

### 3. INTRA- AND INTER-DOMAIN NEGOTIATIONS

#### 3.1 Intra- and inter-domain negotiation models

According to the COPS-SLS model, there is no significant difference between intra- and inter-domain negotiations in service level negotiation perspective.

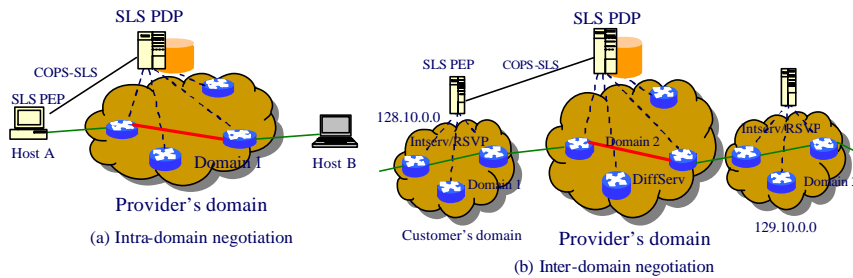


Figure 2. Intra- and inter-domain negotiations

As illustrated in Figure 2, the difference resides in the types of customers and data flows considered in a negotiation process. Indeed, customers in intra-domain negotiations are hosts while customers in inter-domain negotiations are domains. Subsequently, data streams in intra-domain negotiations are usually micro-flows while data streams in inter-domain negotiations are usually aggregates. Other parameters in the negotiation such as traffic conformance, performance, and service schedule parameters [1, 2] do not really differentiate intra- and inter-domain negotiations.

In COPS-SLS PIB [2], above-mentioned difference involves only the class `slsFlowIdParamEntry`. This class represents both micro-flow and aggregate using 13 parameters to identify any layer 3 or layer 4 data stream:

```
slsFlowIdParamEntry OBJECT-TYPE
    SYNTAX SlsFlowIdParamEntry
    STATUS current
    DESCRIPTION
        "The instance of this class identifies a traffic stream"
    ::= { slsFlowIdParamTable 1 }
SlsFlowIdParamEntry ::= SEQUENCE{
    slsFlowIdParamPrid          InstanceId
    slsFlowIdParamAddrType     InetAddressType,
    slsFlowIdParamDstAddr      InetAddress,
    slsFlowIdParamDstPrefixLength  InetAddressPrefixLength
    slsFlowIdParamSrcAddr      InetAddress,
```

```

slsFlowIdParamSrcPrefixLength  InetAddressPrefixLength,
slsFlowIdParamDscp             DscpOrAny,
slsFlowIdParamFlowLabel       Unsigned32,
slsFlowIdParamProtocol         Integer32,
slsFlowIdParamDstL4PortMin     InetPortNumber,
slsFlowIdParamDstL4PortMax     InetPortNumber,
slsFlowIdParamSrcL4PortMin     InetPortNumber,
slsFlowIdParamSrcL4PortMax     InetPortNumber
}

```

To identify a micro-flow, source address (slsFlowIdParamSrcAddr), destination address (slsFlowIdParamDstAddr), source port (slsFlowIdParamDstL4PortMin, slsFlowIdParamDstL4PortMax), destination port (slsFlowIdParamSrcL4PortMin, slsFlowIdParamSrcL4PortMax) and protocol (slsFlowIdParamProtocol) parameters can be used. To identify an aggregate, a DSCP (slsFlowIdParamDscp) value can be used.

Both in intra- and inter-domain negotiations, the customer's and provider's data stream identifications can be different. For example, a host or Intserv/RSVP domain can identify its data streams with source address, destination address, source port, destination port, and protocol. A DiffServ domain can identify its data stream by DSCP (DiffServ Code Point) value. A MPLS (MultiProtocol Label Switching) domain can identify its data stream by the values in MPLS header [7].

If the provider cannot understand the data stream identification of the customer, they cannot recognize the customer's data stream and map it into the corresponding internal data stream identification at the ingress point of the domain. In this case, the provider can ask the customer to make this mapping before entering traffic into its domain assuming that the provider is sure that the customer understands its data stream identification mechanism. Thus, it is necessary to check the data stream identification mechanisms of the customer and the provider before negotiating a service level. For this purpose, COPS-SLS supports this checking in the Configuration phase. During this phase, the customer communicates its data stream identification capabilities to the provider by announcing which parameters of the class slsFlowIdParamEntry it supports.

### 3.2 Example of intra-domain negotiation

This section presents an example using the topology depicted in Figure 2a. Host A negotiates with domain 1 for a data stream towards host B with bandwidth = 64 Kbps, max\_delay = 100 ms, max\_jitter = 25 ms, max\_loss\_ratio = 10E-3. We suppose that domain 1 identifies data stream by micro-flow and use a predefined SLS mode with the following constraints: max\_delay = 100 ms, max\_loss = 10E-3, max\_bandwidth = 1 Mbps. Figure 3 presents the message sequence chart of this negotiation.

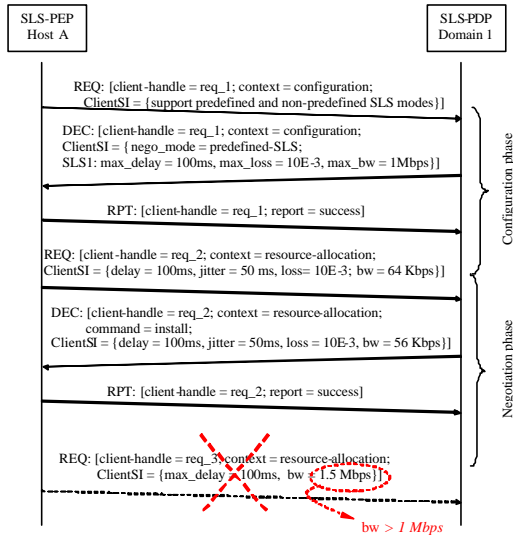


Figure 3. Intra-domain negotiation with predefined-SLS mode

Each request (REQ) message is identified by a client-handle value. The value ‘context = configuration’ indicates that this message is in the Configuration phase. The ClientSI (Client Specific Information) object is used by the PEP to communicate its capability concerning the data stream identification and negotiation modes it supports. In this example, we suppose that host A identifies its data stream on a micro-flow basis and supports both predefined and non-predefined SLS negotiation modes.

In the decision (DEC) message, client-handle value indicates the REQ message to which the DEC message replies. The predefined-SLS is communicated to the PEP in the ClientSI object indicating the constraints about maximum delay, maximum jitter and maximum loss ratio to be used in the request. Once the PEP has installed this configuration, it sends a report (RPT) message with ‘report=success’ to inform the PDP that the configuration has been well installed and the negotiation phase can be started.

It is worth stressing that with the PDP imposed constraints, the PEP systematically and locally checks service level requirements before sending a request to the PDP. In this example, as the jitter demand is not constrained by the PDP, a request for a data stream of 64kbps, max\_delay = 100 ms, max\_jitter = 10E-1, max\_loss\_ratio = 10E-3 is conform to the predefined SLS.

In the REQ message sent by the host A, the value ‘context=resource-allocation’ indicates that this message is in Negotiation phase. The clientSI object is used to convey the desired SLS to the PDP.

Since domain 1 does not have sufficient resources for this request, the PDP sends a decision proposing another SLS specifying ‘bandwidth = 56 Kbps’. Suppose that the application supports many coding mechanisms and is able to change its coding to fit with the constraint. Thus, the PEP sends a ‘report=success’ to accept this service level for the data stream.

If host A, at a given time, needs a data stream of 1.5Mbps, the PEP does not send a request to the PDP for this data stream because the bandwidth requirement is not conform to the constraints of the provider announced during the configuration phase. After accepting a service level, domain 1 makes all the necessary actions to put the contract into effect. The resource allocation in network devices of the domain is out of scope of this document. Many solutions for different QoS mechanisms can be used such as COPS-RSVP [10], COPS for RSVP-PCC [11], COPS-PR for Diffserv [12].

### 3.3 Example of inter-domain negotiation

This section presents an example on Intserv/RSVP – Diffserv inter-domain negotiation. Consider the topology depicted in Figure 2b, domain 1 and domain 3 are two Intserv/RSVP access networks connecting to a Diffserv backbone, which is domain 2. Domain 1 negotiates with domain 2 for a data stream of 1.5 Mbps with maximum delay of 100ms, maximum jitter of 50ms and maximum loss rate of 10E-3. The message sequence chart is presented in Figure 4.

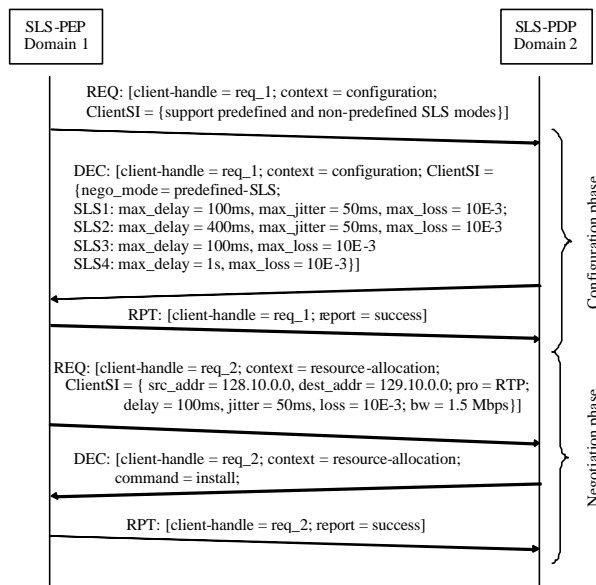


Figure 4. Inter-domain negotiation message sequence chart

In the example of intra-domain negotiation, predefined SLS is used by the provider to place constraints on the performance values which can be used in the REQ message. In this example, we present the negotiation of an aggregate and the capability of the service provider to provision service templates using predefined SLSs. In the Configuration phase illustrated in Figure 4, domain 2 sends a decision message to provision domain 1 four predefined SLSs with different constraints concerning QoS parameter values. The PEP can request a service level conforming to one of the predefined SLSs provided by the PDP.

If domain 2 only supports these four service classes and wants to let domain 1 choose one of them, it may be interesting to define a parameter representing service name or service ID. The PEP can request this service just with the service name or service ID rather than resend all QoS parameters' values in its request. We are considering the possibility of modifying COPS-SLS-PIB to support service name or service ID parameter.

In the Negotiation phase, domain 1 requests a service level for an aggregate sharing 1.5 Mbps between all Real Time Transport (RTP) flows from domain 1 towards domain 3 with  $\text{max\_delay} = 100 \text{ ms}$ ,  $\text{max\_jitter} = 50\text{ms}$ , and  $\text{max\_loss} = 10\text{E-}3$ . This aggregate is identified by the subnet addresses of the source domain (128.10.0.0), the subnet addresses of the destination domain (129.10.0.0), and the protocol value (RTP). If the service ID is supported, the REQ message simply indicates 'serviceID = SLS1' in place of specifying delay, jitter and loss ratio values.

## **4. END-TO-END NEGOTIATION**

### **4.1 End-to-end negotiation model**

In intra- and inter-domain negotiation presented in the previous section, the PDP can decide itself based on the desired service level and the domain policy since the network resources to be guaranteed are within the control of the provider's domain. To guarantee an end-to-end service level for a communication over multiple domains, we need a collaboration of multiple domains along the data path. There are various negotiation models [13] such as bilateral model, hub model and hierarchical model.

COPS-SLS can be used as a signaling protocol for end-to-end QoS negotiation with only some parameters set to particular values. First, the parameter `slsNegoMaxInt` which defines the time interval of renegotiation should be 0 or very small. Second, Service Schedule class should indicate 'immediate' [1]. If the life-time of the communication needs to be signaled, start-time and stop-time parameters in Service Schedule class can be used.

## 4.2 Example of end-to-end negotiation

In this section, we consider an example of end-to-end negotiation illustrated in Figure 5. This example uses the bilateral negotiation model [13]. A domain negotiates with a neighboring domain, which, in turn, negotiates with its neighboring domain until the destination domain.

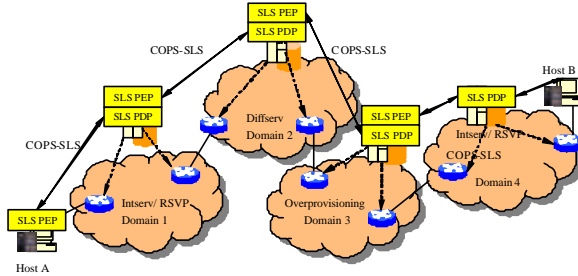


Figure 5. Topology for end-to-end negotiation

In Figure 5, host A negotiates with domain 1 an end-to-end QoS for a data stream of 64 Kbps, maximum delay of 100 ms, maximum jitter of 50 ms, maximum loss rate of  $10E-3$  from A to B. This traffic stream crosses 4 domains. Domain 1 directly managing host A supports Intserv/RSVP. Domain 2 is a backbone deploying DiffServ. Domain 3 is another backbone in which network resources are over-provisioned. Domain 4 directly managing host B supports Intserv/RSVP. The message sequence chart of the negotiation process is depicted in Figure 6. We suppose that all four domains use non-predefined SLS mode. Therefore, there is no constraint in the service level the PEP can request.

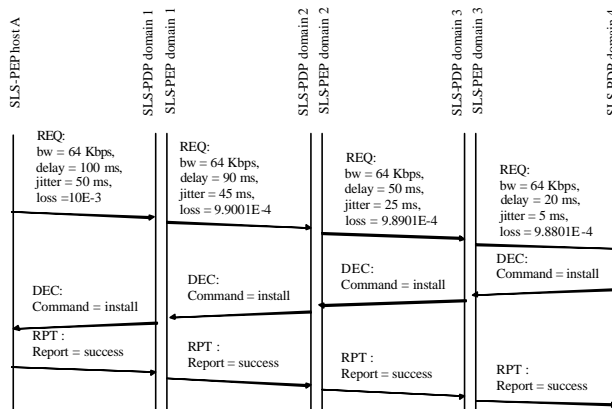


Figure 6. Sequential message diagram of end-to-end negotiation

For the sack of simplicity, only QoS and bandwidth parameters are illustrated in Figure 6. In fact, the SLS in the REQ message sent by host A to domain 1 contains the following information:

```
Context = resource-allocation
SLS = { Flowid: src_addr=A, dest_addr=B, [srcport, destport, protocol].
Traffic conformance: 64kbps, Token bucket
Excess treatment: dropping
QoS: max_delay = 100 ms, max_jitter = 50 ms, max_loss = 10E-3
Service schedule = immediate }
```

	Edge-to-edge service level a domain offers to the traffic stream	Service level calculated by the SLS PEP of a domain to negotiate with its neighboring domain
Domain 1	Class of service: Guaranteed max_delay = 10ms, max_jitter = 5 ms, max_loss = 10E-5	max_delay = 90 ms, max_jitter = 45 ms, max_loss = 9.9001E-4
Domain 2	Class of service: EF max_delay = 40 ms, max_jitter = 20 ms, max_loss = 10E-6	max_delay = 50 ms, max_jitter = 25 ms, max_loss = 9.8901E-4
Domain 3	max_delay = 30 ms, max_jitter = 20 ms max_loss = 10E-6	max_delay = 20 ms, max_jitter = 5 ms, max_loss = 9.8801E-4
Domain 4	max_delay = 10 ms, max_jitter = 5 ms max_loss = 10E-5	

In this example, the SLS PEP uses the following formulas to calculate QoS parameter's values necessary to negotiate with neighboring domain. In fact, the provider can implement other formulas. The issue of algorithm in SLS-PEP is out of scope of COPS-SLS protocol's design.

$$D_{nego} = D_{req} - d_i$$

$$J_{nego} = J_{req} - j_i$$

$$L_{nego} = 1 - (1 - L_{req}) / (1 - l_i)$$

Where:  $i = 1, 2, 3$   
 $D_{req}$ : maximum end-to-end delay specified in the received request  
 $D_{nego}$ : maximum delay that a domain negotiates with its neighboring domain  
 $d_i$ : maximum edge-to-edge delay that domain i offers to the traffic stream  
 $J_{req}$ : maximum jitter specified in the received request  
 $J_{nego}$ : maximum jitter that a domain negotiates with its neighboring domain  
 $j_i$ : maximum edge-to-edge jitter that domain i offers to the traffic stream,  
 $L_{req}$ : maximum loss rate specified in the received request  
 $L_{nego}$ : maximum edge-to-edge loss ratio that a domain negotiates with its neighboring domain  
 $l_i$ : maximum edge-to-edge loss ratio that domain i offers to the traffic stream

Table 1. Requested SLSs and offered service levels

At first, host A sends a REQ message to the SLS PDP of domain 1 to request a data stream of 64 Kbps with max\_delay = 100ms, max\_jitter = 50 ms, max\_loss = 10E-3 as indicated in Table 1.

The SLS PDP of domain 1 retrieves information from the policy repository and decides that it can offer a Guaranteed Service to this request with  $\text{max\_delay} = 10\text{ms}$ ,  $\text{max\_jitter} = 5\text{ms}$ ,  $\text{max\_loss} = 10\text{E-}5$ . Since the SLS PDP of domain 1 is not the final domain of the negotiation, it communicates this request and the above offered service level to the SLS PEP of domain 1. The SLS PEP, on behalf of domain 1, calculates QoS parameters' values to negotiate with its neighboring domain using three formulas in Table 1. Applying above formulas, the PEP of domain 1 builds a REQ message with  $\text{max\_delay} = 90\text{ ms}$ ,  $\text{max\_jitter} = 45\text{ ms}$ ,  $\text{max\_loss} = 9.9001\text{E-}4$  and sends this request to the SLS PDP of domain 2.

Suppose that the SLS PDP of domain 2 offers an Expedited Forwarding (EF) service to this request with  $\text{max\_delay} = 40\text{ ms}$ ,  $\text{max\_jitter} = 20\text{ ms}$ ,  $\text{max\_loss} = 10\text{E-}6$ . Similarly to domain 1, domain 2 is not the final domain of the negotiation process. The SLS PEP of domain 2 sends a REQ message to the SLS PDP of domain 3 specifying  $\text{max\_delay} = 50\text{ ms}$ ,  $\text{max\_jitter} = 25\text{ ms}$  and  $\text{max\_loss} = 9.98901\text{E-}4$ . Suppose that domain 3 offers to this request a service level with  $\text{max\_delay} = 30\text{ ms}$ ,  $\text{max\_jitter} = 20\text{ ms}$ ,  $\text{max\_loss} = 10\text{E-}6$ . Then, the SLS PEP of domain 3 continues to negotiate with domain 4 a service level of  $\text{max\_delay} = 20\text{ ms}$ ,  $\text{max\_jitter} = 5\text{ms}$ ,  $\text{max\_loss} = 9.8801\text{E-}4$ .

Domain 4 finds that host B is in its management scope. Therefore, it is the final domain of the negotiation process. With the QoS offering capability of domain 4 to this request indicated in Table 1, the SLS PDP of domain 4 accepts this request. It sends a DEC message with a 'command = install' to the PEP of domain 3. Upon receipt of the positive decision from domain 4, the SLS PEP of domain 3 sends a decision 'accept' to the SLS PEP of domain 2. The SLS PDP of domain 2 continues the negotiation process by sending a decision 'accept' to the SLS PEP of domain 1. If domain 2 wants to ask domain 1 to mark the packets of the traffic stream, it can include the DSCP value in the decision. After receiving the positive decision from domain 1, the SLS PDP of domain 1 sends a positive decision to host A.

To complete the negotiation process, host A sends a report 'success' to domain 1, domain 1 sends a report 'success' to domain 2, and so on. Upon receipt of a 'report = success', each domain achieves all necessary operations to put the contract into effect. If one of the domains along the path sends a negative decision, the negotiation is failed. The concerned domains send negative decisions until the initiating domain. An end-to-end service level can be negotiated in advance. For example, to prepare a video-conference, the negotiation can be realized some hours in advance by specifying the start-time and maybe the stop-time in the service schedule parameters defined in COPS-SLS-PIB

## 5. CONCLUSION

In this paper, we present the use of COPS-SLS protocol for intra-, inter-domain and end-to-end service level negotiation. First, COPS-SLS allows the negotiation between domains deploying different QoS mechanisms. Second, by supporting predefined SLSs, the provider can place constraints on the service level requests or provision service templates to the customer. Third, to support an end-to-end negotiation, no significant changes are needed in the protocol operation. Only some adaptation such as error-codes and re-negotiation interval values should be achieved. We plan to add to COPS-SLS PIB parameters to support MPLS 's data stream identification and error\_codes needed by different negotiation models. In addition, the relationship between COPS-SLS and existing end-to-end session signaling protocols will be studied.

## 6. REFERENCES

- [1] T.M.T. Nguyen, N. Boukhatem, Y. G. Doudane, G. Pujolle, "COPS-SLS: A service level negotiation protocol for Internet", *IEEE Communications Magazine*, vol. 40, no. 5, May 2002, pp. 158-165.
- [2] T.M.T. Nguyen, N. Boukhatem, Y. El Mghazli, N. Charton, G. Pujolle, " COPS Usage for SLS negotiation (COPS-SLS)", <draft-nguyen-rap-cops-sls-02.txt>", Internet draft, February 2002.
- [3] R. Yavatkar, D. Pendarakis, R. Guerin, "A Framework for Policy Based Admission Control", RFC 2753, January 2000.
- [4] <http://www.cadenus.org/deliverables/d11-final.pdf>
- [5] [http://tequila.telecom.ntua.gr/srnp/toc\\_srnp1.htm](http://tequila.telecom.ntua.gr/srnp/toc_srnp1.htm)
- [6] S. Salsano, "COPS Usage for Diffserv Resource Allocation (COPS-DRA)", Internet draft, October 2001.
- [7] X. Xiao, L. M. Ni, "Internet QoS: A Big Picture", *IEEE Network*, Vol. 13, Issue. 2, March/April 1999, pp. 8-18.
- [8] D. Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.
- [9] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghr ie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith, "COPS Usage for Policy Provisioning (COPS-PR)", RFC 3084, March 2001.
- [10] S. Herzog, Ed., J. Boyle, R. Cohen, D. Durham, R. Rajan, A. Sastry, "COPS usage for RSVP", RFC 2749, January 2000.
- [11] D. Rawlins, L. Yao, R. McClain, A. Kulkarni, "RSVP Policy Control Criteria PIB", <draft-ietf-rap-rsvppcc-pib-01.txt>, Internet draft, March 2002.
- [12] M. Fine, K. McCloghr ie, J. Seligson, K. Chan, S. Hahn, C. Bell, A. Smith, F. Reichmeyer, "Differentiated Services Quality of Service Policy Information Base", <draft-ietf-diffserv-pib-06.txt>, March 2002.
- [13] C. A. Kamienski, D. Sadok, "Service Definition and Negotiation in the Chameleon Architecture", First IEEE International Symposium on Telecommunications, September 2001.