# QoS, Security, and Mobility Management for Fixed and Wireless Networks under Policy-based Techniques

**Guy Pujolle and Hakima Chaouchi**
*LIP6, University of Paris 6, 8 rue du Capitaine Scott, 75015, Paris, France*

**Abstract:** This paper introduces a general policy-based management framework within an IP network. It describes how a policy-based approach can be applied to deal with QoS, security, access control, mobility, etc. The framework presented here is derived from IETF works in Policy Framework working group and in Resource Allocation Protocol working group. Then, we describe some applications that could be handled by policy-based systems. Finally, we present some new evolutions that could be part of the future global policy-based networking architecture.

**Key words:** Internet, QoS, Security, Mobility, Policy-based Management

## 1. INTRODUCTION

The policy-based networking concepts are born from the need to get an overall end-to-end strategy to correlate the business with the overall network actions. Policy-based networking objectives are to deliver a comprehensive architecture that allows the merging of users, applications and resource policy information with network policy actions. The goals of policy-based networking architecture are to address the enforcement of policies in the nodes of the network and to globally manage the system.

A policy may be defined following two perspectives: an explicit goal and actions to guide and determine present and future decisions. Policies are a set of rules to control and manage network resources.

A policy-based networking system defines two main components: a policy decision point (PDP) and policy enforcement points (PEP). The signaling protocol COPS (Common Open Policy Service) is used to communicate policy information between policy enforcement points (PEP) and a remote policy decision point (PDP) within the context of a particular

L. Chapin (ed.), *Communication Systems*

type of client. To get local policy decisions in the absence of a PDP, the PEP can use the optional local policy decision point (LPDP).

This architecture is described in different RFCs [1-9] mainly coming from the work of the rap (Resource Allocation Protocol) and policy (Policy Framework) working groups.

In this paper, we present policy-based networking operations and some new ideas that could be developed to reach a homogeneous structure to control future IP networks. In section 2, we describe the classical policy-based networking (PBN) architecture. In section 3, we present some applications that could be handled by policy-based systems. In section 4, some specific extensions that we developed in our laboratory are introduced. Finally we present some concluding remarks.

## 2. THE BASIC ARCHITECTURE

A policy-based networking (PBN) system needs several components:

- a policy management tool,
- a policy repository,
- a policy decision point (PDP),
- policy enforcement points (PEP).

These components are shown in Figure 1. The policy management tool assists the network manager in the task of constructing and deploying policies, and monitoring status of the policy-managed environment. The policy management tool may be seen as an interface between the network manager and the policy repository.

The policy repository can be defined from two perspectives. First, it can be a specific data store that holds policy rules, their conditions and actions, and related policy data. A database or directory would be an example of such a store. Second, the policy repository may be seen as a logical container representing the administrative scope and naming of policy rules, their conditions and actions, and related policy data. A QoS policy, a security policy or a mobility domain would be an example of such a container.

The policy decision point (PDP) is a logical entity that produces policy decisions for itself or for other network elements that request such decisions. A decision involves actions for enforcement when the conditions of a policy rule are true. Policy enforcement points (PEP) are logical entities that enforce policy decisions.

The PEP may also have the capability to select a local policy decision via its local policy decision point (LPDP). However, the PDP remains the authoritative decision point at all times. This means that the relevant local decision information must be relayed to the PDP. That is, the PDP must be granted access to all relevant information to select a final policy decision. To facilitate this functionality, the PEP must send its local decision information (using its LPDP) to the remote PDP.

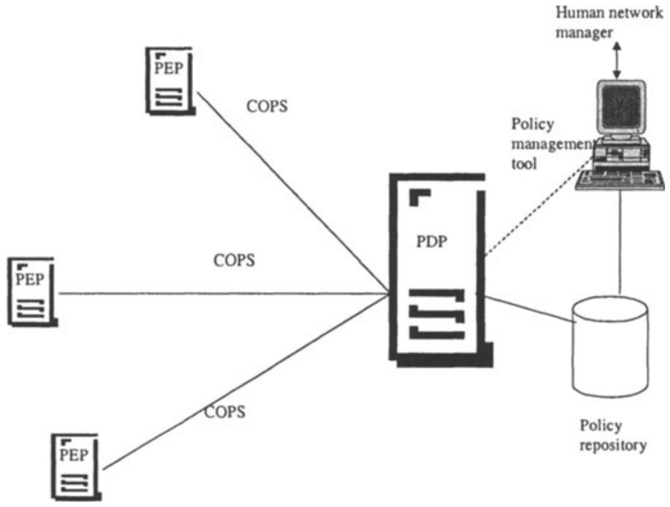

*Figure 1* – The basic PBN architecture

The signaling protocol, COPS, is a simple query/response TCP-based protocol that can be used to exchange policy information between a PDP and its clients, the PEPs. Then, a PEP is responsible for initiating a persistent TCP connection to a PDP. The PEP uses this TCP connection to send requests to and receive decisions from the remote PDP. Communication between the PEP and remote PDP is mainly in the form of a stateful request/decision exchange, though the remote PDP may occasionally send unsolicited decisions to the PEP to force changes in previously approved request states. The PEP also has the capacity to report to the remote PDP that it has successfully completed performing the PDP's decision locally. This capability is useful for accounting and monitoring purposes. The PEP is responsible for notifying the PDP when a request state has changed on the PEP. Finally, the PEP is responsible for the deletion of any state that is no longer applicable due to events on the client side or decisions issued by the PDP. When the PEP sends a configuration request, it expects the PDP to send configuration data via decision messages as applicable for the configuration request. When a policy is successfully installed on the PEP, the PEP has to send a report message to the PDP confirming the installation. The server may then update or remove the configuration information via a new decision message. When the PDP sends a decision to remove a configuration from the PEP, the PEP will delete the specified configuration and send a report message to the PDP as a confirmation.

COPS protocol is designed to communicate self-identifying objects which contain the data necessary for identifying request states, establishing the context for a request, identifying the type of request, referencing previously installed requests, relaying policy decisions, reporting errors, providing message integrity, and transferring client specific/namespace information.

To distinguish between different kinds of clients, the type of client is identified in each message. Different types of clients may have different client specific data and may require different kinds of policy decisions. It is expected that each new client-type will have a corresponding usage RFC specifying its interaction within COPS protocol.

The COPS context object identifies the type of request and message that triggered a policy event via its message type and request type fields. COPS identifies three types of outsourcing events: (1) the arrival of an incoming message (2) the allocation of local resources, and (3) the forwarding of an outgoing message. Each of these events may require different decisions to be complete. The content of a COPS request/decision message depends on the context. A fourth type of event is useful for types of clients that wish to receive configuration information from the PDP. This allows a PEP to issue a configuration request for a specific named device or module that requires configuration information to be installed.

There are two mechanisms by which resources may be allocated: configured (or provisioned, or pre-defined, or pro-active) and signaled (or on-demand, or reactive). Each solution has their strengths and weaknesses. With configured mechanisms, traffic treatment (such as classification, priority, shaping, etc.) can be specified as well as the characteristics of the traffic to receive that treatment. An administrator would observe traffic patterns on the network, compare that with the desired state (based on business or operational needs), and then choose policies that allocate resources accordingly. Such mechanisms may work quite well for traffic such as HTTP, telnet, or FTP, which are tolerant to the variance in flow quality (jitter, packet reordering, etc.).

The outsourcing model is totally different. A policy enforcement device issues a request to ask for a decision for a specific request coming from a user, a program or a process. For example, the arrival of an RSVP message to a PEP requires a fast policy decision to avoid a long delay for an end-to-end set-up. The PEP may use COPS-RSVP to send a request to the PDP, soliciting for a policy decision.

Note that the outsourcing policy scheme differs with configuring policy scheme, but they are not mutually exclusive and operational systems may combine both.

The strength of signaling is that it enables the network to offer QoS guarantees, and to simultaneously be used efficiently. Without signaling, it is necessary either to compromise the quality of the guarantees, or to overprovision the network. In some networks, over provisioning may be a viable option. However, in other networks it may not. If the network manager wants to have the flexibility to not overprovision the network then, an end to end signaling must be available to be used for policy-based admission control decisions.

Signaling mechanisms can provide information beyond the QoS needs to handle the traffic. User information and application identification that could be hidden by IPsec can be provided, thus allowing higher quality information on the traffic.

One of the most difficult parts of PBN concerns policy translations: transformation of a policy from a representation or from a level of abstraction, to another representation or level of abstraction. For example, it may be necessary to convert a PIB data (Policy Information Base, e.g., a named data structure) to a command line format. In this conversion, the translation to the new representation is likely to require a change in the level of abstraction. Although these are logically distinct tasks, they are in most cases hidden in the acts of translating or converting or mapping. Therefore, policy conversion or policy mapping is an important problem that we do not look at in this paper.

## 3. EXAMPLES OF POLICY-BASED NETWORKING ARCHITECTURE

Policy-based systems may be used to manage and control different types of functionalities. In this section we will have a look at different examples where PBN may be applied. A first example concerns admission control schemes. These schemes are responsible for ensuring that the requested resources are available. Moreover, these schemes must take care of temporal constraints, identification and permission.

Policy-based admission control is able to express and enforce rules with temporal dependencies. For example, a group of users might be allowed to make reservations at certain levels only during off-peak hours. In addition, the policy-based admission control should also be able to support policies that take into account identity or credentials of users requesting a particular service or resource. For example, through a PBN scheme, an RSVP reservation request may be denied or accepted based on the credentials or identity supplied in the request.

A second example concerns Authentication, Authorization, Accounting (AAA) schemes. AAA deals with control, authentication, authorization and accounting of systems and environments. The schemes may be based on policies set by the administrator and users of the systems. The use of policy may be implicit or explicit. For example, a network access server can send dial-user credentials to an AAA server, and receives authentication that the user is who he claims, along with a set of attribute-value pairs authorizing various service features. Policy may be implied in both the authentication, which can be restricted by time of day, number of sessions, calling number, etc., and the attribute-values authorized.

A third example concerns quality of service (QoS). QoS refers to the ability to deliver network services according to the parameters specified in a Service Level Agreement. Quality of service is characterized by service availability, delay, jitter, throughput and packet loss ratio. At a network resource level, quality of service refers to a set of capabilities that allow a service provider to prioritize traffic, control bandwidth, and network latency. There are two different approaches to the quality of service on IP networks: Integrated Services (IntServ), and Differentiated Service DiffServ. IntServ approaches require policy control over the creation of signaled reservations, which provide specific quantitative end-to-end behavior for a flow. In contrast, DiffServ approaches require policy to define the correspondence between codepoints in the IP packet DS-field and individual per-hop behaviors to achieve a specified per-domain behavior. A maximum of 64 per-hop behaviors limit the number of classes of service traffic that can be marked at any point in a domain. These classes of service signal the treatment of the packets with respect to various QoS aspects such as flow priority and packet drop precedence. In addition, policy can be used to specify the forwarding of packets based on various classification criteria. The policy controls the set of configuration parameters and forwarding for each class in DiffServ, and the admission conditions for reservations in IntServ.

Another example is provided with VPN configuration. Let us first recall VPN meaning since the term has been widely used with a great deal of confusion. VPN is a set of terminal equipments that can communicate with each other. More formally, a VPN is defined by a set of administrative policies that control connectivity, quality of service, security, etc. among terminal equipment. A classical use of VPNs is security. Rather than impose the network manager to set security mechanisms for individual terminal equipment, policy-based management system can consolidate and synchronize access control lists and related policy information to promote a consistent security policy across the enterprise. For example, the network manager can use a policy-based management system to set policies for selection of tunneling protocols and to update client configuration instead of configuring each security device and each terminal equipment, making the management of the VPN system more scalable.

A last example of the use of PBN architecture on traffic engineering may be considered [10]. An IP traffic engineering policy could be applied for provisioning or allocating resources of an IP network. The allocation should be in correspondence with the quality of service negotiated by the terminal equipment. The use of COPS protocol to dynamically enforce traffic engineering policies yields the generic model shown in Figure 2.

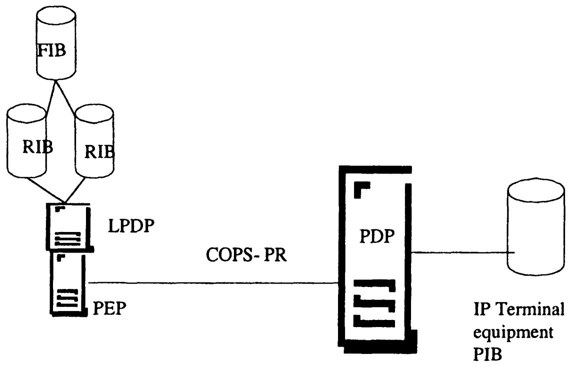

*Figure 2* - A model of an IP traffic engineering policy enforcement scheme

As shown in Figure 2, the edge router is a policy enforcement point, which supports the IP Traffic Engineering (TE) client-type. The IP TE client-type is specified by the PEP to the PDP, and is unique for the area covered by the IP traffic engineering policy, so that the PEP can treat all the COPS client-types it supports as non-overlapping and independent namespaces. The router contains also a LPDP, which can store the routing processes that have been activated in the router. Within the context of enforcing an IP traffic engineering policy, the LPDP is expected to calculate and install the IP TE routes that comply with the QoS requirements expressed in the IP TE-related information that has been received by the PEP. The router has to contain instances of Routing Information Bases (RIB), according to the different routing processes that have been activated. No specific assumption was made about the actual number of RIB instances that can be supported by the router, since this is an implementation specific issue. The PEP has also to contain a Forwarding Information Base (FIB), which will store the routes that have been selected by the routing processes.

The enforcement of an IP traffic engineering policy is based upon the use of an IP TE policy server, the PDP that sends IP TE-related information to the PEP capability contained in the router. The IP TE-related information is stored and maintained in the IP TE Policy Information Base, which will be accessed by the PDP to retrieve and update the IP TE-related information whenever necessary.

The IP TE-related information is conveyed between the PDP and the PEP thanks to the establishment of a COPS-PR connection between these two entities. The COPS-PR protocol provides a named data structure (the PIB), to identify the type and purpose of the policy information that is sent by the PDP to the PEP for the provisioning of a given policy.

As in COPS-PR, IP traffic engineering policy information is described as a named data structure, a PIB. Here, the data structure is described as a collection of PRovisioning Classes. Furthermore, these classes contain attributes that actually describe the IP TE-related information that will be sent by the PDP to the PEPs. These attributes consist of the link and traffic engineering metrics that will be manipulated by the routing processes being activated in the routers to calculate the IP TE routes for a given destination, among other characteristics.

This approach clearly assumes that each service provider will have the ability to instantiate the contents of its own IP TE PIB, according to the routing policies that have been defined for forwarding the traffic within its domain, but also outside of its domain.

## 4. NEW EXTENSIONS TO POLICY-BASED ARCHITECTURE

While the focus of many early systems for policy-based networking has been the control of edge devices such as edge routers, firewalls, or gateways, future systems should have to account for end-user hosts as policy enforcement points. In fact, it is necessary to look at these terminal equipments as PEPs, both to provide finer-grained classification of traffic and to deal with traffic classification problems that can arise when traffic from the user terminal is encrypted. Problems with network congestion and QoS adaptation will be solved by enforcing policies at the terminal equipment, requiring this terminal to be well aware with regards to the network traffic it generates. We believe that in the future the enforcement points could not be edge routers except complicating the way to enforce the policy on these machines.

So, we think that COPS is able to take place in this new architecture as a homogenization element to take care directly of the terminal equipment in its quality of service, its mobility, its security, etc. We can define a new client type that would permit to interface directly the customer with the PDP.

To get a direct negotiation with the PDP, we have proposed an Internet draft [11] using COPS protocol for supporting SLS (Service Level Specification) negotiation. COPS-SLS is an extension of COPS protocol. The advantage when using COPS for SLS negotiation is the inherent flexible characteristic of COPS protocol. COPS may support multiple client-types. So, COPS-SLS protocol needs only to specify corresponding new objects used in this client-type (COPS-SLS). The client-handle object defined by the COPS protocol gives a mechanism for handling various requests in a single PEP. This capability will be used to handle several SLS negotiations from a single PEP.

The PEP in COPS-SLS is just a logical entity which requests network resources for itself or possibly on behalf of other entities. So, the client may be an end-host, or a gateway of a local network or another ISP. The model we have implemented is illustrated in Figure 3.

To negotiate a level of service, COPS-SLS has two phases: Configuration phase and Negotiation phase. The communication starts with the Configuration phase. The PDP uses the Configuration model to configure the Negotiation phase. After that, in the Negotiation phase, the client use the Outsourcing model to request a level of service with parameters conforming to the configuration installed in the Configuration phase. This organization in two phases makes the SLS negotiation dynamic. At any time, when the network sends a new configuration to the client, the Negotiation process will apply these policies in subsequent service level requests.

To negotiate a level of service, the client sends a request indicating its desired service level under the form of instances of PIB classes. Using PIB to represent SLS information makes COPS-SLS flexible and adapted to desired negotiation parameters of network providers. COPS-SLS protocol is designed to permit basic activities in SLS negotiation. The client can request, modify or terminate a level of service. The network can accept or reject a service level request, propose another service level to the client or degrade a service level when necessary.

With COPS-SLS protocol, it is easy for a company to install an Intranet with a policy-based control. This policy-based control may allow some applications to get a good quality of service and some others to be delayed. Packets of these applications may be given a very low level of priority within the company or even may be discarded as a non-appropriate traffic. This solution could be used as a basis for a security system using a firewall.

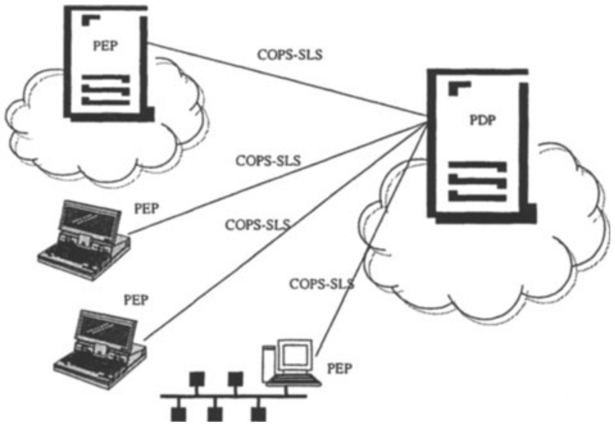

*Figure 3* – The COPS-SLS model

COPS-SLS is suitable for the negotiation of SLS between network providers. A domain may negotiate with another domain to obtain a level of service for inter-domain communications. For example, a DiffServ domain may request another DiffServ domain to guarantee a level of service for all packets having a specific DSCP.

Another application of the policy-based architecture we developed concerns the management of the mobility of the terminal or the mobility of the user. COPS extension for Mobile IP policy registration control was proposed in [12]. This proposal deals with terminal mobility management. We introduce a policy-based architecture to support mobile user and mobile terminal registration, service portability, and QoS negotiation in fixed and wireless network access. The first challenge of this work is to define a policy based architecture to support user and terminal registration to achieve location management. The second challenge is to define a policy based architecture to support fixed or mobile service portability and QoS negotiation. To achieve these challenging goals, we introduce new components in the IETF policy-based architecture (e.g., Figure 4) and we introduce two COPS extensions called COPS-MU (Mobile User) and COPS-MT (Mobile Terminal), which define new policy objects to support user and terminal registration, service portability, and QoS negotiation.

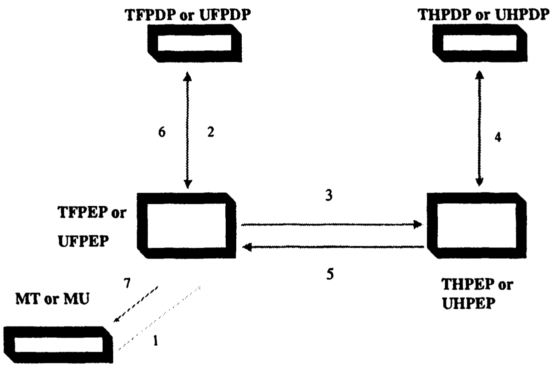

*Figure 4* – The COPS-MU model

Figure 4 describes the new components used in the COPS-MU or COPS-MT architecture related to mobile terminal and mobile user. Some mobile IP terms used in this figure are explained below:

- ***TFPDP/UFPDP*** Terminal/User Foreign Policy Decision Point
- ***TFPEP/UFPEP*** Terminal/User Foreign Policy Enforcement Point
- ***THPDP/UHPDP*** Terminal/User Home Policy Decision Point
- ***THPDP/UHPDP*** Terminal/User Home Policy Enforcement Point.

The process represented by the sequence 1 to 7, in Figure 4, permits to treat registration, service portability, QoS assignment and mobility for terminal and user mobility [13].

Both techniques, COPS-SLS and COPS-MU, may be combined to avoid T/UFPEPs and T/UHPDPs devices in COPS-MU.

Associated with these policies, it is possible to address allocation schemes using DHCP protocol. Policies can dictate how sets of IP addresses are to be allocated and for what duration. It is also possible to address routing policies, VPN policies and many others IP protocols management schemes. Some extensions have been provided in [14].

## 5. CONCLUSION

This paper presented an introduction to policy-based management within an IP network. It also looked at different proposals to use efficiently policy-based management techniques. This management is related to translating high-level user needs to device-specific configuration. An important problem concerns the dynamic of the system. Adaptive policies could handle the changes in the network. Rules reflecting a change can be placed in the policy repository. The change may happen from a threshold or simply from the time of day. Agents on PEPs may indicate these changes to the PDP. The PDP decides about the set of policies that need to be applied. PDP may also choose the policy that satisfies the requirement of a high-level policy determined by a business needs. Another way to consider the problem is to support a policy discovery system that should determine the best policy to apply to a user requesting for a transmission. For example, the policy discovery system may decide to use IPsec if the flow has to traverse the Internet.

The policy is applied on an administrative domain and another challenge concerns interdomain policies. It would be interesting to see how the notion of policies applies across administrative domains. The use of COPS protocol is an available solution to correlate the policies to be chosen when a flow has to cross several administrative domains. An agent negotiation is another possibility to settle the policies to be applied.

Finally, policy monitoring has also to be determined to verify that the network is meeting the desired business needs. The monitoring system checks that the implementation of the policy complies with what was expected by the PDP. Indeed, the high-level policies reflect the SLAs, and the monitoring system must confirm that these SLAs are performed.

As a conclusion, we think that the most important function of policy-based networking systems is to simplify network management and operations in complex networks. These systems provide QoS, security, mobility and much more functions within a homogeneous way.

## References

[1] RFC 2748 – D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, The COPS (Common Open Policy Service) Protocol, January 2000.

[2] RFC 2749 – J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, and A. Sastry, COPS usage for RSVP, January 2000.

[3] RFC 2750 – S. Herzog, RSVP Extensions for Policy Control, January 2000.

[4] RFC 2751 – S. Herzog, Signaled Preemption Priority Policy Element, January 2000.

[5] RFC 2752 – S. Yadav, R. Yavatkar, R. Pabbati, P. Ford, T. Moore, S. Herzog, Identity Representation for RSVP, January 2000.

[6] RFC 2753 – R. Yavatkar, D. Pendarakis, R. Guerin, A Framework for Policy-based Admission Control, January 2000.

[7] RFC 2872 – Y. Bernet et R. Pabbati, Application and Sub Application Identity Policy Element for Use with RSVP, June 2000.

[8] RFC 2940 – A. Smith, D. Partain, J. Seligson, Definitions of Managed Objects for Common Open Policy Service (COPS) Protocol Clients, October 2000.

[9] RFC 3084 – K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith, COPS Usage for Policy Provisioning, March 2001.

[10] C. Jacquenet, A COPS client-type for IP traffic engineering, Internet draft, http://www.ietf.org/internet-drafts/ draft-jacquenet-ip-te-cops-02.txt, June 2001.

[11] T.M.T. Nguyen, N. Boukhatem, Y. El Mghazli, N. Charton, G. Pujolle, COPS Usage for SLS negotiation (COPS-SLS), Internet Draft, <draft-nguyen-rap-cops-sls-02.txt>, April 2002.

[12] M. Jaseemuddin, A. Lakas, COPS usage for Mobile IP, Internet draft, draft-jaseem-rap-cops-mip-00.txt, October 2000.

[13] H. Chaouchi, G. Pujolle, COPS-MU: Policy based user mobility management, Proceeding IEEE Conference on Applications and Services In the Wireless Public Infrastructure, Evry, France, July 2001.

[14] 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.