POLICY-BASED SERVICE LEVEL AGREEMENT MANAGEMENT SYSTEM

Noh-sam Park, Shin-kyung Lee, Gil-haeng Lee
Electronics and Telecommunications Research Institute

Keywords: Service Level Agreement, SLA, SLM, Policy

Abstract: SLA is a negotiated agreement between a customer and the service provider on levels of service characteristics and the associated set of metrics. In this paper, we propose a policy-based SLA management system. We present an approach to react not only when an SLA is violated, but also before imminent SLA violations. We provide a common generic framework capable of components to interwork via XML. The managed SLA metrics are classified into service opening metrics, trouble metrics, and performance metrics. We rely on a proposal for architecture to provide the end-user with SLM from the service subscription to the service termination. Finally, we'll give an example to illustrate a typical scenario to assure customers' SLAs in ADSL network service.

1 INTRODUCTION

A Service Level Agreement (SLA) is part of the contract between the service provider and its consumers. It describes the provider's commitments and specifies penalties if those commitments are not met.

Service Level Agreements (SLAs) are fundamental to business continuity. The bottom line is that they define your minimum levels of availability from key suppliers, and often determine what actions will be taken in the event of serious disruption. As a consequence, they require full consideration and attention and must be constructed extremely carefully. This is not an area in which to cut corners.

The SLA will be wide in scope, covering all key aspects of the service. Typically, it will fully embrace such issues as problem management, compensation (often essential in terms of motivation), warranties and remedies, resolution of disputes and legal compliance. It essentially frames the relationship, and determines the major responsibilities, both in times of normal operation and during an emergency situation. The difficulty, as ever, is usually where to start. Is it possible with a blank piece of paper? This is not usually a good idea, not specifically because of the amount of effort involved, but also because of the greater risk of missing, or perhaps not properly documenting, a major issue.

It is now widely accepted that service provision and receipt should be governed by an agreement. This is essential to define the parameters of the service, for the benefit of both the provider and the recipient. It must obviously cover many other issues, as well as defining the service itself.

SLA does not care how the service is configured or what is the topology of underlying network(Bao Hua Liu et al., 2003). SLA only concerns end-to-end delivery of the services. There has been substantial progress in the management of distributed application. With distributed applications running on underlying networks and systems, the performance of applications is inevitably influenced by the performance of networks and systems.

Park N., Lee S. and Lee G. (2004).
POLICY-BASED SERVICE LEVEL AGREEMENT MANAGEMENT SYSTEM.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 111-117
DOI: 10.5220/0001390801110117

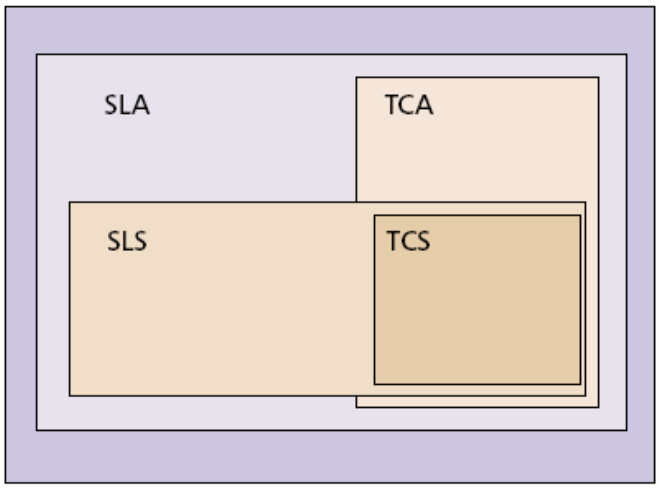

Figure 1: Interrelation between SLA, SLS, TCA, and TCS

According to (Brian et al., 2002), all of the application level performance problems, 45% are caused by network problems. Moreover, the customer wants the higher level of network speed as the contents of multimedia proliferate. Therefore, the need arises for an SLA management process in IP networks. The aims of this work are the definition of an architecture for the implementation of SLAs, and the design of an entity capable to monitor a violation of SLAs.

The remainder of this paper is structured as follows. Section 2 will discuss the Quality of Service (QoS) terminology, and describe the earlier works of service level management system. In section 3, we will explain a framework of our SLA Management system, together with a detailed description of the components. A sample scenario, illustrating the SLA management process over IP networks, will be provided in section 4. In section 5, we'll summarize our work and sum up the conclusions from this study.

2 SERVICE LEVEL AGREEMENT

Several approaches to QoS definition, including those of IETF, ITU, and ETSI are in progress in order to clarify the terminology and eliminate the confusion. In this section, we describe the terminology related to an SLA, and review the related works.

2.1 QoS Terminology

The ITU defines a service level agreement (SLA) as "*a negotiated agreement between a customer and the service provider on levels of service characteristics and the associated set of metrics. The content of SLA varies depending on the service offering and includes the attributes required for the negotiated agreement*" (ITU-T Rec, 2001). An SLA may be in form of a document containing names of the parties signing the contract. It should be composed of service level objectives, service monitoring components, and financial compensation components. Service level objectives encompass QoS parameters or class of the service provided, service availability and reliability, authentication issues, the SLA expiry date, and so on. Service monitoring specifies the way of measuring service quality and other parameters used to assess whether the service complies with the SLA. It may also include an agreement on form and frequency of delivering the report on service usage. The financial component may include billing options, penalties for breaking the contract, and so forth (ITU-T Rec, 2001).

The notion of service level specification (SLS) was introduced to separate a technical part of the contract from SLA. It is defined as "*a set of parameters and their values which together define the service offered to a traffic*" (Grossman, 2002). It specifies a set of values of network parameters related to a particular service. The IP transport services are technically described by SLSs.

A traffic conditioning agreement (TCA) is an agreement specifying packet classification rules and traffic profiles as a description of the temporal properties of a traffic stream, such as the rate and burst size. In order to force a customer's traffic conformance to the profile particular metering, marking, discarding, and shaping rules are defined. The treatment of out-of-profile packets is also specified by a TCA. According to the IETF definition, "*TCA encompasses all of the traffic conditioning rules explicitly specified within a SLA along with all of the rules implicit from the relevant service requirements and/or from a DiffServ domain's service provisioning policy*" (Blake et al., 1998).

The traffic conditioning specification (TCS) is a set of parameters with assigned values that unambiguously specify a set of classifier rules and a traffic profile. A TCS is a technical part of TCA. A TCS is also an integral element of an SLS (Grossman, 2002).

Interrelations between SLA, SLS, TCA and TCS are shown in Fig. 1(Gozdecki et al., 2003).

2.2 Related Works

The importance of SLA has been recognized and widely accepted by ASP's, ISP's, etc. This section reviews features of various SLA management systems.

Reference (Leff et al., 2003) examines the requirements on a grid's infrastructure to support SLAs and describes a prototype implementation that satisfies them. It specifically focuses on the dynamic offload infrastructure needed to meet SLAs related to varying workload conditions. The components has the ability to formally define an SLA, detect an SLA violation, scale up resources dynamically in response to an SLA violation. However, it is limited to react only when an SLA is violated, not predict SLA violations.

In (Chakravorty et al., 2003), an architecture for end-to-end QoS control in a wired-wireless (UMTS) environment is proposed with dynamic SLA-based resource provisioning. It is achieved in CUE (CADENUS-UMTS Extension) framework. CUE architecture adds two new components, CUE-SM and CUE-RM, that can be used to provision end-to-end QoS in a wired-wireless network. It uses a combined mix of dynamic SLA-based and policy control schemes. The main functions include automation of r-SLA (retail SLA), static or dynamic negotiation of r-SLAs. Adopting QoS negotiation, it is possible to make a decision about user QoS in real time.

In a view of contract management, T.J. Watson Research Center has developed SAM(Buco et al., 2003). The e-business SLA contract execution manager SAM enables the provider to application provider to deploy an effective means of capturing and managing contractual SLA data as well as provider-facing non-contractual SLM data. SAM assists service personnel to prioritize the processing of action-demanding quality management alerts. And it automates the prioritization and execution management of approved SLM processes on behalf of the provider.

In order to share management information across interdomains, (Bhoj et al., 2001) elaborated a web-based architecture. The architecture can be used for automatically management of SLA for internet services. The authors also demonstrated how a service provider could offer verifiable and meaningful pre-defined SLA behaviors to their customers.

3 FRAMEWORK OF SLMS

We propose a form of architecture for policy-based SLA Management System (SLMS) using web service. It provides a common generic framework capable of its components to interwork via XML. We design the user interface for system operators using SLMS. Operators use can search SLA metrics, violation details, and can monitor SLA in real time.

3.1 SLMS Components

We categorize the SLA metrics into service opening metrics, trouble metrics, and performance metrics. Service opening means that end-user must be able to use the network service at the date of agreements. Trouble metrics includes the trouble recovery time, the sum of trouble time, and the number of troubles. Performance metrics are related to the QoS of network such as packet delay, packet loss.

In order to efficiently manage the SLA, the warning messages are sent to system operators in real time while monitoring the SLA. System operators check the details and take an action to prevent the violation of SLA.

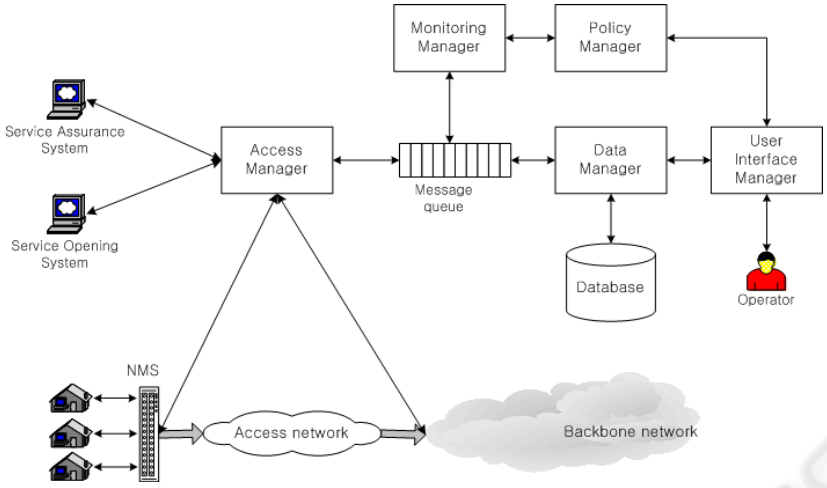

Figure 2: Architecture of SLMS

In this context, we rely on a proposal for architecture to provide the end-user with service level management (SLM) based on policy. The functional blocks are:

- AM - Access Manager
- DM – Data Manager
- MM – Monitoring Manager
- PM – Policy Manager
- UM – User Interface Manager

The AM is the entity that receives the information related service opening, trouble and performance. It is responsible for translating the information into XML format, pushing the translated XML document into the message queue. Furthermore, the AM collects the network performance data.

The DM reads the XML data from the message queue, classifies the data according to the SLA metrics. As the DM manages the information in the database, it can response to the UM the retrieve and save the SLA related data.

The MM plays the important role of monitoring the violation of SLA metrics. It reads the monitored data through the DM at the defined interval, and compares the current data with SLA metric. If the MM detects the violation, it sends the violation information through the message queue.

The PM is closely related to MM. Policy is the editable file which contains attribute-value pairs. Policy contains the flag if the metrics is monitored, and the monitoring interval. The MM reads the policy file and parses attribute-value pairs. Depending on the value, we can monitor the specific metrics, or not.

Operators manage the SLA of end-users by utilizing the SLMS. The UM interacts with the operators, and provides the variety of data. Additionally using the UM, SLA metrics can be retrieved, updated, and added according to the change of the network service. Operators can configure the policy of SLM such as the execution of monitoring or not. Fig. 2 represents the architecture of our SLMS.

3.2 Monitoring SLA Metrics

Monitoring is the core function of SLM to prevent the violation of SLA. Our system has the monitoring component which checks the threshold at first, and compares the metrics value secondly. The threshold is the value which can be alerted to the operator by sending the ‘warning’ message. If the operator receives the message from the system, he/she checks the details, and can take an action to prevent the violation of SLA.

As the aforementioned metrics classification, the MM monitors the following metrics : service opening, trouble, and performance. At the system initiation stage, the MM reads the policy file and parses the attribute-value pairs. For example, if ‘packet delay’ metrics of ADSL service is marked as ‘not monitored’, MM will not monitor the corresponding data. As deciding the policy, the MM creates threads in order to monitor the categorized metrics.

Table 1:. SLA Metrics

MetricsCode	Description	Threshold	Value	Unit
ADS10	ADSL Service Opening	2	8	Day
ADS20	ADSL Trouble Delay Time	120	180	Minute
ADS21	ADSL Monthly Trouble Time	20	24	Hour
ADS31	Packet Loss Rate	3	5	%

Using the UM, the operator can configure whether the metrics are monitored or not. If the metrics is set not to be monitored, the MM will not execute the monitoring function. But if the history of warning and violation is recorded in the database, and can be retrieved by the UM.

Also, another policy-based monitoring can be accomplished by configuring the various preferences. The interval of monitoring can be changed by using UM. If the operator changes the interval, UM sends the message to message queue. While listening to the message queue, MM receives the event of interval change. MM aborts the current living threads, and invokes threads again with the new interval.

At the defined thread invoke time, the MM periodically creates threads. Threads retrieve the monitored data, threshold and metric. Firstly, the MM thread compares the data with the threshold. If the current value is greater than the threshold, the 'warning' message is sent to the AM in XML format.

As time passes, the MM thread will detect the violation of SLA by comparing the current value with value of the metrics. If the violation event occurs, the 'violation' message will be sent to the message queue, and at the same time the violation details are recorded in the database by the DM. The threads are disposed after execution. This procedure is iterated on processing time.

4 POLICY-BASED SLA MANAGEMENT

We manage SLA metrics as the code with value and threshold. If the other metrics should be added or deleted, we can simply manage the metrics only to add/delete the related metrics code into/from the database(Table 1).

In this section, we will explain the SLA management over xDSL services by illustrating a sample scenario. From the service subscription, our system will monitor in order to meet the SLA. And trouble recovery and network performance must be satisfied in order not to violate the metrics value.

4.1 Service Opening Management

Service must be available before the customer's hoping date. We assume that a customer would like to subscribe an ADSL service, and wants to use the service in 10 days from the requesting date. Furthermore, the customer wants a high quality of service. The request is received by service-opening system, and that system passes the required data to the our system.

As soon as SLMS receive the subscription data, the monitoring process begins. The threshold of service open metric is 2 days before the user's hoping date. So, no event is sent to the system operator from SLMS in 8 days from the requested date. During that time, the operator can ask of service-opening system to check the current opening state.

Service opening can be done successfully in 10 days. The service-opening system passes the result with the service quality (e.g. line speed). If the service quality does not satisfy the SLA, the system operator requests an order again. So, the quality of service is guaranteed.

If the service is not applicable to the customer after 8 days, the warning message is sent to the system operator. The system operator can send a command to the service-opening system in order not to violate the service open metric.

Although SLMS alerts with the warning message, it is possible not to accomplish the order. In that case, our system shows the violation message(Fig. 3). More time passes, more money must be refunded to the customer. So the system operator needs to hurry the service opening process.

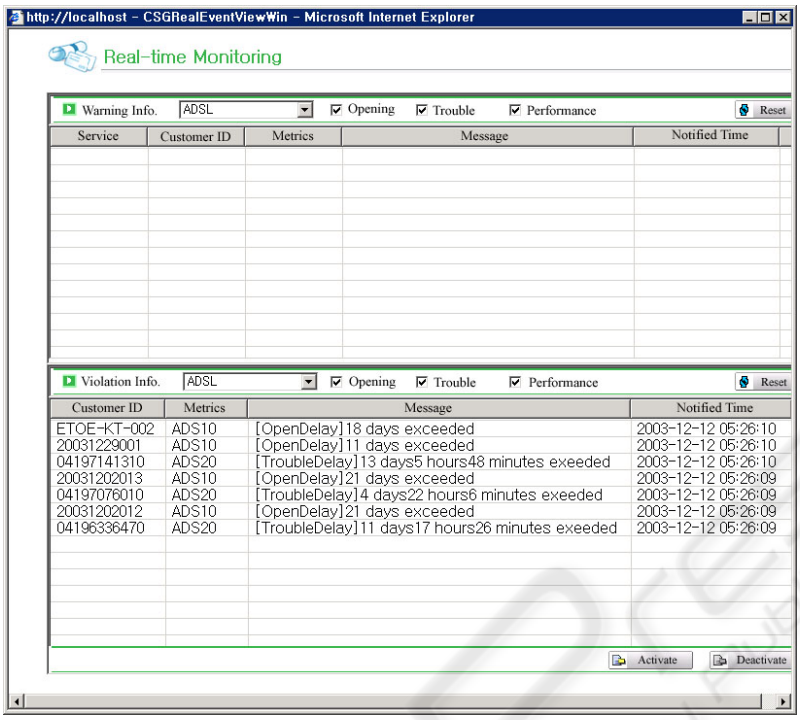

Figure 3: A sample GUI for real-time monitoring

4.2 Service Trouble Management

During the service time, the user can not use the service due to the network provider's responsibility. For example, it includes the periodic network examination, system breakdown, and so on. The customer is assured to be able to use the service in the specified period. If not so, the customer receives the money in proportion to the exceeded time.

The customer makes a report to notify the network trouble if he/she cannot use the network service. The customer wants to use the service in the service recovery time. The received trouble report is received by service-assurance system, and that system passes it to SLMS. In the same way of opening monitoring, SLMS monitors not to violate the metric : service recovery time.

Notwithstanding the user may be ignorant of the service outage, it is possible of the network provider to detect it. The same process is executed in case of automatically detected trouble, but not overlapped. Whether the customer knows the trouble or not, SLMS assures the service recovery time.

Additionally, we manage the metrics related trouble in the specified period : the sum of trouble time, the number of troubles. As individual troubles are recorded into database, we manage these metrics easily to add the time and count. Our system assures that the total trouble time must not be exceeded to 24 hours in a month; the count of troubles must be less than 5 in a month.

4.3 Service Performance Management

If the network provider does not satisfy the quality of network, the customer is disappointed and may find the other provider. In that point, the service performance is the most important thing to both customer and provider. We have the functionality of managing the following metrics : packet loss rate, packet delay, availability.

SLAs can be classified in retail-SLA and wholesale-SLA(D'Arienzo et al., 2003). The retail SLA refers to the agreements between an end-user and a service provider. Conversely, a wholesale-SLA is an agreement between network operators, and takes into account traffic aggregates flowing from one domain to another. As we rely on retail-SLA, the managed section of network performance is limited between the end-user and the backbone network. Now we make a research to solve this

limitation by using the user-side agent which collects the network performance information.

SLMS collects the performance data by polling the equipment from the centralized server. The server is located in the backbone network, and a number of end-users are attached to the target equipment. As ADSL service uses dynamic host configuration protocol (DHCP), it is impossible to collect the performance data using the fixed IP. So, the same performance data of equipment are applied to the attached end-users.

In contrast to other metrics, the violation of service performance is made with the average value within the specified period. Temporarily the service performance may be declined, and the warning event may be sent. If the system operator takes an action to prevent a violation, it is burden because a violation may not be happened. So we provide the trend of network performance to the system operator. Seeing the trend, the operator determines if it requires an action or not.

5 CONCLUSIONS

We propose a form of architecture for policy-based SLA Management System. We first describe the QoS terminology, including SLS, SLA. And policy-based SLMS are introduced with detailed description of its components. According to the policy, we can configure the metrics. Our system has the capability to manage the SLA from the service subscription to the service termination. It is possible to monitor in real-time in order not to violate the metrics. We design the user interface for system operators using SLMS.

As we rely on retail-SLA, the managed section of network performance is limited between the end-user and the backbone network. Future work has been working by using the user-side agent which collects the network performance information. According to the implications of the research, future work has been conducted to interwork Operation Supporting Systems (OSSs) such as the refund system and NMS.

REFERENCES

Bao Hua Liu, P. Ray, S. Jha, *Mapping distributed application SLA to network QoS parameters, Telecommunications*, 2003 (ICT 2003), pp.1230-1235, 2003

Brian L. Tierney, *End-to-End Application Monitoring using the Distributed Monitoring Framework*, Lawrence Berkeley National Laboratory, 2002

ITU-T Rec. Y.1241, *Support of IP-based Services Using IP Transfer Capabilities*, Mar. 2001

D. Grossman, *New Terminology and Clarifications for Diffserv*, IETF RFC 3260, Apr. 2002

S. Blake et al., *An Architecture for Differentiated Services*, IETF RFC 2475, Dec. 1998

J. Gozdecki, A. Jajszczyk, R. Stankiewicz, *Quality of service terminology in IP networks*, Communications Magazine, IEEE, pp.153-159, 2003

A. Leff, J.T. Rayfield, D.M. Dias, *Service-level agreements and commercial grids*, Internet Computing, IEEE, pp.44-50, 2003

R. Chakravorty, I. Pratt, J. Crowcroft, *A framework for dynamic SLA-based QoS control for UMTS*, Wireless Communications, IEEE, pp.30-37, Oct, 2003

M. Buco, Rong Chang, L. Luan, C. Ward, J. Wolf, P. Yu, *Managing eBusiness on demand SLA contracts in business terms using the cross-SLA execution manager SAM*, ISADS, pp.157-164, Apr. 2003

P. Bhoj, S. Singhal, S. Chutani, *SLA Management in Federated Environments*, 5-24 Comp. Nets., vol. 35, no.1, Jan. 2001

M. D'Arienzo, M. Esposito, S.P. Romano, G. Ventre, *Automatic SLA Management in SLA-aware architecture*, Telecommunications, 2003 (ICT 2003), pp.1402-1406, 2003

G. Cortese, R. Fiutem, P. Cremonese, S. D'antonio, M. Esposito, S.P. Romano, A. Diaconescu, *CADENUS: creation and deployment of end-user services in premium IP networks*, Communications Magazine, IEEE, pp.54-60, Volume: 41 , Issue: 1 , Jan. 2003