USING LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL FOR SERVICE LEVEL SPECIFICATIONS ADMINISTRATION

Valentin Hristov

South- West University- Blagoevgrad, Bulgaria
T.+359738889131; E. v_hristov@swu.bg

Abstract

A directory service is a simplified database. The Lightweight Directory Access Protocol is a distributed directory service protocol, and is based on a client-server model and runs over TCP/IP. The LDAP allows to configure networks for supporting different levels of services.

The purpose of present paper is to propose a modified schema for supporting Service Level Specifications based on LDAP directories, and study its performance under variety of access patterns.

1. INTRODUCTION

The Lightweight Directory Access Protocol (LDAP) was originally intended to be a lightweight alternative protocol for accessing X.500 directory services through the widespread TCP/IP protocol stack. This model of directory access was borrowed from the DIXIE and Directory Assistance Service protocols.

LDAP is an application protocol for querying and modifying directory services running over TCP/IP [1]. A LDAP directory is a set of objects with similar attributes organised in a logical and hierarchical manner, i.e. the directory is a tree of directory entries. Each entry has a unique identifier, i.e. its Distinguished Name (DN). This consists of its Relative Distinguished Name (RDN) constructed from some attribute(s) in the entry, followed by the parent entry's DN. An attribute has a name (an attribute type or attribute description) and one or more values.

A client starts an LDAP session by connecting to an LDAP server. After that it sends an operation request to the server, and the server sends responses in turn. With some exceptions, the client need not wait for a response before sending the next request, and the server may send the responses in any order.

LDAP defines operations for querying and updating the directory. Operations are provided for adding and deleting an entry from the directory, changing an existing entry, and changing the name of an entry. Most of the time, LDAP is used to search for information in the directory.

The LDAP search operation allows some portion of the directory to be searched for entries that match some criteria specified by a search filter. Information can be requested from each entry that matches the criteria.

The current Internet operates on a best-effort basis, in which all packets are treated equally. Thus, the improvement of network service models with mechanisms to provide multiple service levels to users is actual problem. Researchers in the DiffServ community have proposed storing these policies in a central or distributed policy repository administered and accessed using a directory service such as LDAP [3], [4]. In this scenario, the policy repository is updated when the network provider negotiates new Service Level Specifications, or renegotiates existing contracts, and also when the policies need to reflect changes in network topology or traffic levels. Network elements frequently access the policy database, and download the current set of rules according to which customer traffic is served.

The purpose of present paper is to propose a modified schema for the administration of Service Level Specifications (SLS) based on LDAP directories, and study its performance under variety of access patterns.

2. SCHEMA FOR SUPPORTING SERVICE LEVEL SPECIFICATIONS

Recently, there has been much interest in network service models with mechanisms to provide multiple service levels to users. The two main approaches under discussion are the integrated service model, which supports quality of service (QoS) levels by allowing per-flow resource reservation

using RSVP signaling, and the differentiated service model, which provides multiple service classes which are served using different per-hop behaviors. In either model, the network provider negotiates a service level specification with a customer, defining aspects of network behavior such as the type of service user packets will receive, and the constraints the user traffic must adhere to. The Service Level Specification (SLS) may be dynamically renegotiated, based on changes in the customer requirements or network conditions. The network access points and internal routers implement the classification, resource control, and administrative policies associated with SLSs.

In fig. 1 is depicted Generic SLA Architecture [6]. The Domain Manager (DM) generally manages the network domain. It communicates with the policy server that administrates policies, rules and actions for different services stored in a policy repository. In addition, the network provider provisions the network in order to provide the service contracted to customers. The provisioning is physical (adding or removing network elements) and logical (partitioning or configuring network elements).

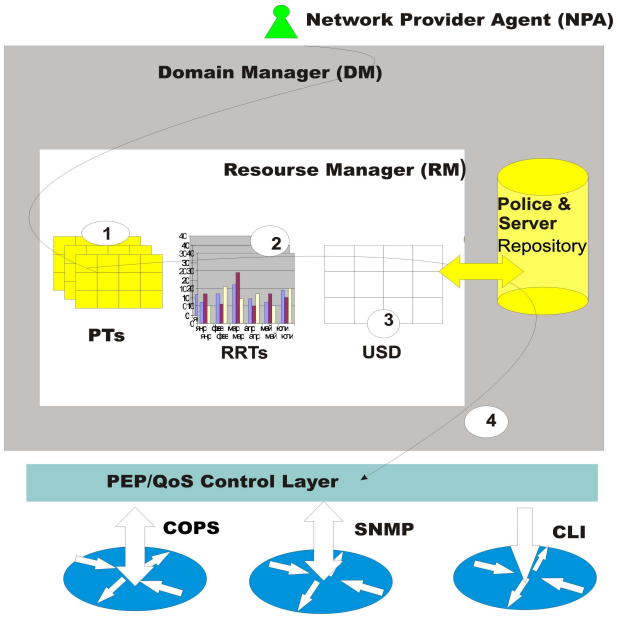

Fig. 1

The network configuration information may be maintained in LDAP directories, and downloaded periodically by routers. This allows the network provider to adjust configurations (for example, buffer space, or packet drop precedences) with a finer granularity in response to network usage feedback.

The architecture based on first approach provides immediate bandwidth reservation when capacity is available, as well as allows bandwidth resource to be reserved in advance (fig. 2).

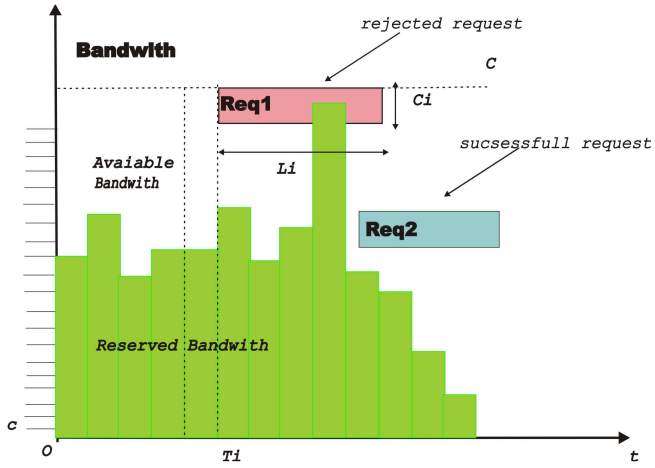

Fig. 2

A preliminary schema using LDAP for configuration of DiffServ networks has been proposed in [3]. The various aspects of a service, such as the traffic profile the user traffic must conform to in order to receive the service, and the forwarding rules for conforming traffic, are captured in a set of policies.

The *Architecture of Network QoS Control Using LDAP* consists of a management tool, a policy repository, a policy decision entity, and a policy enforcement entity. Fig. 3 shows the functional relations between these different entities. In the context of the service environment under consideration, the management tools are used by the network administrator to populate and maintain the LDAP directory with policies. Management tools may or may not reside on the same host as the directory server. Enforcement entities apply policy rules. A decision entity and enforcement entity are usually assumed to reside at each edge device, or network access point. The edge device is referred to by its location and would most likely be placed at the access point between a local subnet and the backbone network, or at the boundary between backbone networks of two service providers. At initialization, the edge device identifies its interface addresses. It determines the set of policies required for these interfaces, and downloads the corresponding classification policy rules from the LDAP server, as well as the service specifications referred by the policies. Subsequently, the edge device may poll the server periodically to learn of modifications to the directory, and download its set of policy rules if the directory is modified. If asynchronous mode operations are supported by the directory service, the downloading

of policy rules could also be triggered upon changes in the policy rules.

The decision entity downloads policy rules from the repository, through a LDAP client. The enforcement entity queries rules from the decision entity and carries out packet handling and monitoring functions.

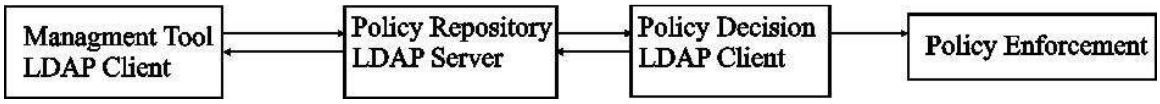

Fig. 3

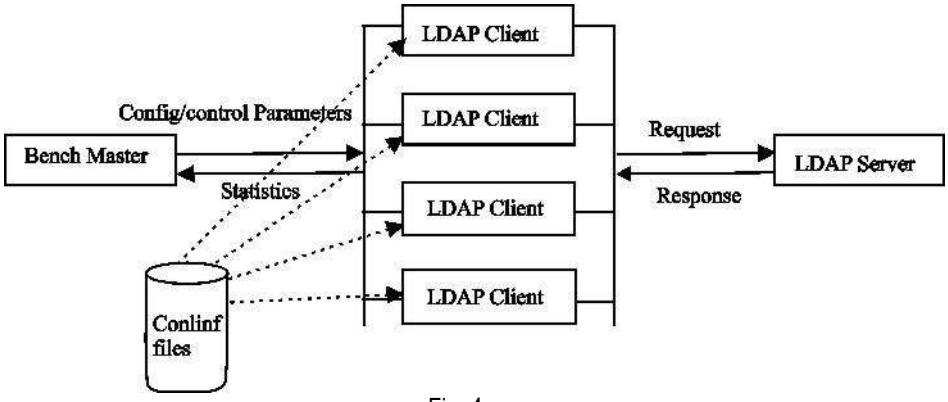

Fig. 4

A customer attaches to the network at one or more interfaces belonging to an edge device. Each interface is identified by an IP address. At each interface, one or more policies may be defined, and customer packets are monitored and processed according to these policies. Each policy is associated with a service level which defines actions on the part of network elements in handling customer packets. A policy may be applied on the basis of source/destination IP addresses, transport protocols, source/destination ports, and other parameters such as default port, URLs, etc.

Policy rules are stored in the LDAP directory as SLS PolicyRules objects (derived from the *Policy* class described in [3]). SLS PolicyRules objects may have attributes specifying the policy name, priority level of the rule, and the network interfaces to which the rule may be applied, as well as references to objects which specify the traffic profile, period of validity of the rule, type of RSVP service or DiffServ action, etc.

The directory structure of the LDAP directory used in proposed schema for supporting SLS is shown in Fig. 5.

Each *Customer* entry has a set of associated *Interface* entries. The *Policy* entry directly under the *Customer* specifies policy rules common to multiple interfaces belonging to the customer, while the *Policy* entry for each *Interface* specifies the policy rules specific to customer traffic at that Interface. In general, the *Policy* entry refers to one or more of the *Service* entries in the directory to specify the service to be received by the corresponding traffic.

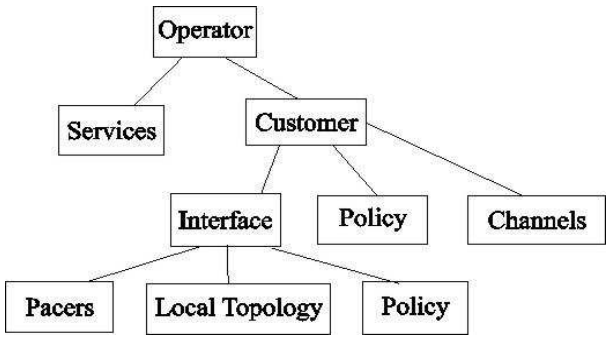

Fig. 5

The other entries shown in the LDAP directory include *Channel* and *Pacer* entries. A channel is a virtual pipe between an ingress edge device and an egress edge device. A pacer is the abstraction that limits the total amount of traffic that can be sent out into the backbone network at an access point.

The search filter for the search operation was constructed from the Interface address of interest, and the corresponding *Policy* object.

3. PERFORMANCE ANALYSIS OF SLS ADMINISTRATION SCHEME

Usually, the response delay at the LDAP server is obtained using the result of a nonpreemptive priority-based M/G/1 queue [2]. Thus, if we want to evaluate the response delay at the LDAP server, we only should consider the messages having higher priority than LDAP ones and ignore the lower priority messages, but these assumptions are not quite realistic in common case. The response delay at the LDAP server can be estimated more precisely by proposed bellow simulation model. The simulator is created with GPSS (General Purpose Simulation System) World Student version.

The simulation model is represented by following GPSS– block diagram (fig. 6).

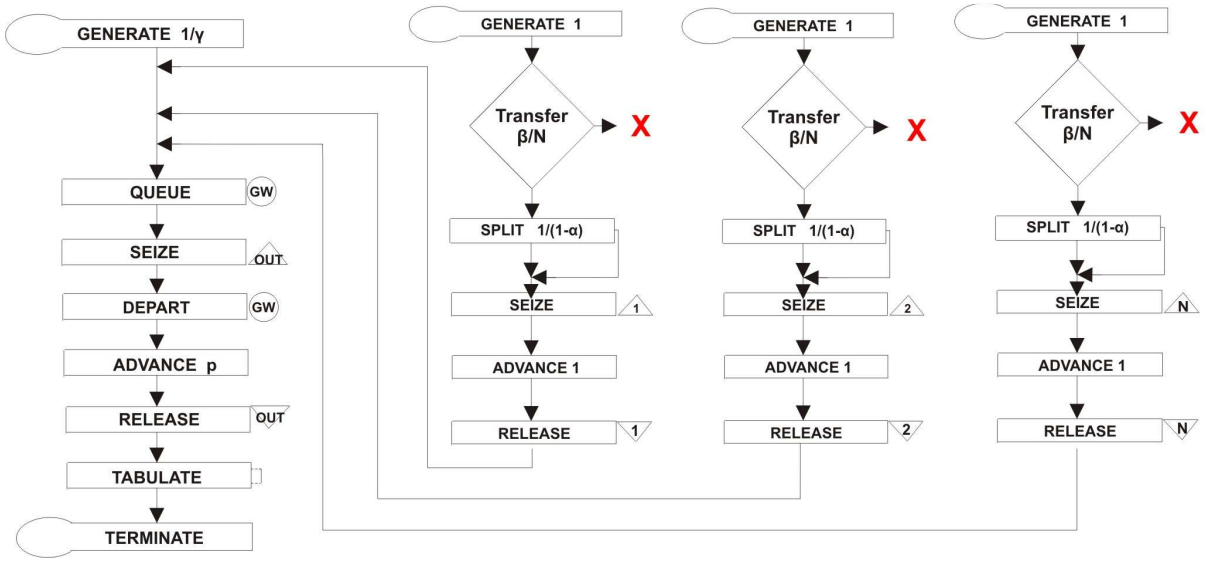

Fig. 6

There are N model segments which correspond to server processing of LDAP requests (in manner one thread one request) and one model segment which corresponds to the transmission (over communication channel) of LDAP as well as non- LDAP messages.

The number of LDAP clients (which generate requests) is N. Note that LDAP server can be multiprocessor system with M processors and each processor can start L threads, i.e. K=M.L. If N>K, the model segments which correspond to processing of LDAP requests would be realized with only K facilities, e. g. duplicating model segments- *1,2,…k* the necessary times.

The modeling process aims at getting the response delay for LDAP requests. The delay includes two components- search time and transmission delay (latency due transport over communication channel). The search time, or processing time due to bind (open the connection) and search in directory actually increases slightly at heaviest load. The time slot in this model is time that one thread processes one byte of the LDAP request.

The entry size for these simulations is random and realistic values in each data item, and the default directory size is 10 000 entries. We make the assumption that these entry sizes are geometrically distributed with mean value $1/(1-\alpha)$=490 bytes. The probability that the LDAP clients start a new request is denoted by β, therefore the probability that a client starts a new request is β/N. We assume β=0.008 and N=10.

The bandwidth of the communication channel (through Internet) is considered fixed, but only a fraction σ of it is available for the transmission of LDAP messages. The ratio between the transfer rate from LDAP server and the communication channel bandwidth is denoted by p. We assume p={1, 2, 3, 4, 5} and 10 Mbps channel.

The system load is:

$$\rho = \frac{\beta . p}{\sigma(1-\alpha)} \qquad (1)$$

Also, we assume ρ=0.4, 0.5, 0.65, and 0.98 and connection rate characteristics for the two type (LDAP as well as non- LDAP) messages. The load at the communication channel for non-LDAP message is generated by source (fig.6) with mean rate- γ, i.e:

$$\gamma = \left(\frac{1-\sigma}{\sigma}\right).\left(\frac{\beta}{1-\alpha}\right). \qquad (2)$$

Thus, the desired values of load- ρ correspond to the values of γ (The latter is an adjustable parameter).

Table1 shows the LDAP response delay, or latency versus load generated by LDAP and non-LDAP messages.

Table1

Delay,us	ρ=0.98	0,65	0,5	0,4
p=1	48608	1869,5	953,1	639,58
2	48119	1857,8	948,93	637,51
3	47630	1845,7	944,44	635,18
4	47140	1833,4	939,81	632,74
5	46650	1820,9	935,12	630,25

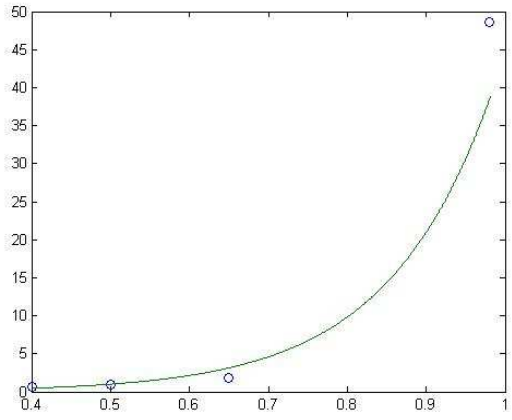

Fig. 7

Fig. 7 represents graphically first row of this table, as well as trendline for delay versus LDAP and non- LDAP load. Note, that the dimension of Y- axe is ms.As one can see the LDAP response delay increases with the system load, which is logical.

4. CONCLUSIONS

In this work, we propose a modified LDAP schema for supporting Service Level Specifications. This article also provides a study of response delay at the LDAP server, which is used to configure networks for supporting different levels of services.

In order to decrease LDAP response delay, or improve performance, the dual processor server could be deployed. The dual processor server shows similar performance at low loads, and the advantage increases to give roughly 40% smaller latency at higher loads for the total response time. The reduction in latency is observed mainly due to the reduction in the so-called connect time.

References

[1] Chieng D., Marshall A., Parr G. SLA Brokering and Bandwidth Reservation Negotiation Schemes for QoS-aware Internet // eTrans. on Network and Service Management. — 2005. — 1st Q. ,pp. 39—49.

[2] Fathi H., at al. Optimization of SIP Session Setup Delay for VoIP in 3G Wireless Networks,IEEE Transactions On Mobile Computing, Vol. 5, No. 9, September 2006, p. 1121

[3] Handbook on Quality of Service and Network Performance. ITU-T, 2003.

[4] Su H.K., Yau Z.Z., Wu C.S., Chen K.J. Session-Level and Network-Level SLA Structures and VoIP Service Policy over DiffServ-Based MPLS Networks // IEICE Trans. Commun. - 2006. - V. E89-B. - № 2, pp.. 383-391.

[5] Ефимушкин В.А., Ледовских Т.В. Бизнес-модели SLA-отношений // В сб.: Технологии информационного общества: Тезисы докл. Московской отраслевой научно-технической конференции, 23—25 апреля 2007 г. , М.: Инсвязьиздат, 2007, с.13.

[6] Ефимушкин В.А., Ледовских Т.В. Особенности управления SLA для обеспечения требований качества услуг в пакетных сетях, сп. Электросвяз No 10, 2008 г., с. 8- 11.