Proc. Int. Conf. on Dublin Core and Metadata for e-Communities 2002: 195-196
© Firenze University Press

Policy Control Network Architecture using Metadata

Ray S. Atarashi
Communications Research Laboratory
ray@crl.go.jp

Shigeru Miyake
Hitachi, Ltd.
yake@sdl.hitachi.co.jp

Fred Baker
Cisco Systems
fred@cisco.com

Abstract

Quality of service (QoS) technology has been implemented to be applied to new applications on the next-generation Internet. However, as new applications such as P2P and stream application have many kinds of features and requirements, some additional features should be added to current QoS control technology. Policy definition for transport layer in a domain and among domains is being discussed at IETF to set a standard process, however detailed policy corresponding to the application or contents information according to the application semantics has not been discussed. Therefore we developed QoS policy control mechanism using metadata which is defined as a structured data according to the application semantics. Though metadata and transport mechanism can be located into quite different positions in the concept of network layers, we made them successfully collaborated by defining meta policy. In this paper, we describe our system architecture to define a meta policy based on the requirements and information contents from the application as a high level layer concept to be able to classify the network behavior. Our approach enables to multiple QoS control and collaboration among domains.

Keywords: *Metadata, Quality of Service, Diffserv.*

1. Introduction

On the next generation Internet technology, a lot of new applications have been developed such as voice, video stream, and database transaction. New multimedia applications use various types of media with different features. Each type requires a particular quality of communication service to be transported on the Internet. For example, voice data packets should be sent in order, without jitter. The database transaction packets can be sent in order, packet loss is a serious problem, however, a short delay is acceptable.

To achieve such different transfer requirements for each data type, following technologies are introduced in last a few years. The QoS for service differentiation is called "differentiated service (Diffserv)" [1]. Diffserv architecture consists of definition of transport service classes, detection of data flows, and control of data transmission according to the defined classes. Each class of service mapped to The packets that flow on the network are classified by Diffserv code point (DSCP) [2] has different ways of regulating network behavior, defining of transmission parameters and dropping packets. The classification is put in the packet header. The regulation of network behavior is called a "policy".

On the transport layer communications, applications are identified by transport protocol, port number and a pair of source and destination addresses. QoS control is required by the applications and service types. However, applications can not control detailed QoS because decision point of QoS control is low level transport. For the deployment the QoS technology, high level QoS requirement and policy from applications should convert to low level QoS control on the transport. However, it is not considered in QoS technology.

Additionally, some application needs multiple QoS control on the connection that has same transport protocol, port number and a pair of source and destination addresses. For example, web service provide many kinds of media data on the same protocol such as text, voice, and so on. Even if the media type which program used are same, QoS requirements are different according to various those meanings. For instance, in an emergency, a lot of people communicates each other, e.g calling ambulance, communication with family and friends, and so on. These communications have priority classes and expected to transmit differently.

Most application establishes peer to peer connection across policy domains that managed consistent policy through network nodes. It is needed to make consensus about policy among domains. Since Diffserv architecture defines only packet marking and per-hop forwarding behavior on network node, it is difficult to exchange policies requested each application.

To overcome these problems, we developed a policy control mechanism by application. Our mechanism adopted metadata to describe application poli-

https://doi.org/10.23106/dcmi.952107117

cies. Though metadata can be used for contents and applications to manage information, we confirmed that metadata can describe QoS policies for transport communications.

2. Policy Control Mechanism using Metadata

Metadata is the first level that the application provides the policy. At present, this policy described by the metadata does not control QoS functions. Our mechanism conveys application policy defined by metadata to the Diffserv DSCP. We adopt Dublin Core Metadata [5]in this paper because Dublin Core is the most popular metadata in the digital resource and its registry system is strongly needed for keep consensus among some domains.

The data and application need to be identified to the network, in order to gain service from the network appropriate to it. However, this information is usually available to the application only in its terms - object format, content, owner, and so on. What the network concerned is the type of service that the application requests. Such type of services should be expressed in terms of the local network management policy. Therefore, we need a translator.

We propose that the translator should have some form of API. It may resolve requests using a local configuration file which describe the meta policy. Meta policy select the appropriate DSCP value for the IP packet. When meta policy select DSCP value, it considers the operating system and application's attributes which are stored in LDAP directory. The application then invokes an interface method to apply the DSCP to the outbound data stream. We defined "meta policy" to translate from metadata to DSCP.

Metadata is described by resource administrator who controls application level policy. Meta policy is created by service administrator and network administrator. Service administrator gives application policy, network administrator specifies how to converts application policy to transport policy. Note that meta policy stores both application policy and transport policy at the local host computer and the local host computer specifies the DSCP value. In conventional QoS models, the transport policy is stored in the policy servers which mark the DSCP value. Since the policy servers cannot access the application policy at the local host computer, the conventional QoS models could not consider the application policy.

Fig. 1 shows system architecture which use metadata registory for keep same policy among some domains. application A sets a DSCP, and application B sets it according to a B's local meta policy.

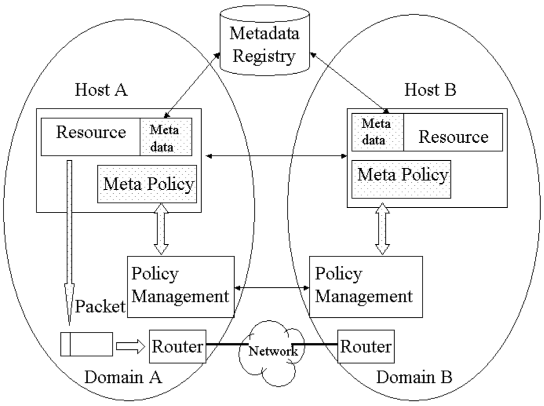

Figure 1. Policy mechanism architecture for two domains

3. Conclusion

Detailed QoS control is strongly required in the next generation Internet applications. Metadata is going to important for not only structuring and discovery digital resource but also communication interaction. This mechanism is discussing at IETF to make standard and deploy Diffserv QoS. An Request for comments (RFC) is going to publish as Best Current Practice (BCP) and start to discuss to make consensus [6].

In addition, This mechanism is going to be included in International Telecommunication Union, Telecom Standardization (ITU-T) F.706 recommendation: "Service Description for an International Emergency Multimedia Service".

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services", RFC2475, Dec., 1998.

[2] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC2474, Dec., 1998.

[3] Takeshi Aimoto, Shigeru Miyake, "Overview of DiffServ Technology: Its Mechanism and Implementation", IEICE transactions on information and systems, Vol.83, No.5, pp957–964, 2000.

[4] Weibel, S., Kunze, J., Lagoze, C., M.Wolf, "Dublin Core Metadata for Resource Discovery", RFC2413, Sep., 1998.

[5] Dublin Core Metadata Initiative, http://www.dublincore.org/

[6] R. Atarashi, F. Baker, "Reflexive DSCP Policy", Internet-Draft, IETF, June., 2002.

https://doi.org/10.23106/dcmi.952107117