

Architecture of End-to-End QoS for VoIP Call Processing in the MPLS Network

2004. 9. 29

**National Computerization Agency
(Republic of Korea)**

ChinChol Kim

[mail to] : cckim@nca.or.kr

Contents



Abstract

1. Introduction

2. QoS Architecture for VoIP in the NGN

3. Architecture of End-to-End QoS for SIP
Call Signaling

4. Implementation

5. Performance Analysis

6. Conclusion

Abstract



- This paper proposes
 - ✓ the architecture of **end-to-end QoS for VoIP call processing in the MPLS-based NGN** supporting the IPv6
 - ✓ The architecture include QoS resource management and differentiated call processing by extending SIP
- Also, we analyze the performance of call processing in the MPLS network
- A performance analysis
 - ✓ shows that there is **a considerable difference in end-to-end call setup delay depending on service priority**, in setting up SIP calls in the MPLS network

1. Introduction (1)



- With the fast development of network technology,
 - ✓ **VoIP is one of the core technology of NGN** and VoIP's core protocols are H.323 and SIP signaling protocols
 - ✓ At present, the text-based SIP has been adopted as the standard for NGN
- SIP must provide
 - ✓ a service quality better than the quality provided by the PSTN for call setup in NGN
 - ✓ and **offer priority-based call processing**, depending on the traffic properties of application services

1. Introduction (2)

- ❑ The VoIP service in NGN must guarantee
 - ✓ **call quality for voice data transfer** and **call setup quality for call setup**, in which the latter must precede the former
- ❑ On the other hand, **call setup quality does not guarantee end-to-end QoS**, since it goes through MPLS router and multiple SIP servers
- ❑ In order to resolve this problem,
 - ✓ This paper proposes the architecture of end-to-end QoS for VoIP call processing in an MPLS-based NGN
 - ✓ **QoS resource management** coordinates service priority in call processing
 - ✓ **Differentiated call processing technology** processes calls, applying the service priority negotiated in the SIP server

2. QoS Architecture for VoIP in the NGN

□ The call quality

- ✓ refers to speech coding distortion, terminal noise and overall delay caused by packetization, buffering, codecs
- ✓ can be guaranteed by QoS technology, including MPLS, Diffserv and IntServ

□ On the other hand, call setup quality

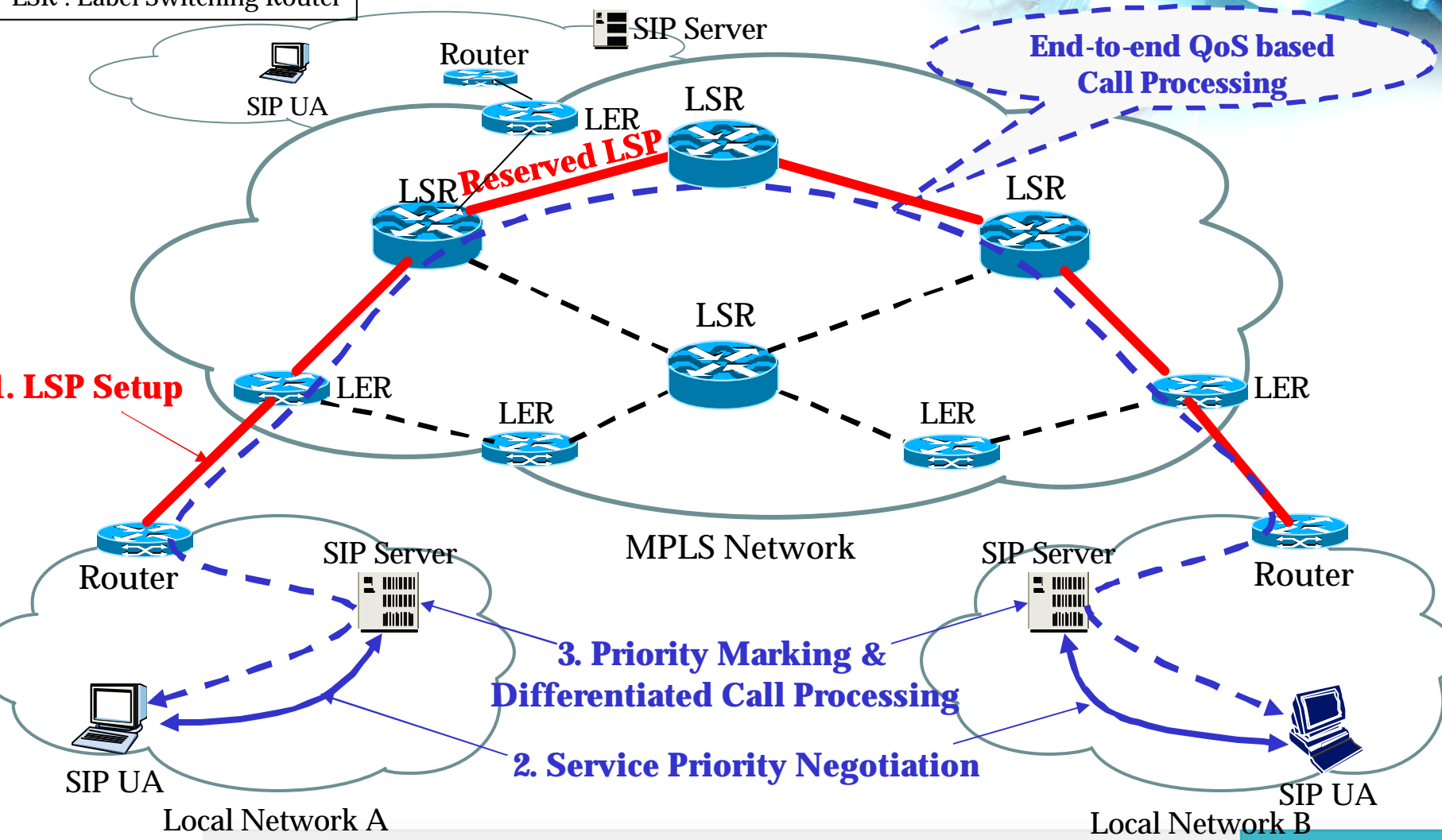
- ✓ refers to guaranteeing call setup, which is classified into call setup delay in the network and VoIP server
- ✓ Call setup delay in the network can be guaranteed through QoS technology, such as MPLS and DiffServ
- ✓ However, **there is no standard technology that ensures call setup delay in the VoIP server**

□ Therefore, we need

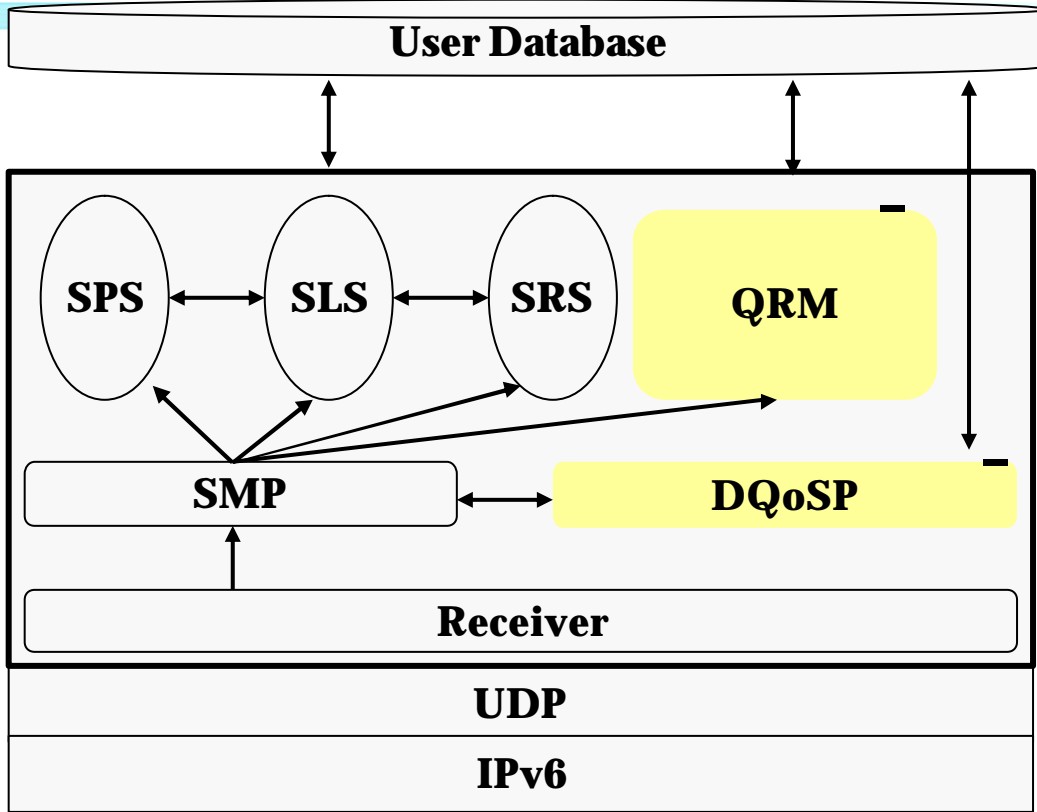
- ✓ **various forms of QoS mechanisms in the VoIP server** to guarantee end-to-end QoS for call setup quality

3. Architecture of End-to-End QoS for SIP Call Signaling

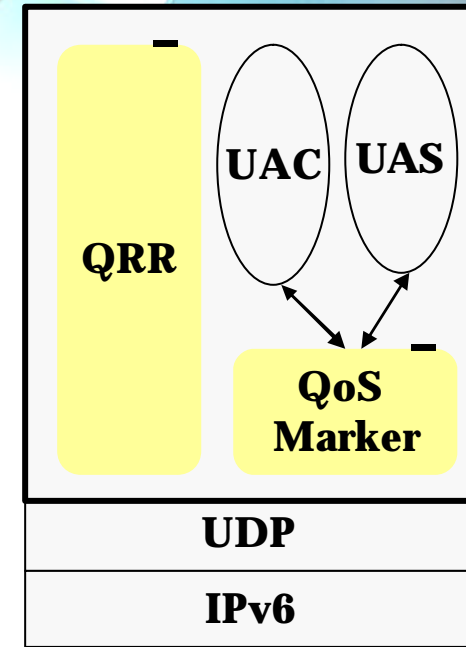
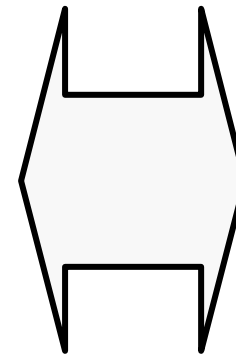
LER : Label Edge Router
LSR : Label Switching Router




3.1 System structure



SIP6d Daemon



SIP6 UA

 •SIP Extension

- SMP : SIP Message Parser
- QRM : QoS Resource Management
- QRR : QoS Resource Requester
- DQoSP : Differentiated QoS Processor
- SLS : SIP Location Server

- UAC : User Agent Client
- UAS : User Agent Server
- SPC : SIP Proxy Server
- SRS : SIP Redirect Server

3.2 SIP Message Extension and Flow (1)

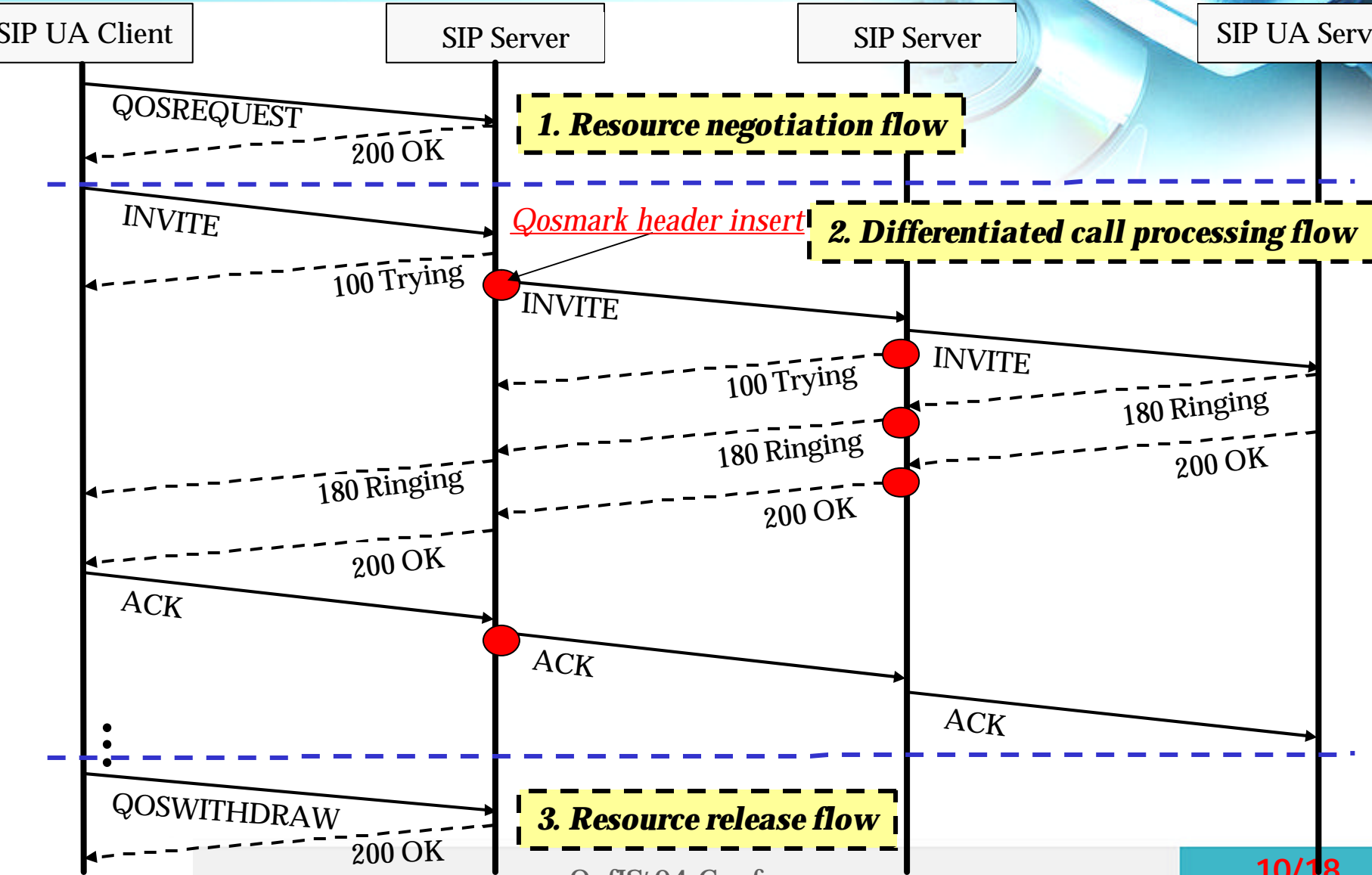
Table 1. SIP Method extension

Method Type	Description
QOSREQUEST	QOSREQUEST method is used by SIP UA to negotiate service priority with the SIP server. It sets up the service priority requested by the user through the Qosinfo header field.
QOSWITHDRAW	QOSWITHDRAW method is used by SIP UA to nullify negotiated service priority. It specifies the service priority to be nullified through Qosinfo header field.

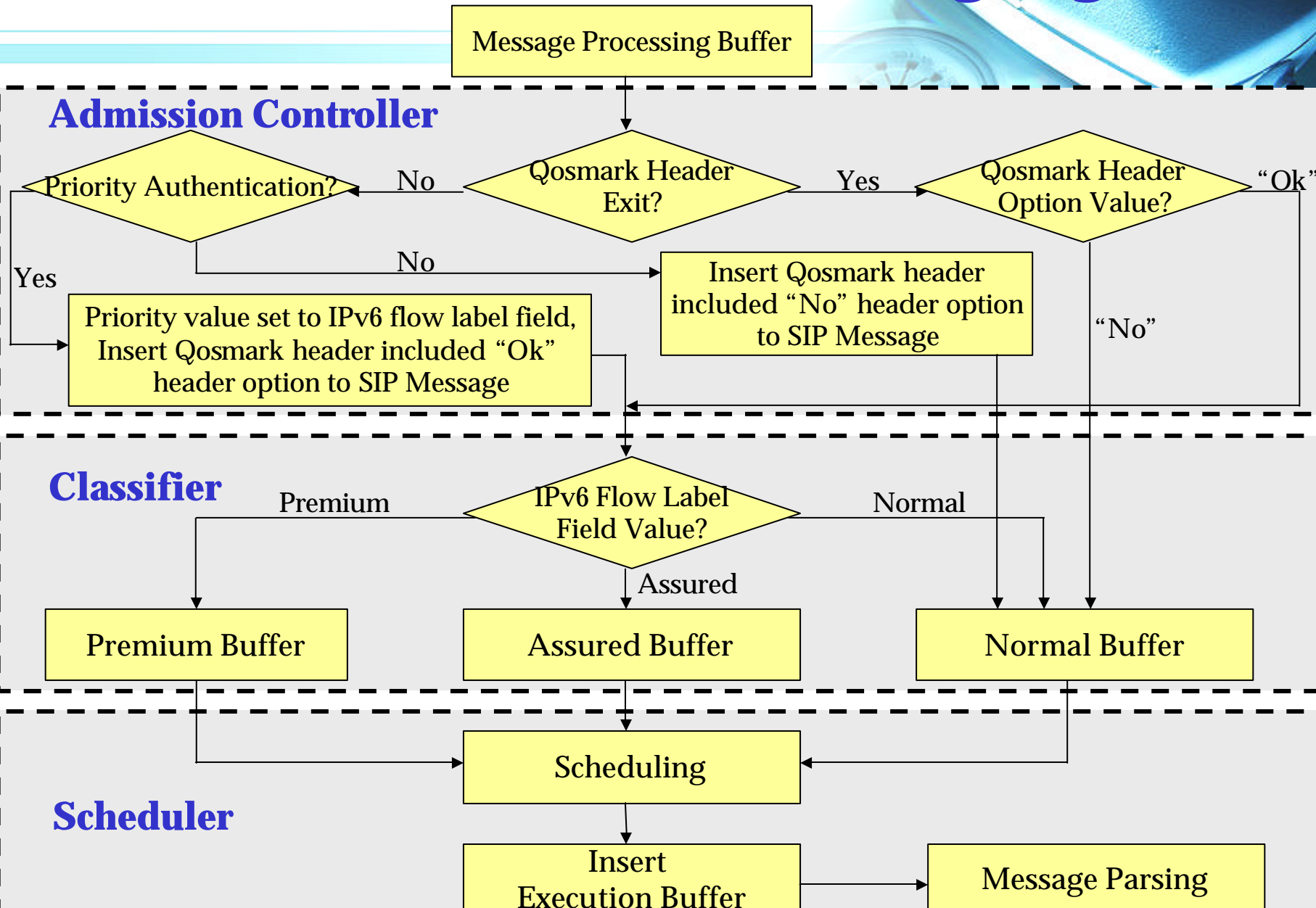
Table 2. SIP Header and Header Option extension

Header Type	Description
Qosinfo	Syntax Formalism : Qosinfo:“desired/release”=“ServiceLevel” “Desired” header option is used to set up the service priority requested by the user in QOSREQUEST method. It is also used to set up the negotiated service priority in 200 OK response messages. “Release” header option is used to release the service priority negotiated by the user in QOSWITHDRAW method and 200 OK response messages.
Qosmark	Syntax formalism : Qosmark : “ok / no” This is included in request/response message transferred to SIP message pass. If the header option in Qosmark header field is “Ok,” the SIP server applies differentiated call processing technology. If it is “No,” the SIP server doesn't.

3.2 SIP Message Extension and Flow (2)



3.3 Differentiated Call Processing Algorithm



4. Implementation

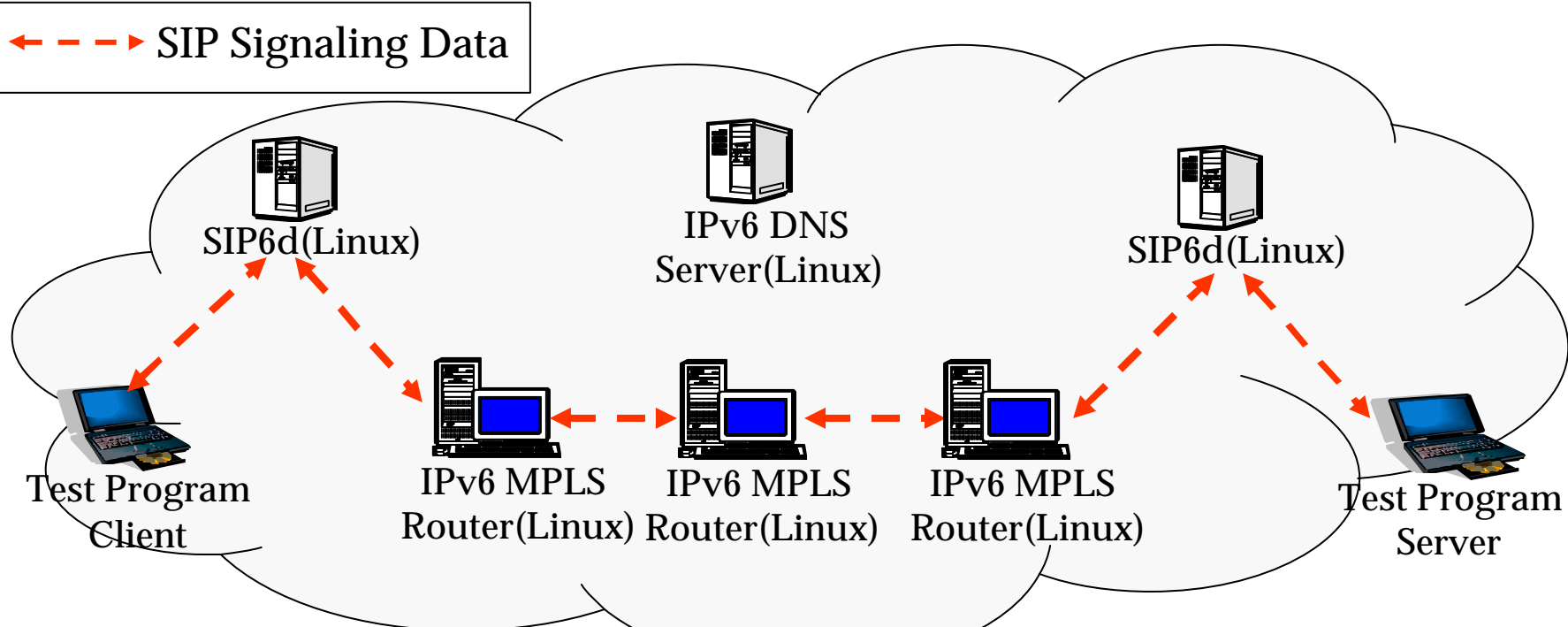


- We referred to the SIP source code from Columbia University
 - ✓ SIP6d is **implemented using C language in Linux system that supports IPv6**
 - ✓ The SIP UA is implemented using Tcl/Tk and C++
 - ✓ MySQL is used to manage user information
 - ✓ The major modules are implemented using POSIX thread technology

5. Performance Analysis (1)

It is composed of

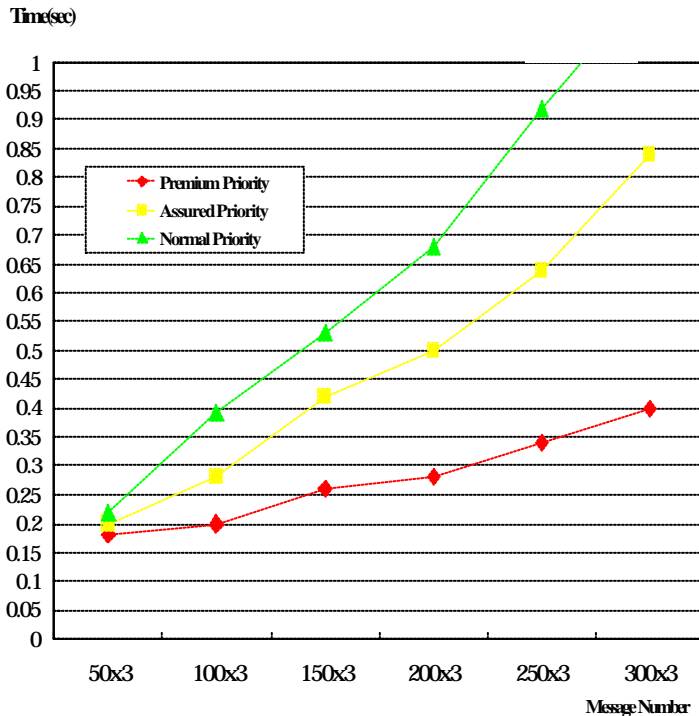
- ✓ two Linux servers with SIP6d, two PCs with test programs, and three Linux servers, which are used as routers with MPLS modules based on the software provided by Sourceforge.net



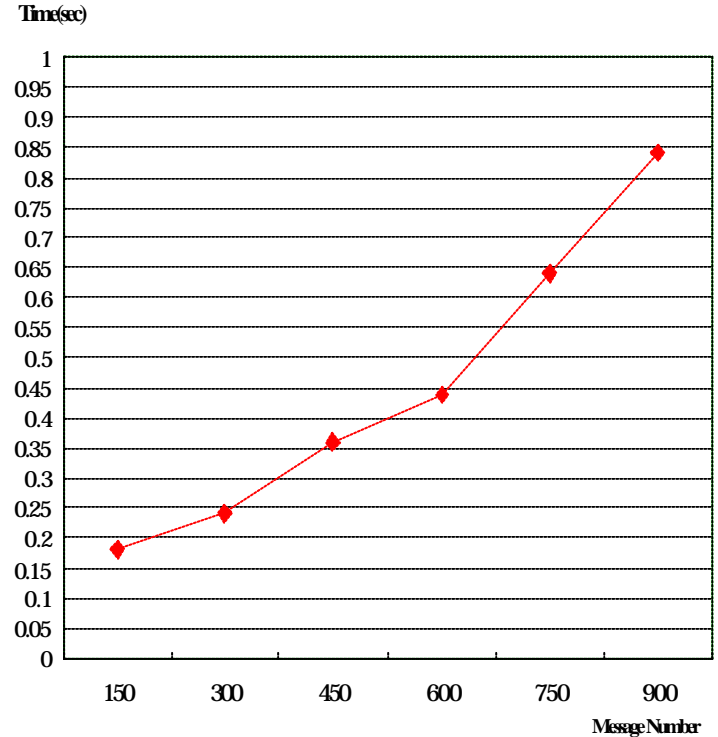
5. Performance Analysis (2)

- The experiment has adopted the following procedure
 - ✓ **Firstly**, the test client program generate equal number of three different INVITE messages and simultaneously transfers to SIP6d in the number of 50, 100, 150, 200, 250, and 300 messages of each priority.
 - ✓ **Secondly**, the two SIP6ds process the received messages, using differentiated call processing technology and transfer them to the test client server. **Each message is moved along the LSP path predetermined in the three routers**
 - ✓ **Next**, the test server program transfers 200OK response messages. Then, the test client program receives 200OK response messages coming through two SIP6ds, and measures session setup time, i.e. the average end-to-end call setup delay time. **For the comparison**, INVITE messages without priority are generated and sent to the **SIP6d without a differentiated call processing function** in the number of 150, 300, 450, 600, 750, and 900.

5. Performance Analysis (3)



(a) End-to-end Call Setup Delay in SIP6d supporting Differentiated Call Processing

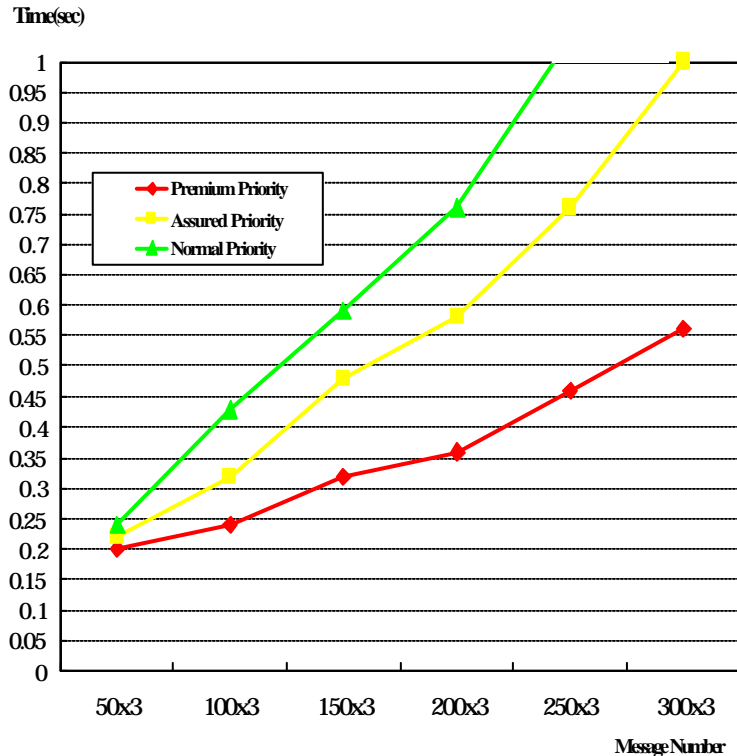


(b) End-to-end Call Setup Delay in SIP6d

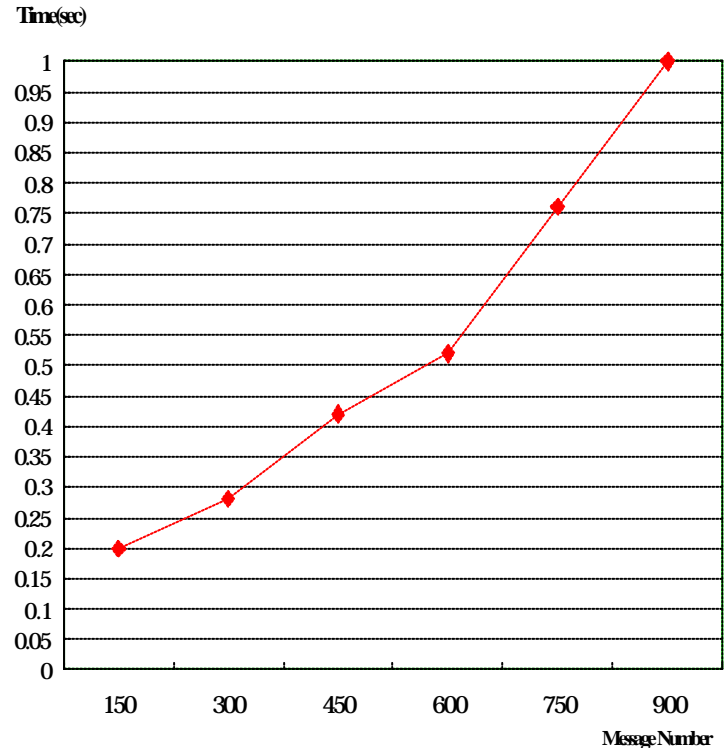
Fig. 6. End-to-End Call Setup Delay in SIP6d over MPLS Network

As one can see from the graph, the SIP6d that supports differentiated call processing shows a difference in call setup delay when processing messages. In particular, INVITE messages with premium priority have very short call setup delay. Therefore, we can see that INVITE messages with higher service priority have far shorter call setup delay than those with lower service priority. However, SIP6d that does not support differentiated call processing has no difference in call setup delay.

5. Performance Analysis (4)



(a) End-to-end Call Setup Delay in SIP6d supporting Differentiated Call Processing



(b) End-to-end Call Setup Delay in SIP6d

Fig. 7. End-to-End Call Setup Delay in SIP6d over Non-MPLS Network

Also, End-to-end call setup delay over MPLS Network show better performance than end-to-end call setup delay over Non-MPLS Network

6. Conclusion

- This paper proposes an architecture of end-to-end QoS for VoIP call processing in the MPLS network
 - ✓ The differentiated call processing technology **reserves resources by extending SIP**, and **minimizes end-to-end call setup delay for specific calls** by using priority scheduling technology in the application level
 - ✓ It also has an advantage of **setting up the service priority through the flow label field of IPv6 header**, considering future MPLS label mapping
- A performance analysis has showed
 - ✓ that SIP6d provides a very fast processing rate for messages with high service priority
 - ✓ These results prove that we can **provide excellent performance for call setups that require real-time or service priority when providing future voice service in the NGN based on MPLS**



Q&A

If you have a question, you send mail to me.

I will send response mail to you.