

특집

차세대 개방형 네트워크 QoS 미들웨어 기술

정희현, 송정호, 김병민, 김우성 (한국대학교 전자통신공학과)

I. 서론

우리나라의 통신서비스는 유선전화서비스를 시작으로 최근에는 데이터 서비스, 이동 통신서비스 등 다양한 형태의 서비스가 제공되고 있다. 그러나 현재 서비스 시장은 포화 상태에 이르고 있어 통신사업자들은 신규 시장을 확보하기 위하여 노력하고 있다.

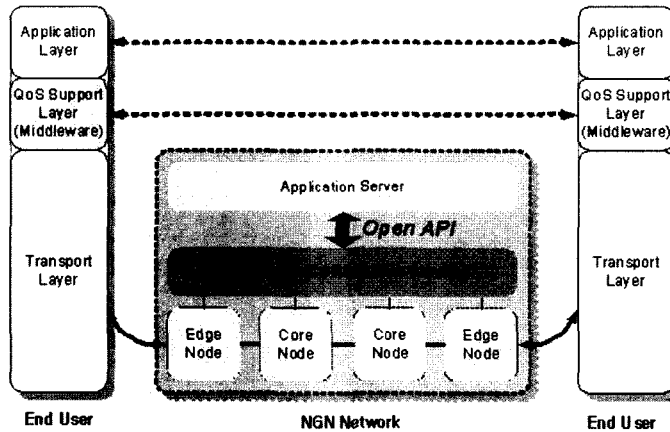
신규 시장 확보는 새로운 지능형 서비스의 개발을 통해 가능할 것으로 예상되는데, 기존의 수직적인 통신망구조에서 서비스의 개발은 쉽지 않다. 최근 하부 전달 망의 특성에 독립적인 Open API(Application Programming Interface)를 이용하여 망의 기능을 개방하고, 제3자 서비스 개발자들과의 제휴를 통해 다양한 형태의 유무선통합 지능형 서비스 개발을 유도하여 신규 서비스 창출이 가능한 차세대 네트워크에 대한 관심이 고조되고 있다¹⁾.

한편, 네트워크가 IP를 중심으로 통합되고, 멀티미디어에 기반한 다양한 서비스들과 사용자의 취향에 따른 특성화 서비스에 대한 요구가 증가함에 따라 각각의 서비스의 특성에 맞는 QoS의 제공을 위하여 네트워크 상황에 따라 적

절하게 대응하고, 사용자에게 차등화된 서비스를 제공해야 할 필요성이 대두 되었다. 본 고에서는 차세대 네트워크 환경에서 보다 효과적인 QoS의 제공을 위한 미들웨어 기술 현황 및 그 발전방향에 대하여 알아본다.

II. 차세대네트워크에서의 미들웨어

차세대 네트워크에서는 Open API를 통하여 망자원을 개방하여 기존의 수직적인 망구조를 탈피하려는 움직임을 보이고 있다. 차세대 네트워크의 구조는 그림 1을 통하여 잘 알 수 있다. Open API는 응용서버에 존재하는 통신서비스 로직과 하부 전달망 사이의 규격화된 공개 인터페이스로서, 하부 전달망의 특성과 형태에 관계 없이 다양한 통합 서비스의 제공을 가능하게 하는 핵심 기술이다. 서비스 로직은 Open API를 이용하여 통신 서비스의 구현에 필요한 하부 전달망에 대한 제어를 할 수 있다. 현재 대표적인 Open API에는 Parlay Group의 Parlay/OSA API와, Parlay X API가 있는데 전자는 CORBA(Common Object Request Broker Architecture)나 웹 서비스를 이용하여 하부 전



〈그림 1〉 차세대 네트워크의 구조

달망에 대한 제어가 이루어지고, 후자는 웹 서비스를 사용한다. 각 종단의 사용자는 Open API를 통하여 제어되는 하부 전달망을 통해서 연결된다¹⁾²⁾⁴⁾.

차세대 네트워크의 이와 같은 움직임은 사용자에게 다양하고 개인취향에 특성화 되어있는 서비스를 제공하기 위함이다. 현재까지의 통신 서비스는 주로 음성에 기반한 서비스가 주류를 이루고 있었다면 향후 통신망 서비스는 그림 2에 보이듯이 망 사업자가 제공하는 자원을 기반으로 멀티미디어, 개인화, 이동성의 특성이 복합적으로 연계된 서비스가 차세대 통신망 서비스의 방향을 주도할 것으로 예상된다. 하지만 이런 서비스들의 성공을 위해서 사용자들이 만족할만한 품질의 서비스를 제공하는 것이 중요하다. 따라서 QoS의 제공이 가장 중요한 조건이라고 할 수 있다. 차세대 네트워크에서 QoS는 그림 1에서 보였듯이 Open API를 이용한 하부 전달망 QoS와 각 종단간의 QoS로 나누어진다.

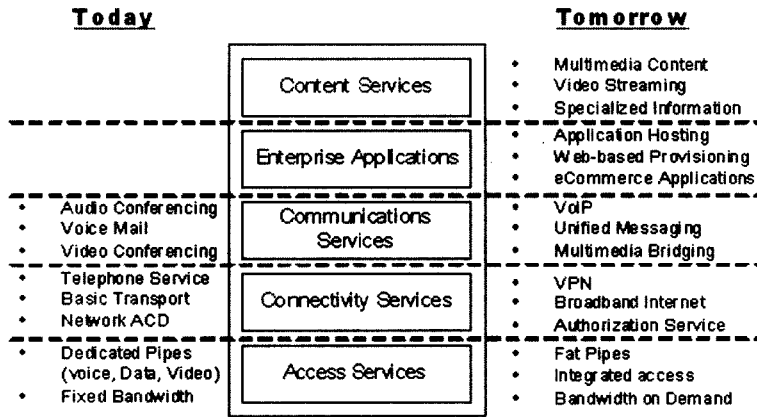
II-1장에서는 차세대 네트워크 미들웨어의 핵심인 Parlay/OSA API와 Parlay X API의 QoS

제공 방법에 대하여 알아보고, II-2장에서는 Open API가 하부 전달망에 대한 제어를 위해 사용하는 CORBA의 서버와 클라이언트 간 QoS제공 방법을 알아본다. 종단간 QoS제공에 대해서는 III장에서 알아본다.

1. Parlay/OSA API 와 Parlay X API의 QoS 제공 방법

Parlay/OSA API는 하부전달망의 QoS 제어를 위해 Connectivity Management와 Presence & Availability Management (PAM) SCF(Service Capability Feature)에서 QoS 제공을 위한 API를 제공한다. 앞서 설명하였듯이 Parlay/OSA API는 종단간 관점에서의 QoS제공 방법이 아니라 망사업자의 경계 안에서의 QoS 제공에 대하여 이야기한다²⁾.

Connectivity Management SCF는 응용서버를 소유한 서비스 운영자와 네트워크 제공자 사이에서 네트워크를 통해서 서비스 제공자의 패킷이 전달될 때 QoS 파라미터들의 설정을 위한 API를 포함한다. 네트워크 제공자는 QoS의 단



〈그림 2〉 차세대 서비스의 유형⁽⁵⁾

계에 따른 일정한 템플릿의 집합을 서비스 운영자에게 제공하여, 서비스 운영자의 선택에 따라 QoS를 제공한다. 서비스 운영자는 Framework SCF에 의해 인증과 권한 부여 과정을 거친 이후에 Connectivity Management SCF의 사용이 가능해진다.

즉 서비스 운영자는 `getQoSMenu()`, `getTemplateList()`, `getTemplate()` 등의 함수들을 통해서 사용 가능한 QoS 서비스의 목록을 받아올 수 있고, 받아온 QoS 서비스 목록 중에서 유효한 서비스의 사용에 대한 요청을 하고, 네트워크 제공자로부터 승인을 받은 후에 데이터 패킷을 전송할 수 있다. 이 밖에도 Connectivity Management SCF는 네트워크의 상태를 알아오는 함수, 현재 제공받고 있는 QoS의 정도를 알아오는 함수 등 다양한 함수를 제공한다. 이런 함수들을 통해서 현재 네트워크의 상태와 네트워크를 통해서 제공받고 있는 QoS의 정보를 알 수 있다. QoS에 대한 정보는 delay, jitter, loss 등으로 표현되고, 각각에 대한 평균과 측정기간, 최대치, 최소치 등 자세한 정보를 제공한다. Connectivity Management SCF에서 제공하는

인터페이스와 함수는 표 1과 같다.

Presence & Availability Management SCF는 디지털 인증, agent의 특성과 존재, entity의 기능과 상태, 그리고 다양한 통신 형태를 위한 entity의 존재와 사용가능성에 대한 정보의 유지, 공개 등에 대한 표준을 제공한다. 이런 표준은 다양한 네트워크 기술에 대한 호환성을 제공하여 여러 네트워크 사이를 넘나드는 서비스의 개발을 쉽게 하고 사용자에게 유연성과 그들의 통신에 대한 제어를 제공한다. Presence & Availability Management SCF에서는 서비스의 사용을 위해서 `getAuthToken()` 함수를 이용하여 권한을 부여 받는다. 이후 받아온 토큰을 이용하여 `getIdentityPresence()` 함수를 호출하고, 서비스의 속성에 대한 정보를 받아온다. 또 `getAvailability()`를 호출하여 사용 가능한 서비스에 대한 정보를 받는다. 처음 `getAutoToken()`을 이용하여 받아온 토큰은 여러 번 사용이 가능하다. 이 밖에도 agent에 대한 정보를 계속해서 유지해주기 위한 다양한 함수들을 제공한다. 이런 함수들을 통하여 현재 제공받고 있는 agent에 대한 정보를 유지하고, 원하는 서비스

〈표 1〉 Connectivity Management SCF의 인터페이스^[2]

Interface Class IpConnectivityManage	
getQoSMenu(): IpInterfaceRef	클라이언트에서 QoS menu 인터페이스에 대한 참조를 받아오기 위해 사용한다.
getEnterpriseNetwork(): IpInterfaceRef	네트워크 제공자로부터 서비스 사용자가 제공받고 있는 서비스의 정보를 가진 인터페이스의 핸들을 얻기 위해 사용한다.
Interface Class IpEnterpriseNetwork	
getSiteList(): TpStringList	네트워크 제공자의 서비스를 통해서 연결되어 있는 서비스 사용자 site의 ID 목록을 받아온다.
getVPrN(): IpInterfaceRef	이전에 제공받고 있던 virtual private network(VPrN)의 정보를 가진 인터페이스 핸들을 얻기 위해 사용한다.
getSite(siteID:in TpString) : IpInterfaceRef	주어진 siteID에 의해서 특정한 site의 정보를 가진 인터페이스 핸들을 얻기 위해 사용한다.
Interface Class IpEnterpriseNetworkSite	
getSAPList():TpStringList	사용자 VPN의 service access point(SAP)의 list를 얻기 위해서 사용한다. VPrP의 endpoint로 사용될 수 있다.
getSiteID(): TpString	이 site의 ID를 얻는데 사용한다.
getSiteLocation(): TpString	Site의 위치를 얻는데 사용한다.
getSiteDescription(): TpString	이 site와 관계된 설명을 받아온다.
getIPSubnet(): TpIPSubnet	이 site의 IP subnet 정보를 얻는다.
getSAPIPSubnet(sapid: in TpString) : TpIPSubnet	주어진 SAP ID의 IP 주소를 얻어 온다.
Interface Class IpQoSMenu	
getTemplate(templateType: in TpString): IpInterfaceRef	네트워크 제공자로부터 모든 QoS 파타미터를 포함하는 명확한 템플릿의 인터페이스를 얻기 위해 사용한다.
getTemplateList(): TpStringList	QoS서비스에 대한 템플릿 목록을 받아온다. Ex) Gold, Silver, ..
Interface Class IpQoSTemplate	
getTemplateType(): TpString	템플릿의 종류를 받아온다.
getDescription(): TpString	템플릿에 포함된 QoS서비스에 대한 설명을 받아온다.
setSlalD(slalID: in TpString): void	Service level agreement(SLA) ID를 저장하기 위하여 사용한다.
setPipeQoSInfo(PipeQoSInfo: in TpPipeQoSInfo): void	템플릿에 의해서 제공되는 virtual provisioned pipe의 종단이나, Load 파타미터, 트래픽의 방향 등에 대한 정보를 포함하는 pipe QoS 정보를 설정 위하여 사용한다.
getPipeQoSInfo(): TpPipeQoSInfo	Pipe QoS 정보를 얻기 위하여 사용한다.
getValidityInfo(): TpValidityInfo	네트워크 제공자가 템플릿을 위하여 설정한 시간주기를 받아온다.
setValidityInfo(validityInfo: in TpValidityInfo): void	새로운 VPrP를 위하여 필요한 시간주기를 정하기 위하여 사용한다.

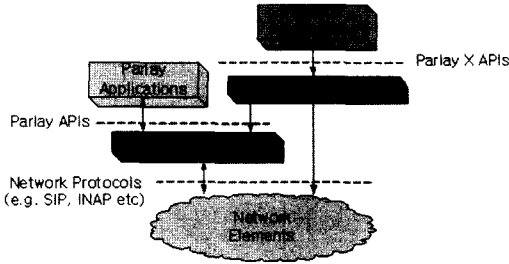
setProvisionedQoSInfo(ProvisionedQoSInfo TpProvisionedQoSInfo): void	서비스 운영자 QoS 파라미터를 위해 필요한 값들을 설정하기 위하여 사용한다.
getProvisionedQoSInfo(): TpProvisionedQoSInfo	템플릿과 관련된 기본값을 얻기 위하여 사용한다.
getDsCodepoint(): TpDsCodepoint	템플릿에 의해 제공되는 QoS 서비스의 DiffServ Codepoint를 얻기 위하여 사용한다.
Interface Class VPrp	
getVPrpList(): TpStringList	서비스 운영자의 네트워크를 위하여 이미 설치된 virtual provisioned pipe를 위한 VPrp ID의 list를 얻기 위해 사용한다.
getVPrp(vPrpID: in TpString): IplInterfaceRef	virtual provisioned pipe의 핸들을 얻는다.
createVPrp(templateRef: in IplInterfaceRef): IplInterfaceRef	새로운 virtual provisioned pipe를 생성한다. Pipe QoS정보, provisioned QoS정보, SLA ID등을 포함한다.
deleteVPrp(vPrpID: in TpString): void	virtual provisioned pipe를 삭제하는데 사용한다. VPrp는 다음과 같은 상태 중 하나를 가질 수 있다: active, pending, disallowed
Interface Class IpVPrp	
getVPrpID(): TpString	virtual provisioned pipe의 ID를 얻기 위하여 사용한다.
getSlalD(): TpString	SLA ID를 얻기 위하여 사용한다.
getStatus(): TpVprpStatus	virtual provisioned pipe이 상태를 얻기 위하여 사용한다.
getProvisionedQoSInfo(): TpProvisionedQoSInfo	Provisioned QoS 정보의 셋을 얻기 위하여 사용한다.
getValidityInfo(): TpValidityInfo	VPrp가 유효한 시간주기를 얻는데 사용한다.
getPipeQoSInfo(): TpPipeQoSInfo	VPrp를 위한 pipe QoS 정보를 얻고 설정하는데 사용한다.
getDsCodepoint(): TpDsCodePoint	virtual provisioned pipe의 DiffServ Codepoint를 얻는데 사용한다.

를 제공받을 수 있다. 또 Presentity라 불리는 data type을 통하여 정보를 제공하는데 이것을 통하여 사용자와 네트워크의 상태, 위치 등을 알 수 있고, 현재 제공되는 통신의 우선순위를 알 수 있다.

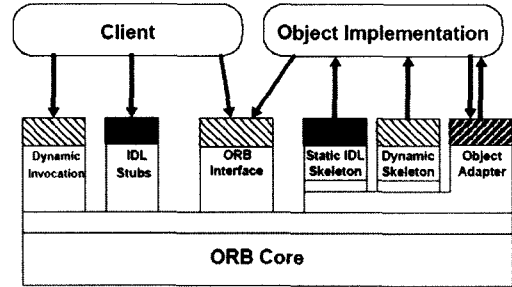
한편, Parlay X API는 통신망에 대한 전문 지식이 없는 IT 분야 종사자들에게 손쉬운 통신망 제어 방법을 제공 함으로서 IT 기반의 텔레포니 서비스 로직을 개발할 수 있도록 하기 위해

Web서비스 개념을 이용하여 설계되었다. Parlay X API는 80%의 서비스 로직이 망 자원의 20%만 이용한다는 80/20 원칙과, 복잡한 Parlay API를 보다 단순하게 정의하자는 KISS(Keep it Simple, Stupid) 원칙에 의해 정의되었다³⁾.

Parlay X는 현재의 Parlay/OSA API를 더욱 추상화한 API형태로 제공된다. 따라서 망의 구조와 망의 기능에 대하여 잘 알지 못하는 개발자



〈그림 3〉 Parlay/OSA API와 Parlay X API



〈그림 5〉 Object Request Broker의 구조^[6]

General	Introduction
Common Data	Common Data Definition
Framework	
User Interaction	
Mobility	
Terminal Capability	
Data Session Control	Payment
Generic Messging	Account Management
	User Status
Account Management	Terminal Location
Charging	
Policy Management	

〈그림 4〉 Parlay X API와 Parlay API^{[2][3]}

역시 추상화된 함수들을 통하여 서비스 로직을 보다 손쉽게 개발할 수 있도록 해준다. 그림 3은 Parlay X API와 Parlay/OSA API와의 관계를 나타내었다. Parlay X API는 Web 서비스나 Parlay/OSA API를 통하여 전달망에 대한 조절이 가능하다.

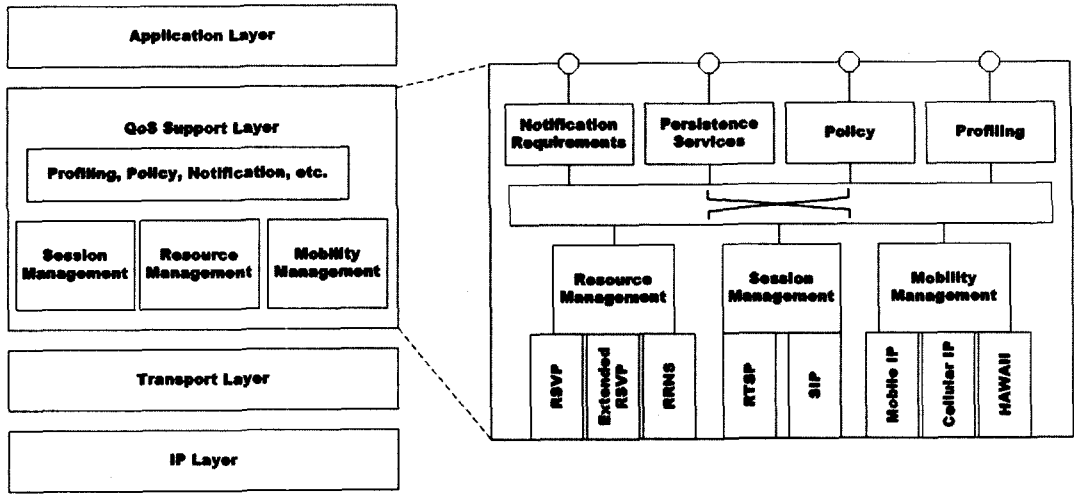
그러나, 그림 4와 같이 Parlay X API와

Parlay/OSA API의 구성에 대하여 살펴보면 Parlay X API는 Parlay/OSA API와는 다르게 서비스 로직에 구현과 관련하여 다양한 API를 제공하지만 QoS에 관련한 API는 현재 버전에 존재하지 않는 것을 알 수 있다. Parlay X API를 이용한 서비스는 다양하고 단순화된 API를 이용해서 쉽게 개발될 수 있지만 전달망의 자원 등 QoS에 대한 조절이 전혀 이루어 지지 않기 때문에 네트워크에 장애가 발생하였을 때 서비스를 원활하게 제공할 수 없다. 따라서 Parlay X API에도 위에서 언급한 내용과 같이 전달망의 QoS를 보장할 수 있는 API가 추가되어야 한 단계 더 뛰어난 서비스를 제공할 수 있을 것이다.

2. CORBA의 QoS 제공 방법

지금까지는 Parlay/OSA API와 Parlay X API의 QoS 제어에 대하여 알아보았다. 이번 절에서는 Parlay/OSA API와 Parlay X API의 요청을 하부 전달망에 전달하는 CORBA에 대하여 알아본다.

CORBA는 OMG(Object Management Group)에서 제정한 미들웨어로서 ORB(Object Request Broker)는 개념을 이용한다. ORB는 클라이언트 프로그램이 네트워크에서



〈그림 6〉 QoS support Layer

서버가 어디 있는지, 또는 서버 프로그램이 어떤 인터페이스를 가질지 인식하지 않고서도, 서버 프로그램이나 객체로부터 서비스를 요구할 수 있도록 한다. ORB들간의 요구와 응답을 성립시키기 위해 프로그램들은 GIOP(General Inter-ORB Protocol)를, 인터넷에서는 IIOP(Internet Inter-ORB Protocol)를 사용한다. IIOP는 GIOP의 요구와 응답을 각 컴퓨터의 인터넷 TCP 계층으로 대응시킨다. CORBA의 ORB 구조는 그림 5와 같다. CORBA의 클라이언트는 요청의 생성을 위해서 Dynamic Invocation이나 IDL Stubs를 이용한다. 또한 몇 개의 함수는 ORB 인터페이스를 통해서 구현된 객체에 직접 연결할 수 있다. Request는 객체에 의해서 생성된 skeleton들에 의해서 객체로 전달되게 된다. CORBA는 Real-Time CORBA, Minimum CORBA, QoS등의 여러 가지 표준을 제공하고 있으며 다른 미들웨어와의 mapping에 대한 API도 제공한다⁶⁾.

CORBA는 CORBA 클라이언트들과 서버들

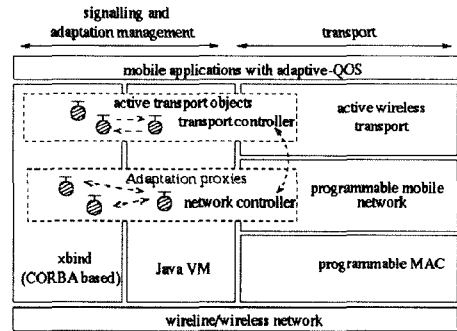
사이의 메시지 전달에 대한 QoS를 제공하기 위하여 Rebind Support, Synchroni-zation Scope, Request and Reply Priority, Request and Reply Timeout, Routing, Queue Ordering등 5가지의 파트에서 API를 제공한다. Rebind Support 파트에서는 객체를 호출하는 동안 rebinding이 발생할 때, 투명성에 대한 정도를 표현하거나 새롭게 투명성의 단계를 지시하는 등의 API를 제공한다. Synchronization Scope에서는 단방향 오퍼레이션에 대하여 동기화의 정도를 지시하는 API를 제공한다. Request and Reply Priority, Request and Reply Timeout, Routing, Queue Ordering 파트에서는 각기 요청과 응답에 대한 우선순위를 지정해 줄 수 있다. 또한 각각의 우선순위에 따른 라우팅 방식과 큐의 정렬방식까지 조절할 수 있도록 한다. 또한 요청과 응답에 대한 시간초과와 각각의 시작시간과 종료시간에 대한 기록이 가능하도록 한다.

III. 종단간의 QoS제공을 위한 미들웨어

차세대 네트워크에서 Parlay/OSA API는 종단간의 관점에서의 QoS제공 방법이 아니라 망사업자의 경계에서 QoS의 제공에 대하여 이야기한다. 하지만 QoS는 미디어형태 별로 다르게 정의될 수 있고, 응용 서비스 별로 다르게 정의될 수 있기 때문에 종단간에서도 QoS 제어가 가능하여야 한다. 또한 차세대 네트워크에서 제공하는 QoS 방식에서는 종단간 QoS 제어에 대한 관리주체가 모호하고, 최종적으로 사용자가 느끼는 QoS 정도가 명시되지 않으므로 네트워크 내부에서의 QoS 뿐만 아니라 종단간 QoS 역시 필요하다. 하지만 차세대 네트워크는 다양한 액세스 네트워크에 연결되어 있어 종단간의 응용 서비스를 위한 통합적인 QoS 제공이 쉽지 않다. 그러므로 종단간 QoS를 제공하고 여러 액세스 네트워크에 상관없이 동작할 수 있는 응용 서비스를 개발하기 위해서는 미들웨어의 사용이 필수적이다. 이러한 종단간의 QoS 지원은 그림 6과 같이 응용 계층과 전송 계층 사이에 QoS 지원을 위한 미들웨어를 추가함으로써 제공할 수 있다.

QoS 지원 미들웨어는 서비스 로직으로부터 어떤 서비스에 관한 요구가 들어왔을 때 서비스에 대한 특성이나, 서비스의 품질, 서비스의 정책 등에 관한 정보를 고려하여, 이것을 기반으로 네트워크 상황에 적응적인 대역폭 할당이나 종단간 세션을 조절해야 한다.

일반적인 유선전화 환경에서는 call의 품질이 갑작스럽게 떨어질 가능성이 적지만 사용자가 이동성을 가지거나 패킷망에 연결되어 있다면, 네트워크 혼잡이나 이동으로 인한 패킷 손실로 call의 품질이 급격하게 저하될 우려가 있다. 이



〈그림 7〉 Mobeware의 구조^[7]

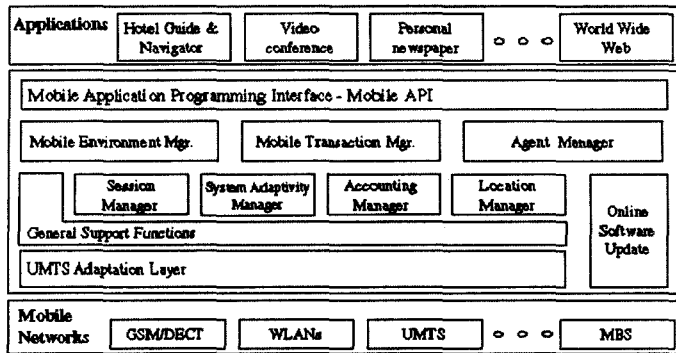
러한 경우 미들웨어는 네트워크의 모니터링 정보나 클라이언트의 피드백 메시지를 기반으로 QoS의 단계를 적절히 조절하고, 이동중인 사용자의 경우 이동경로를 예측하여 충분한 양의 대역폭을 확보하고, 패킷망의 경우 패킷이 전송될 라우팅 경로를 미리 지정하고 사용될 대역폭을 예약하여 call의 사용중간에 서비스의 품질이 떨어지는 것을 방지할 수 있다.

또한 실시간 멀티미디어 call인 경우에는 일반적인 call의 경우보다 더 많은 관리를

필요로 한다. 만약 네트워크의 상태가 나빠져 충분한 대역폭 보장이 불가능해지면, 제공받고 있던 QoS의 품질을 다시 조절하거나 종단간의 세션설정을 효과적으로 조절함으로써 사용자에게 끊임 없는 멀티미디어 전송을 보장한다. 다시 말해, 끊임 없는 서비스를 원하는 사용자에게는 QoS의 품질을 저하시켜 끊임 없는 서비스를 제공해주고, call의 품질을 중요시하는 사용자에게는 서비스를 일시 정지 한 후 네트워크의 상태가 양호해졌을 때 다시 서비스 해 주도록 세션을 조절해야 한다.

이와 같이 종단의 미들웨어에서 QoS를 제공하는 연구는 현재 여러 곳에서 진행되고 있다.

콜롬비아대학의 DCC는 QoSSocket이라는



〈그림 8〉 MASE의 구조[8]

QoS API를 개발하였고, 콜롬비아대학의 COMET에서는 Mobile 단말, 베이스 스테이션, ATM 스위치간에 멀티미디어 통신을 제공하기 위한 Mobiware 프로젝트를 진행 중에 있다. Mobiware에서는 종단간에서 QoS 시그널링과 제어를 위해서 JAVA RMI, CORBA 기반의 분산 컴포넌트 기술을 이용한다. Mobiware의 구조는 그림 7과 같다.

Mobiware는 전송 계층에서 종단간의 음성, 화상, real time 데이터 서비스를 제공하고, segmentation 과 reassemble, rate control, flow control, playout control, resource control, buffer management 등의 기능을 수행할 수 있다. 또한 Mobiware의 기능으로는 QoS controlled handoff, mobile soft-state, flow bundling이 있다. 단말의 서비스 로직은 네트워크 상황이 변함에 따라 적응적인 QoS를 보장한다.

또 다른 연구로는 ACTS의 OnTheMove Project로서 유럽의 기업들이 컨소시엄을 구성되어 UMTS(Universal Mobile Telecommunications System) 기반의 Mobile 장치상에서 수행되는 미들웨어, MASE(Mobile Application Support Environment)와 Mobile-API가 개발되었다. MASE의 주된 목표는 멀티미디어 서비스

를 이동단말 사용자에게 까지 확장하고, 이동망에서 그 서비스를 제공하는데 어려움이 없게 하는 것이다. MASE의 구조는 그림 8과 같다.

MASE는 종단간에서 QoS의 관리를 위하여 Session manager, system adaptivity manager, location manager, accounting manager, the UMTS adaptation layer 등의 기능을 지원한다. Session manager는 QoS, user/terminal profile 등에 기초하여 적합한 UMTS Adaptation Layer를 결정하여, 세션에 대한 연결과 종료를 제공한다. UMTS Adaptation Layer는 UMTS를 위해 개발된 미들웨어로서 네트워크 통신의 개설, 종료, 유지할 수 있으며, incoming call에 대한 경고의 기능을 제공한다.

IV. 결론

현재 네트워크의 형태는 서비스의 개발이 쉽지 않은 기존의 수직적인 구조에서 벗어나, 수평적 구조의 차세대 네트워크로의 발전할 것으로 예상된다. 차세대 네트워크의 서비스는 기존의 단순한 전화 서비스에서 새로운 형태의 서비스, 즉 개인에 특성화 되고 멀티미디어의 요소를 포함하는 서비스로 발전할 것이고, 이런 멀티미디어

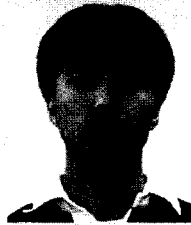
어 서비스를 만족할 만한 수준으로 제공하기 위해서는 QoS를 확실하게 제공해주어야 한다.

본고에서는 차세대 네트워크 미들웨어의 QoS 제공방법에 대하여 알아보았다. 차세대 네트워크의 QoS는 Open API를 통한 하부 전달망에 대한 QoS 제어와 종단간의 QoS로 구분할 수 있다. Open API의 대표적인 Parlay API는 하부 전달망의 QoS를 위해서 Connectivity Management와 PAM SCF 등을 제공한다. 또한 종단간의 QoS는 응용 계층과 전달계층 사이에 QoS 지원 미들웨어를 추가함으로써 제공할 수 있다. 따라서 차세대 네트워크에서 QoS 미들웨어 기술은 차세대 네트워크에서 원활한 서비스를 위한 중요한 기술이 될 것이다.

참고문헌

- [1] 정의현 김화성, "유무선 통합서비스를 위한 Open API", 한국통신학회지 정보통신 Vol.20, No.11 Nov. 2003
- [2] "Parlay Specification V4.1."
- <http://www.parlay.org/>
- [3] "Parlay X Web Services Specification V1.0." -
<http://www.parlay.org/>
- [4] "Parlay/OSA - a New Way to Create Wireless Services."
- <http://www.parlay.org/>
- [5] 최영일, "Parlay/OSA API 표준화 동향", 개방형 서비스 기술 워크샵, 2003년 7월.
- [6] "Common Object Request Broker Architecture: Core Specification" - <http://www.omg.org>
- [7] "The Mobicore Toolkit" - <http://com.et.columbia.edu/mobicore/>
- [8] "Middleware for a New Generation of Mobile Networks: The ACTS On TheMove Project" -
http://www.isoc.org/inet96/proceedings/a6/a6_3.htm

저자소개



정의현

2003년 광운대학교 전자공학부 학사
2003년-현재 광운대학교 전자통신학과 석사과정



홍정표

2003년 광운대학교 전자공학부 학사
2003년-현재 광운대학교 전자통신학과 석사과정



김형민

2002년 광운대학교 전자공학부 학사
2002년-현재 광운대학교 전자통신학과 석사과정



김화성

1981년 고려대학교 전자 공학과 졸업
1983년 고려대학교 전자 공학과 석사
1996년 Lehigh univ. 전산학 박사
1984년-2000년 ETRI 책임 연구원
2000년-현재 광운대학교 전자공학부 교수
주관심분야 NGN 미들웨어 환경, 이동인터넷
QoS, 그리드컴퓨팅