

Providing End-to-End QoS for Multimedia Applications in 3G Wireless Networks

Katherine Guo Sampath Rangarajan M. Ali Siddiqui Sanjoy Paul

Center for Networking Research, Bell Laboratories, Lucent Technologies, Holmdel, NJ, USA.

ABSTRACT

As the usage of wireless packet data services increases, wireless carriers today are faced with the challenge of offering multimedia applications with QoS requirements within current 3G data networks. End-to-end QoS requires support at the application, network, link and medium access control (MAC) layers. We discuss existing CDMA2000 network architecture and show its shortcomings that prevent supporting multiple classes of traffic at the Radio Access Network (RAN). We then propose changes in RAN within the standards framework that enable support for multiple traffic classes. In addition, we discuss how Session Initiation Protocol (SIP) can be augmented with QoS signaling for supporting end-to-end QoS. We also review state of the art scheduling algorithms at the base station and provide possible extensions to these algorithms to support different classes of traffic as well as different classes of users.

Keywords: End-to-end QoS, 3G wireless network, real-time multimedia applications, SIP, PPP, GRE, RLP

1. INTRODUCTION

The main motivation for the transition from circuit switched networks to packet switched networks is to provide flexible voice and data services with reduced cost to service providers and end users. In order to provide multimedia services, there has been tremendous effort in wireline networks to address issues in bandwidth management and quality of service provisions. In contrast, the effort in wireless networks is at its beginning stage. 3GPP which is the standard body in charge of UMTS standards has recently added the IP Multimedia Subsystem (IMS) to UMTS network architecture release 5 to provide differentiated services to different classes of traffic.^{1,2} 3GPP2 which is the standard body in charge of CDMA2000 standards is also adding IMS for CDMA network architecture towards the goal of a harmonized 3GPP/3GPP2 IMS network architecture.³

Different multimedia applications have different QoS requirements. For example, applications such as email or file transfer have variable bandwidth requirement, but not very strict delay or loss requirement because of their non-real time nature. Both video and audio streaming can tolerate some small initial start-up delay, and 5-second jitter because the default buffer size at the client is set to 5 seconds for popular streaming clients such as RealPlayer, Apple QuickTime Player and Microsoft Media Player.⁴⁻⁶ Video streaming requires bandwidth in the range of 100 Kbps to 10 Mbps depending on video codec; whereas bandwidth requirement for audio streaming is a lot smaller, in the range of 20-320 Kbps depending on audio codec. Streaming applications can tolerate some packet loss even when UDP is used for transport depending on the loss tolerance of the media codec used.

Online games are designed with small message sizes, therefore bandwidth requirement is very low. However, compared with multimedia streaming, real-time, multi-player online gaming has more strict delay requirement because of its interactive nature. For example, round-trip delay for messages in real-time games of more than 75 ms could lead to un-smooth gaming experience for players.⁷

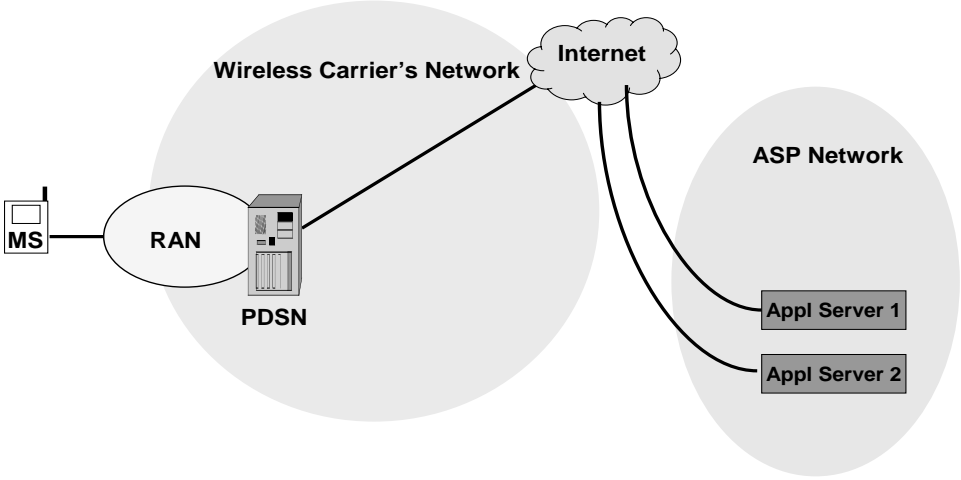
Voice over IP (VoIP) bandwidth requirement is low. For example, if Enhanced Variable Rate Codec (EVRC) is used, 9.6 Kbps is the maximum rate. But it has very strict delay and jitter requirements. One-way latency from one mobile handset to the other mobile handset should not exceed 250 ms, and the tolerable jitter is 20 ms.

To summarize, applications such as multimedia streaming require high bandwidth, whereas applications such as VoIP and online gaming require low delay and low jitter; other applications such as video conferencing require both high bandwidth and low delay and low jitter. As a result, proper support for end-to-end QoS in wide area 3G networks is

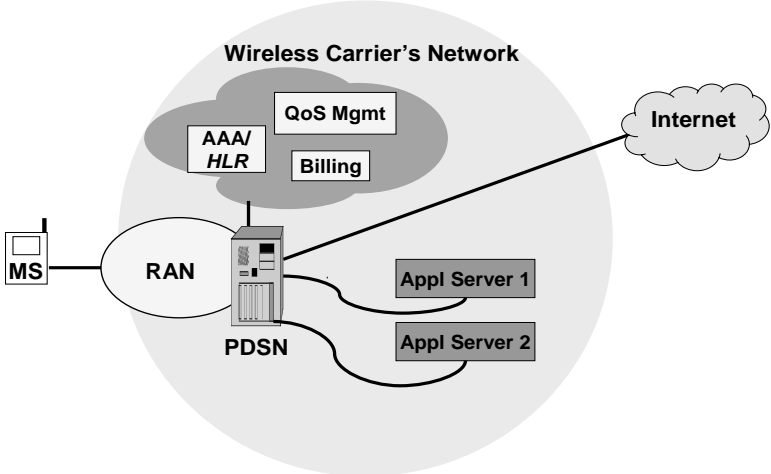
Address: 101 Crawfords Corner Road, Holmdel, NJ 07733, USA. Email: {kguo, sampath, alisidd, sanjoy}@bell-labs.com.)

becoming increasingly important. In general, QoS means low latency, low delay, low jitter, low loss, adequate bandwidth and above all, good end-user experience. However, all the metrics do not necessarily apply to all applications.

On top of different QoS requirements for different applications, different users also have different QoS needs. For example, service providers can offer Gold, Silver and Bronze class of services to different users. When there is network resource contention, traffic for Gold users will be given higher priority than that of Silver and Bronze users.



(a) dumb pipe



(b) smart pipe

Figure 1. The role of wireless carriers in offering services.

With the advent of multimedia services, wireless carriers or wireless service providers will carry increased traffic. They are also faced with the threat of becoming dumb bit pipes and the challenge of lost revenue opportunity. As shown in Figure 1(a), value-added multimedia services are provided by application service providers (ASPs) outside of the wireless carrier's network, and wireless carrier's network is used only for transport. However, wireless carriers cannot afford to be just providers of transport. They have no choice but to offer novel value-added services retaining control and billing for QoS of these services within their own networks. As shown in Figure 1(b), by moving the control (including authentication with AAA/HLR, QoS management and billing) and application servers inside its network, the wireless carrier can provision QoS parameters for different classes of traffic and different classes of users (for example, Gold, Silver and Bronze). For services not authorized by the carrier, namely for those services provided by external ASPs, the

carrier can offer only best effort service and hence provide a demotivator for ASP services.

End-to-end QoS requires support at application layer, network layer, link layer, and medium access control (MAC) layer. This paper discusses existing CDMA2000 network architecture and shows its shortcomings that prevent support for QoS in Radio Access Network (RAN). Changes in RAN are then proposed within the standards that enable QoS support consisting of signaling in the control plane and resource allocation in the data plane.

The rest of the paper is organized as follows: Section 2 presents the QoS architecture in CDMA2000 and provides a high level description of a QoS enabling architecture. Section 3 discusses how SIP and SDP are used for session setup and QoS negotiation within the QoS enabling architecture. Section 4 discusses the network layer and link layer modifications that are required within current CDMA2000 architectures to support service differentiation between different classes of traffic. Section 5 discusses modifications to existing MAC layer scheduling algorithms at the base station to provide service differentiation between different classes of users as well as different classes of traffic. Finally, Section 6 concludes the paper.

2. ARCHITECTURE FOR QOS SUPPORT

2.1. Current architecture without QoS support

Let us briefly review the current CDMA2000 architecture.^{8,9} Figure 2 shows the components and the protocol stack at each of the components of this architecture. The Mobile Station (MS) consists of two components, Terminal Equipment (TE), for example a laptop, and Mobile Terminal (MT), for example a PCMCIA CDMA2000 card. The Radio Access Network (RAN) encompasses the part of the network from the MS to the Packet Data Serving Node (PDSN) and PDSN is the gateway between RAN and the IP network (Internet)*. A data connection with the network needs to be first established from the MS. This is achieved by setting up a PPP¹⁰ connection between the TE and the PDSN. Data packets are transported from the TE to the PDSN in the form of PPP frames. Packets from the Internet, destined to the TE and received at the PDSN are transported over the same PPP connection to the TE. Notice that traffic for all applications running on the TE goes through the same PPP connection between the TE and the PDSN.

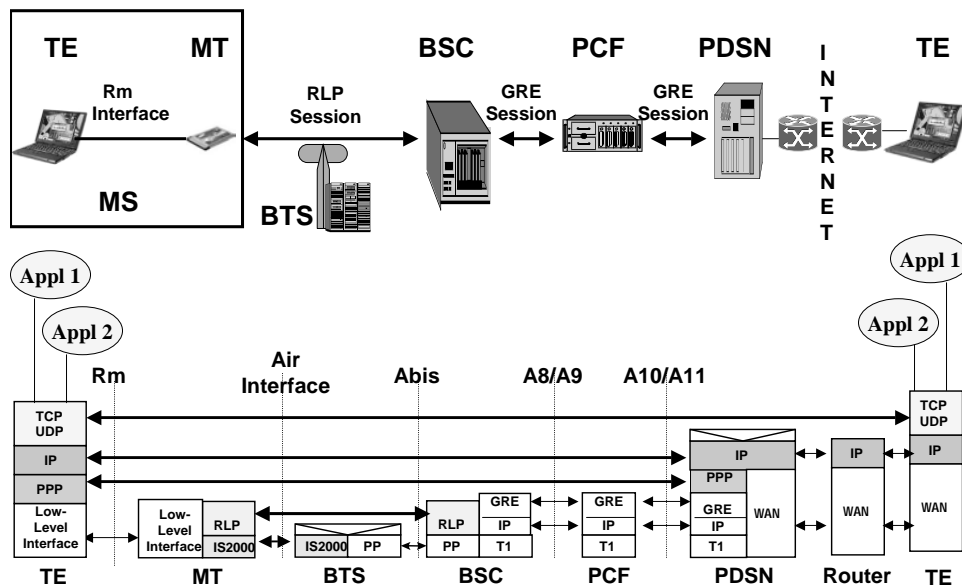


Figure 2. CDMA2000 Architecture

Between the TE and the MT, a serial link carries traffic in the form of a byte stream. The interface between the TE and the MT is the Rm interface.¹¹ The CDMA2000 Base Station (BS) consists of two components, Base Transceiver Station (BTS) and Base Station Controller (BSC). The CDMA2000 airlink is between the MT and the BTS both of which have

* According to CDMA2000 standards, PDSN connects RAN to the IP network, but it is not part of RAN.⁸

radio interfaces. Normally, multiple BTSs are connected to a BSC through point-to-point links. The interface between the BTS and the BSC is referred to as the Abis interface.¹² PPP frames received at the MT (in the reverse direction from TE to PDSN) and at the BSC (in the forward direction from PDSN to TE) are transported over a Radio Link Protocol (RLP) session to the BSC and MT respectively.¹³ There is one RLP session per PPP session. Data received at the MT are framed into RLP frames before they are sent to the BSC over the air interface. Data received at the BSC are framed into RLP frames before transported to the MT over the Abis interface. RLP allows for retransmission of RLP frames thereby hiding frame errors from the higher layer protocols.

There is another component between the BSC and the PDSN which is the Packet Control Function (PCF). The main functionality of the PCF is to direct PPP connection requests from the TE to the appropriate PDSN that should handle the TE. Data traffic between the BSC and the PCF are sent over Generic Routing Encapsulation (GRE)¹⁴ tunnels, one per PPP session. The GRE frames between the BSC and the PCF are normally transported over a private network (could be an IP network). Between the BSC and the PCF, A9 interface is used for signaling control messages and A8 interface is used to transport data packets.⁹

Data between the PCF and the PDSN are transported over another set of GRE tunnels, again one per PPP session. The network that connects the PCF with the PDSN could be the public IP network. Between the PCF and the PDSN, A11 interface is used for signaling control messages and A10 interface is used to carry data packets.⁹

End-to-end QoS requires modifications to the above architecture at the application layer, network layer, link layer and MAC layer. A cross-layer, end-to-end QoS architecture consisting of signaling in the control plane and resource allocation in the data plane is highlighted in Section 2.2 to support QoS. Each component of this architecture is discussed in detail in Sections 3, 4 and 5.

2.2. A high level view of a QoS enabling architecture

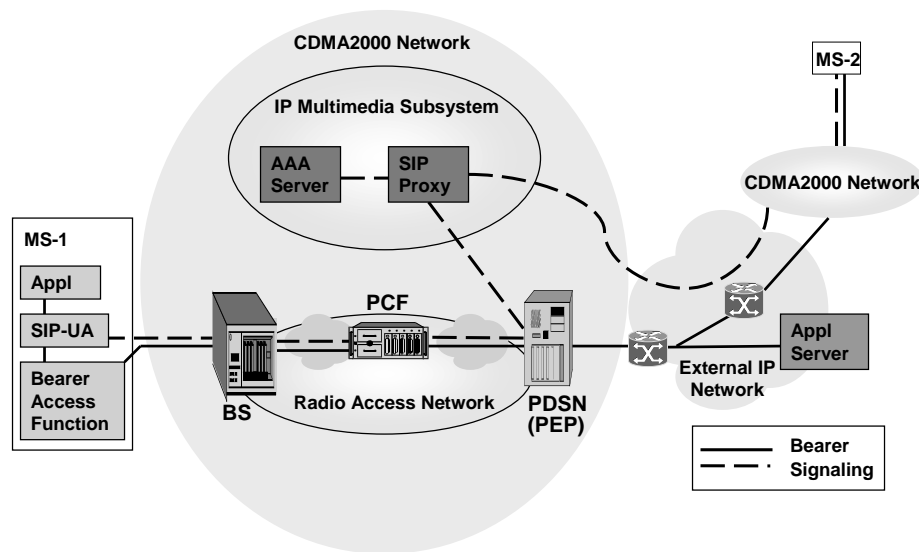


Figure 3. End-to-end QoS Architecture in CDMA2000 Networks based on IMS.

Figure 3 depicts the end-to-end QoS architecture in CDMA2000 networks based on IMS. In general there are two categories of connections, one between two MSs, the other between an MS and an application server.

In the first category, the end-to-end signaling between the initiating mobile station MS-1 and the corresponding mobile station MS-2 is conducted at the application layer using Session Initiation Protocol (SIP).¹⁵ The application in MS-1 specifies the QoS request and passes it to SIP User Agent (SIP-UA). Based on the request, SIP-UA initiates a negotiation between the SIP-UA in MS-2 through SIP Proxies[†] and AAA servers in MS-1's home network and MS-2's

[†]In IMS, Call Session Control Function (CSCF) is the node hosting the SIP proxy.² Figure 3 only shows QoS related entities in IMS. For the complete IMS architecture, see 3GPP specification.²

home network respectively. To simplify the discussion, the example in Figure 3 shows the case where both MS-1 and MS-2 reside in their respective home networks, and the QoS request specifies the QoS parameters for traffic in both directions between the two MSs. First, SIP-UA in MS-1 sends its QoS request to SIP proxy in its home network, SIP proxy checks with AAA server to see if this request is authorized for MS-1 based on the profile for the user currently using MS-1. If it is authorized, SIP proxy notifies the Policy Enforcement Point (PEP) within the PDSN. PEP will regulate traffic according to the QoS parameters that have been authorized for MS-1. After MS-1's SIP proxy validates MS-1's QoS request, it passes the request to the SIP proxy for MS-2. The SIP proxy for MS-2 checks with the AAA server for MS-2 to validate the QoS request, and passes the request to MS-2's SIP-UA. After receiving MS-1's request, MS-2's SIP-UA responds with its own QoS request taking into consideration MS-1's authorized request it has just received. MS-2's QoS request takes the reverse path from MS-2's SIP-UA to MS-1's SIP UA. After this two-way handshake, both MS-1 and MS-2 have agreed on the set of authorized QoS parameters to use.

In the second category where a MS is initiating a connection to an application server, SIP is only used to negotiate QoS parameters within the MS's wireless network through its SIP proxy and AAA server. Notice in both cases, the BS, PCF and PDSN all serve only as transport. Details of application layer signaling is discussed in Section 3.

After end-to-end QoS signaling has been conducted in application and network layer, access bearer QoS signaling needs to be conducted in the RAN. This is conducted in link layer through modifications to Abis interface between the BSC and the BTS, A9 signaling interface between the BSC and the PCF, and A11 signaling interface between the PCF and the PDSN. To support QoS, the current link layer and network layer architecture has to be modified to support multiple RLP sessions and GRE tunnels to transport packets on a single PPP session to and from a TE.¹⁶ The details of the architectural modifications along with their justification are presented in Section 4.

In the data plane, we assume QoS is offered through DiffServ in the external IP network, and in the IP network between the PCF and the PDSN within the RAN. Normally, the BSC and the PCF are connected using a private network which could be an IP network. If it is an IP network, QoS for data transported between the BSC and the PCF could be offered using DiffServ as well. Section 4 presents the details of how DiffServ is realized within the RAN. Because of the asymmetric nature of data traffic (more traffic flows from the Internet to the MS than in the opposite direction), scheduling of RLP frames at the BSC to be sent to the MT becomes an important consideration. Without QoS support such scheduling is based on Radio Frequency (RF) signal conditions of the radio channel. With QoS, to provide service differentiation, modifications are required to these scheduling algorithms. This issue is addressed in Section 5.

3. APPLICATION LAYER: SIP AND SDP

The IMS architecture uses Session Initiation Protocol (SIP)¹⁵ for session management and media and QoS negotiation. Session management covers initiation and termination of a SIP session using a three-way handshake with INVITE, OK and ACK messages. Media and QoS negotiation is conducted with UPDATE messages and by using Session Description Protocol (SDP)¹⁷ payload of SIP messages.¹⁸

SIP is an application layer control and signaling protocol for creating, modifying, and terminating multimedia sessions with one or more participants. SIP invitations (INVITE message) and other type of SIP messages carry session descriptions in SDP that allow participants to agree on a set of compatible media types and QoS requirements. SIP provides a registration function that allows users to upload their current locations with proxy servers, and proxy servers can help route requests to the user's current location. SIP runs on top of different transport protocols such as TCP or UDP.

Figure 4 displays a simple session setup and termination process between two SIP User Agents UA1 and UA2 involving their respective SIP proxy servers, Proxy1 and Proxy2. Proxy1 is the proxy server in charge of UA1's domain, and similarly, Proxy2 is in charge of UA2's domain. Before a session initiation takes place, we assume both SIP UAs have registered their current location with their respective SIP proxies. SIP proxies have also registered with some form of Domain Name Server (DNS) so that through inquiries to the DNS, any other SIP proxy can route the request for UA1 to Proxy1 and route the request for UA2 to Proxy2.

SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request and at least one response. When UA1 wants to make a connection to UA2, it sends a INVITE request to Proxy1. Proxy1 replies with a "100 Trying" response signifying Proxy1 is working on finding UA2. After consulting the DNS with the domain of UA2, Proxy1 routes the request to Proxy2 which is in charge of UA2's domain and has the current location of UA2.

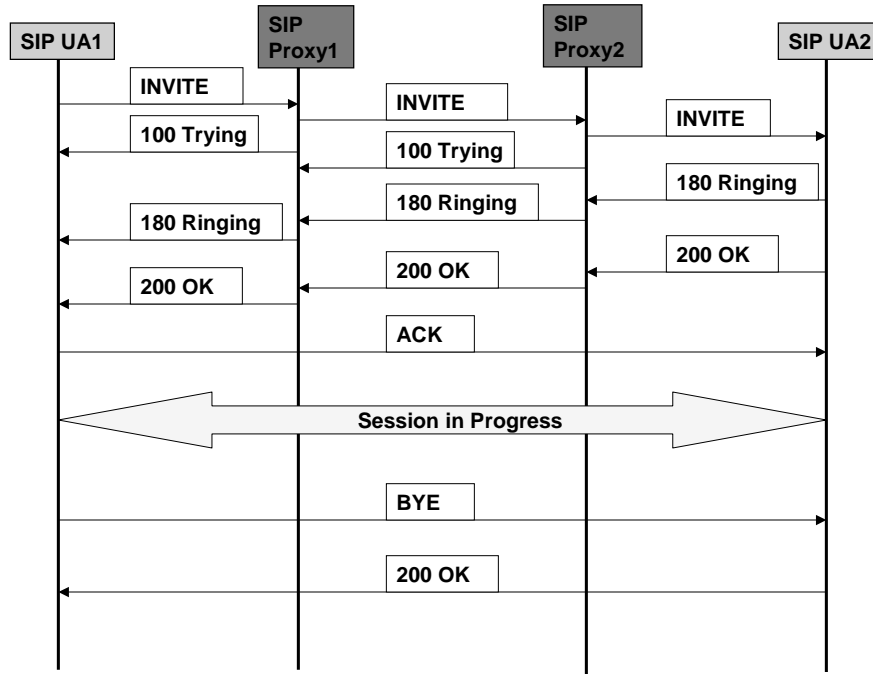


Figure 4. Session setup and termination using SIP.

Proxy2 in turn forwards the request to UA2 and returns a “100 Trying” response to Proxy1. UA2 returns with a “180 Ringing” response reporting UA2 is being ringed, and this response is forwarded to Proxy2, Proxy1, and eventually to UA1. After UA2 decides to accept the call, it sends a “200 OK” response to UA1 through Proxy2 and Proxy1. This response contains the IP address of UA2’s current location, so UA2 can be directly contacted from now on. After receiving the “200 OK” response, UA1 sends an ACK message directly to UA2, bypassing the SIP proxies, and finishes the three-way handshake. A session is now established, and data traffic flows directly between UA1 and UA2. Either UA can terminate a session. In this example, UA1 sends a BYE request to terminate the session, and UA2 replies with a “200 OK” response.

The basic session setup process can be extended with QoS negotiations using SDP and UPDATE request. Figure 5 presents a typical example of how a session with QoS is setup between SIP UAs through their respective SIP proxies in IMS.^{2,18} Assume user1 is currently using the TE where UA1 resides and user2 is using the TE where UA2 resides. As shown in the IMS architecture in Figure 3, each SIP proxy is associated with a AAA server at which user profiles are stored.

First the initiator UA1 sends INVITE request towards UA2 to initiate a session. In addition, this request also contains UA1’s QoS proposal in *SDP1*. After Proxy1 receives this INVITE request, it checks with its AAA server to see whether user1 is authorized to use the QoS parameters specified in *SDP1*, and modifies *SDP1* into *SDP1'* if needed to reflect the authorized QoS parameters. Then Proxy1 routes the INVITE request with *SDP1'* to Proxy2. Proxy2 in turn checks with its associated AAA server to see if *SDP1'* is authorized for the user2, and based on the results from AAA, modifies *SDP1'* into *SDP1''* if needed before passing the INVITE request to UA2.

After receiving UA1’s modified proposal in *SDP1''*, UA2 replies with a response “183 Session Progress” with its QoS proposal in *SDP2*. This response is processed by Proxy2 followed by Proxy1 before being routed to UA1.

After this round of offer-answer process,¹⁹ both UA1 and UA2 agree on *SDP2* to be the set of authorized QoS parameters. And a session can be established. However, another round of offer-answer process can happen before a session is established if either UA wants to change QoS parameters. This is realized by another round of offer-answer process using the UPDATE message.²⁰ In this example, UA1 sends an UPDATE request with *SDP3* towards UA2. After Proxy1 receives the request, it checks with its AAA server and modifies *SDP3* into *SDP3'* before routing the UPDATE

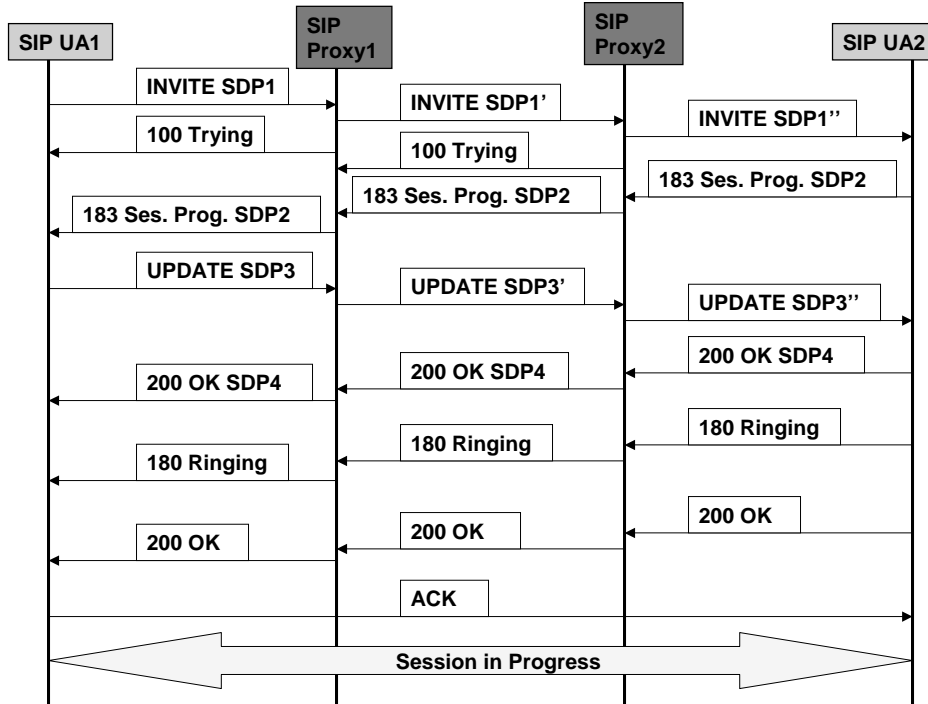


Figure 5. Session setup and QoS negotiation using SIP and SDP.

request to Proxy2. Proxy2 in turn checks with its AAA server and modifies *SDP3'* into *SDP3''* before forwarding the request to UA2. UA2 responds with a “200 OK” reply containing *SDP4* specifying the final QoS parameters agreed between UA1 and UA2.

At this point, session establishment resumes and UA2 sends a “180 Ringing” response back to UA1. After UA2 decides to accept the call, it sends UA1 a “200 OK” response to the INVITE request, and UA1 replies with a ACK message finishing the 3-way handshake, after which the session starts.

Notice that we have assumed SIP proxies are allowed to modify SDPs in SIP message bodies in QoS negotiation process. However, IETF RFC 3261¹⁵ explicitly disallows SIP proxies from manipulating the content of SIP message bodies mainly because of the end-to-end security feature offered by SIP. If end-to-end authentication is used, SIP proxies can not alter the SDP. If end-to-end encryption is used, SIP proxies will not even be able to look at the SDP to modify it. In order to overcome the restriction of SIP proxy, recently there have been several IETF drafts proposing new SIP headers to carry QoS information regarding the session.^{21,22} Therefore, to be RFC 3261 compliant, SIP proxies can change SIP headers during the QoS negotiation phase instead of modifying SDPs.

The QoS negotiation process shown in Figure 5 can be illustrated further in a detailed scenario. For simplicity, we assume user profile specifies the same bandwidth in both send and receive directions for a user. Further we assume user1 has 5 Mbps bandwidth in its user profile, and user2 has 1 Mbps bandwidth in its user profile.

In this scenario, UA1 sends INVITE with *SDP1* specifying 10 Mbps bandwidth requirement and two supported codecs *A* and *B*. The bandwidth requirement for codec *A* is higher than 1 Mbps, and for codec *B*, lower than 1 Mbps. After Proxy1 receives this INVITE, it checks with its AAA server and discovers user1 is only allowed 5 Mbps bandwidth, it modifies *SDP1* into *SDP1'* specifying 5 Mbps bandwidth requirement and passes the INVITE to Proxy2. After Proxy2 receives the INVITE with *SDP1'* as its payload, it checks with its AAA server and discovers user2 is only allowed 1 Mbps bandwidth. Proxy2 then constructs *SDP1''* specifying 1 Mbps bandwidth requirement before putting in on the INVITE request to UA2. So the INVITE request UA2 receives contains *SDP1''* specifying UA1 is requesting 1 Mbps bandwidth and two codecs. UA2 agrees with this QoS proposal, and copies *SDP1''* to *SDP2* and puts *SDP2* on the “183

Session Progress” reply message to UA1, this messages is routed through Proxy2 and Proxy1 with *SDP2* intact because all the QoS parameters have already been authorized by AAA servers.

After UA1 receives *SDP2*, it knows the QoS parameters authorized by AAA servers and agreed on by UA2. However, since *SDP2* only specifies 1 Mbps bandwidth, UA1 could not use codec *A* because of bandwidth constraints. As a result, UA1 decides to drop *A* from its supported codec list, and leave only one supported codec *B*. It sends an UPDATE request with *SDP3* specifying 1 Mbps bandwidth and one supported codec *B*. After Proxy1 receives this UPDATE, it checks with its AAA server and discovers user1 is allowed 5 Mbps bandwidth, which is more than what user1 requests for. Therefore it sets *SDP3'* to be the same as *SDP3*, and passes the UPDATE request to Proxy2. After Proxy2 receives the UPDATE with *SDP3'* as its payload, it checks with its AAA server and discovers user2 is allowed 1 Mbps bandwidth. Proxy2 then sets *SDP3''* to *SDP3'* and puts it on the UPDATE request to UA2. So the UPDATE request UA2 receives contains *SDP3''* specifying UA1 is requesting 1 Mbps bandwidth and one codec *B*. UA2 agrees with this QoS proposal, and copies *SDP3''* to *SDP4* and puts *SDP4* in the “200 OK” reply message to UA1, this messages is routed through Proxy2 and Proxy1 with *SDP4* intact because all the QoS parameters have already been authorized by AAA servers. After this point, the QoS negotiation phase finishes, and the rest of the session establishment process proceeds.

4. NETWORK AND LINK LAYERS

4.1. Data flow without service differentiation

To revisit the data flow within the current CDMA2000 architecture, let us refer back to Figure 2. In the reverse direction from the TE to the PDSN, PPP frames are transported as a byte stream over the Rm interface between the TE and the MT. From the MT to the BSC, the byte stream is segmented into RLP frames and carried over a single RLP session through the BTS (radio terminal) over the Air Interface and the Abis Interface to the BSC. At the BSC, data traffic becomes a byte stream again and is segmented into multiple GRE frames and tunneled over the established GRE tunnel to the PCF over the A8 data interface. At the PCF, these GRE frames are forwarded to the PDSN using the corresponding GRE tunnel over the A10 data interface. In the forward direction from the PDSN to the TE, IP packets received at the PDSN are sent over the corresponding GRE tunnel to the PCF which in turn forwards the GRE frames over the corresponding GRE tunnel to the BSC. At the BSC, data is converted to a byte stream which is then sent as RLP frames to the MT. The MT then sends the data as a byte stream over the serial link to the TE. Normally, GRE frames between the BSC and the PCF would be carried over a private network (possibly an IP network) and between the PCF and the PDSN over a public IP network.

There are two main points to be noted here. First, a single PPP session carries all data traffic in the form of PPP frames from the TE to the PDSN and from the PDSN to the TE. These PPP frames are transported over the same RLP session between the TE and the BSC and over the same GRE tunnel between the BSC and the PCF and between the PCF and the PDSN. Thus, there is one-to-one mapping between a PPP session, an RLP session, a GRE tunnel between the BSC and PCF and a GRE tunnel between the PCF and the PDSN. Second, when PPP data frames are sent from the TE as a byte stream to the MT, PPP frame boundary information is lost. This means, a PPP frame can span multiple RLP frames and GRE frames; RLP and GRE frames may contain full or parts of multiple PPP frames as well. In the reverse direction, in order to enable regeneration of PPP frames at the PDSN, HDLC framing is performed before PPP frames are sent from the TE.²³ In the forward direction, when IP packets are received at the PDSN and are put into PPP frames to be sent to the TE, again HDLC framing is performed.

4.2. Impediments to providing service differentiation with current architecture

When no QoS differentiation is required on traffic to and from a single user, the above session establishment procedure of mapping all data to and from a TE onto a single RLP session and then onto single GRE sessions is adequate. However, when QoS differentiation is required, this procedure does not provide the capability to handle different classes of traffic for the same user differently.

To illustrate this difficulty, let us first consider traffic in the reverse direction from the TE to the PDSN. When PPP frames are sent from the TE, class based scheduling, for example, based on TCP and IP information, could be performed on the PPP frames. But once the PPP frames are converted to a byte stream and sent over the Rm interface and then as RLP frames to the BSC, these RLP frames are delivered at the BSC in sequence. This is ensured by sequence numbers on the RLP frames. Further, when the byte stream is framed into GRE frames and sent over the GRE tunnels between

the BSC and the PCF and then between the PCF and the PDSN, these GRE frames are also delivered in sequence at the delivery points. Consider a mix of real-time and non-real-time traffic being sent over the same RLP session.

- Firstly, it is possible that real-time traffic and non-real-time traffic form part of the same RLP frame. This means there is no way to handle these two classes of traffic differently once they leave the TE.
- Secondly, each RLP session has parameters such as number of retransmissions that applies to all RLP frames in the session. As a result, even though retransmissions may not make sense for real-time traffic, there is no way to turn it off just for that class of traffic.
- Thirdly, because of ordered delivery of RLP, if a RLP frame that carries non-real-time traffic is lost, real-time RLP frames that follow the lost frame cannot be delivered to higher layers even if they are received at the BSC. The delivery of these real-time RLP frames can only happen after the lost RLP frame is retransmitted and is subsequently received at the BSC.

Similar to the implication of ordered delivery of RLP frames, because GRE frames are also delivered in sequence, there is no way to provide service differentiation within the IP networks that carry traffic between the BSC and the PCF and between the PCF and the PDSN.

Traffic in the forward direction from the PDSN to the TE faces a similar problem. When IP packets from the Internet are received at the PDSN, some type of class-based scheduling can be performed at the PDSN before they are sent over the PPP session. But once this is done, there is no flexibility to provide differentiated services within the IP network until the frames reach the BSC. This is because all classes of traffic is carried over the same GRE tunnel between the PDSN and the PCF and the same GRE tunnel between the PCF and the BSC. Furthermore, at the BSC, there is no way to differentiate the traffic and hence class-based scheduling cannot be implemented.

4.3. CDMA2000 service architecture with multiple classes of service

In order to get around the problems illustrated in Section 4.2, a service architecture to support service differentiation will use multiple RLP sessions and multiple GRE tunnels to carry traffic to and from a TE.¹⁶ Following the standards, we assume there is a one-to-one correspondence between a TE and a user, and use TE and user interchangeably. There is still a single PPP session carrying traffic from the TE to the PDSN and from the PDSN to the TE. This means there is a single PPP state machine per user at the TE and at the PDSN. But different PPP frames on this PPP session, each carrying a different class of traffic, will go over different *service instances* between the MT and the BSC.⁸ Each service instance is implemented by a separate RLP session. Between the BSC and the PCF and between the PCF and the PDSN, there are multiple GRE sessions per TE, each of which carry a different class of traffic.

Once each single class of traffic is carried in a separate RLP session, RLP sessions can be used to prioritize service classes between the MT and the BSC. For traffic from the BSC to the MT, class-based scheduling can be used at the BSC to give priority to RLP frames on specific RLP sessions. For traffic in the other direction, the MT can schedule traffic on different RLP sessions. In addition, RLP session parameters can be configured to match service class for the traffic. For example, retransmission can be turned off for a RLP session carrying real-time traffic, and turned on for a RLP session carrying non-real-time traffic.

Similarly, different GRE sessions are used to handle different classes of traffic. GRE frames that carry different classes of traffic can be handled differently because they are transported over different GRE sessions. For example, IP packets that encapsulate GRE frames that belong to different GRE tunnels (and hence different service classes) can be marked with DiffServ markings²⁴ to specify the class of traffic that they carry. This way, if a private IP network is used between the BSC and the PCF, intermediate routers can handle different classes differently. Similarly, routers on the IP network between PCF and PDSN, which is normally the public IP network, can also handle different classes of traffic differently.

It should be noted that even with this architecture, service differentiation using IntServ²⁵ is not generally possible in the RAN between the BSC and the PDSN. IntServ requires per flow state to be maintained at each of the routers. This means that the inner IP and TCP/UDP headers contained within the PPP frames need to be visible within the RAN. However, IP header compression as well as PPP compression are normally applied at the end-points, TE on one side and

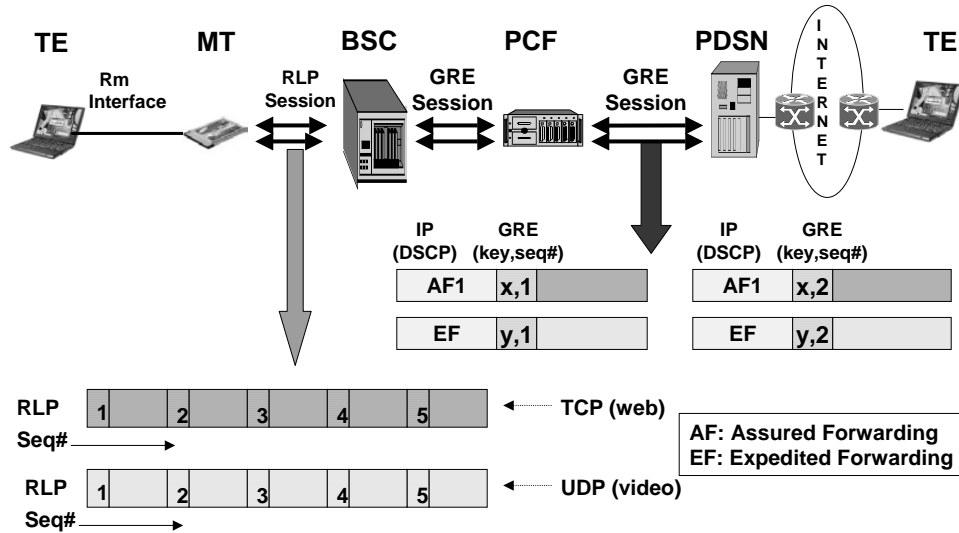


Figure 6. Multiple RLP and GRE sessions to support service differentiation.

PDSN on the other side. The compression makes it impossible for flow information to be made visible at intermediate routers.

Figure 6 illustrates the service architecture that supports service differentiation. Two classes of traffic, video streaming and web browsing, to the same TE are carried over different RLP sessions. For the video traffic, the RLP retransmission could be turned off. Over the IP network between the BSC and the PCF and between the PCF and the PDSN, two classes of traffic are sent on different GRE tunnels and different DiffServ code points (DSCP) are used to handle the two traffic classes differently. The web traffic is sent under the Assured Forwarding Mode and the video traffic is sent under the Expedited Forwarding Mode.²⁴ Notice that unique GRE keys between the two endpoints are used to identify GRE tunnels.¹⁴ In the example shown in Figure 6, GRE key x is used to identify web traffic and y is used to identify video traffic.

Besides multiple RLP sessions and multiple GRE sessions, to separate different classes of traffic to offer QoS, there are some issues with the Rm interface that need consideration. The Rm interface can be implemented under two models, the *Relay Model* and the *Network Model*. Under the Relay Model, the end-points of the PPP session are the TE and the PDSN and the MT simply forwards the byte stream received from the TE by segmenting it into RLP frames. Under the Network Model, the PPP session from the TE is terminated at the MT. Another PPP session is used to carry traffic from the MT to the PDSN.

Under these two models, there are two issues to be considered. Firstly, traffic that belongs to different service classes will still be sent over the Rm interface on the serial link between the TE and the MT as a byte stream. The traffic need to be de-multiplexed onto different RLP sessions based on service classes. With Network Model, this is not a problem. As the PPP session from the TE is terminated at the MT, and the MT can inspect the TCP/IP headers of the data packet inside a PPP frame to identify the service class and send the PPP frame on the appropriate RLP session. With Relay Model, the PPP session is not terminated at the MT. This implies solutions need to be found so that a single byte stream from the TE can be de-multiplexed onto multiple byte streams based on service classes and sent over separate RLP sessions. Some form of framing and de-framing is needed between the TE and the MT.

Secondly, even with the above service architecture supporting service differentiation, a single PPP session is still used to carry traffic from the TE to the PDSN and vice-versa. Because traffic from a single PPP session is split into multiple byte streams and prioritized, PPP frames may arrive out of order at PPP end-points. This means, stateful PPP compression will not work if it is performed on the PPP stream. Only stateless compression can be performed, if it is to be performed at the PPP endpoints. If stateful compression is required, they need to be performed on the PPP frames on a per link basis after the traffic is split into multiple byte streams at the MT.

4.4. End-to-end service differentiation

The above discussion illustrates how service differentiation can be provided within the RAN. Outside of the RAN on the Internet, service differentiation can be provided either using IntServ or DiffServ. Such service differentiation has to be consistent with that provided within the RAN. This means, at the PDSN, traffic from the Internet as well traffic towards the Internet needs to be classified appropriately. Consider IP packets from the Internet received at the PDSN. The PDSN should have enough policy information to map IP packets onto the appropriate GRE tunnel towards the TE irrespective of whether service differentiation is provided outside of the RAN or not. Similarly, if service differentiation is to be provided outside of the RAN (say, using DiffServ), the PDSN should have the required policy information to mark IP packets arriving from the RAN on different GRE tunnels with the appropriate DiffServ markings before they are sent on the Internet.

In Section 4.3, it is pointed out that IntServ cannot be used to differentiate traffic within the RAN. But RSVP signaling that is required to download IntServ policy onto the routers can be used to serve another purpose. RSVP can be used to download policy information from the TE onto the PDSN so that packets received from the Internet can be classified and sent on appropriate GRE tunnels, and hence appropriate RLP sessions, towards the TE. In addition, packets received from the RAN can be appropriately DiffServ marked before they are sent on the Internet.

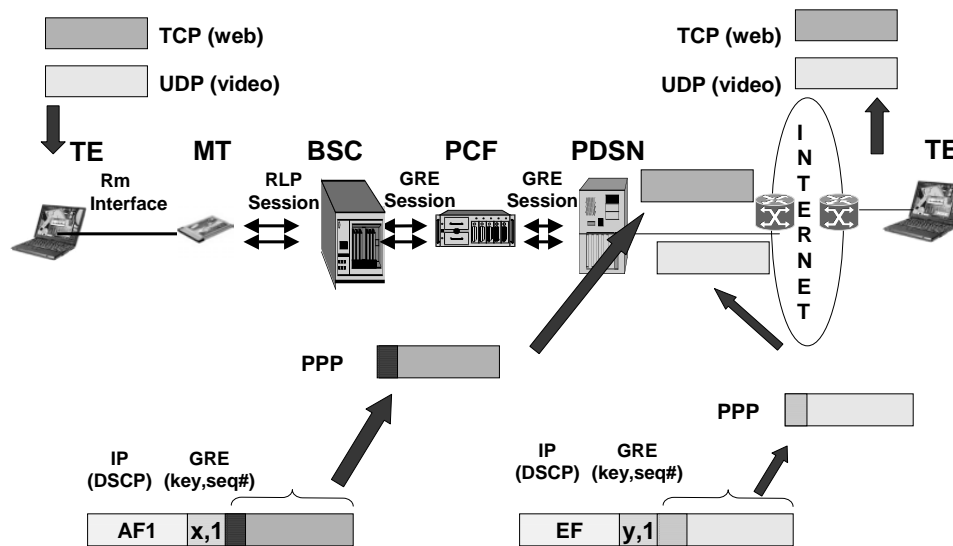


Figure 7. Traffic classification at the PDSN.

Figure 7 illustrates the traffic classification procedure at the PDSN. For this example, again, we consider two classes of traffic, video streaming and web browsing. In addition, assume that no service differentiation is provided outside of the RAN. Packets arrive at the PDSN from the Internet without any DiffServ marking. By looking at the TCP/UDP and IP headers on the arriving packets, the PDSN is able to frame these IP packets within PPP and mask the (outside) IP header that encapsulates the GRE frame with the appropriate DiffServ marking to be used within the RAN. Afterwards, the PDSN is able to forward the web and video traffic onto different GRE tunnels (one with key x and the other with key y) where the video traffic is sent using the Expedited Forwarding Mode and the web traffic is sent using the Assured Forwarding Mode. The PDSN requires policy information to perform this function and as indicated above, RSVP may be used to download this information at the PDSN. Figure 7 shows each TCP/IP packet arriving from the Internet being encapsulated within one PPP frame. But TCP/IP packets could be fragmented and put in multiple PPP frames if the arriving packet is larger than the Maximum Receive Unit (MRU)¹⁰ at the receiver, which in this case is the TE.

5. MAC LAYER SCHEDULING

In Section 4, we have discussed how DiffServ could be used to provide differentiated service between the BSC and the PDSN (in both directions) by handling packets on different GRE tunnels differently. We have also discussed how traffic

from the TE to the BSC that are sent on different RLP sessions could be scheduled at the TE appropriately to provide differentiation. But given the asymmetric nature of data traffic and the wireless link capacity, one of the important issues to investigate is the scheduling of RLP frames at the BSC that are sent towards the TE. When data for different users arrive at the BSC from the PCF they are segmented into RLP frames and are queued to be sent over the corresponding RLP sessions. In a CDMA2000 1xRTT system, simultaneous transmission to multiple users is possible.²⁶ But in a CDMA2000 EvDO system, time slots are assigned to different users and scheduling at the BSC decides which user should get the next time slot to send out data.²⁷ This problem is illustrated in Figure 8. With QoS, this issue becomes complicated by the fact that the scheduling algorithm should not only decide which user should get the next slot, but also which traffic class for that user should get that slot. This is the focus of discussion in this section.

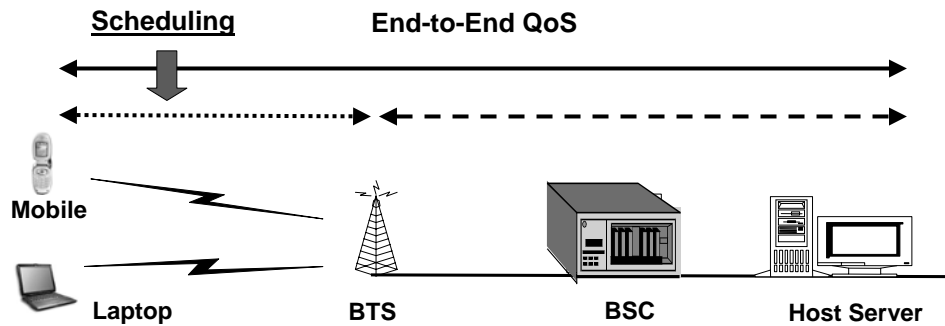


Figure 8. Scheduling at the BTS.

5.1. CDMA2000 EvDO scheduling without service differentiation

In a CDMA2000 EvDO system, scheduling of RLP frames is required even without QoS considerations, due to the time slotted nature of transmission. There are two main considerations: a) the scheduling algorithm should be such that the wireless channel throughput is maximized, and b) scheduling should be such that the users are treated fairly.

Currently, CDMA2000 EvDO algorithms at the BSC works as follows. Each user provides continuous feedback to the base station about its current rate of reception based on Carrier to Interference Ratio (C/I). The feedback is carried in the form of Data Rate Control (DRC) parameter. This provides the BSC with a measure of the quality of the channel as perceived by different users. Based on the DRC, the BSC decides which user's data should be sent on the next time slot. This is shown in Figure 9. If the only goal is to maximize the throughput on the wireless channel, the strategy would be to always schedule the next slot to the user with the best DRC. This would mean that users whose reception quality is poor would be starved. To be fair also to different users, the goal is constrained by the requirement that each user is allocated equal time (or the same number of time slots) over a period of time.

In order to maximize the overall throughput within the above mentioned fairness requirement, the scheduling algorithm used is based on weight $W_i = DRC_i/R_i$ where DRC_i is the DRC value for user i and R_i is the exponentially averaged MAC throughput received by user i so far. At every time slot, the user with the maximum value W is scheduled to receive data on that slot. This algorithm is a Proportional Fair (PF) Scheduling Algorithm.²⁸ By definition, a PF algorithm maximizes the sum of the logarithm of the individual user throughput. Note that such a scheduling algorithm provides higher throughput to users with better RF conditions at the same time providing a measure of fairness by giving equal number of time slots to all users over a period of time. Scheduling based on an equal-time scheme is a better fairness criteria among users. In contrast, an algorithm based on an equal-throughput criteria will force the overall sector throughput to be close to the worst individual throughput multiplied by the number of the supportable users.

5.2. CDMA2000 EvDO scheduling with service differentiation

The scheduling algorithm discussed in Section 5.1 needs to be modified to support service differentiation. This can be done at two levels.

Firstly, service differentiation can be provided between users by introducing a *fudge factor* (F) so that the weight W is scaled to prefer premium users (for instance, Gold members). For example, instead of computing the weight for

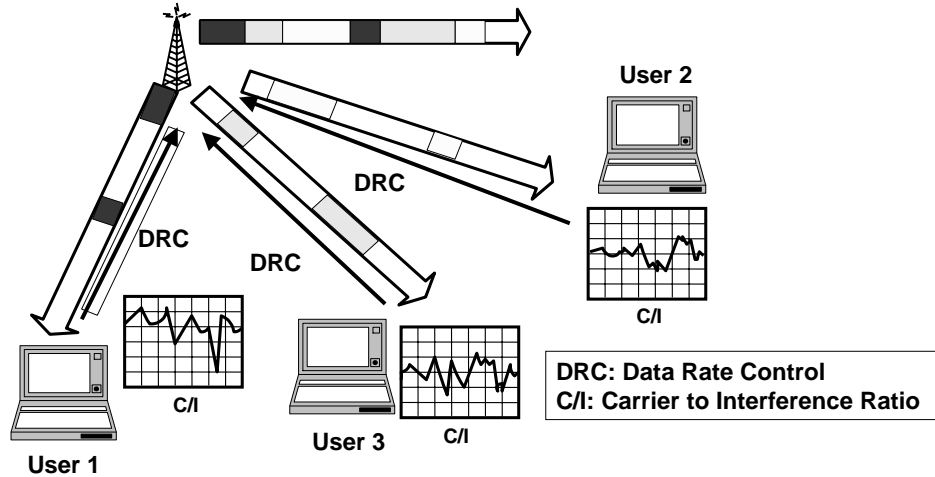


Figure 9. DRC based scheduling for multiple users.

user i as $W_i = DRC_i/R_i$, for a premium user i , it could be scaled up to be $F \times DRC_i/R_i$ where $F > 1$. F can be a function of subscription level of the premium user.

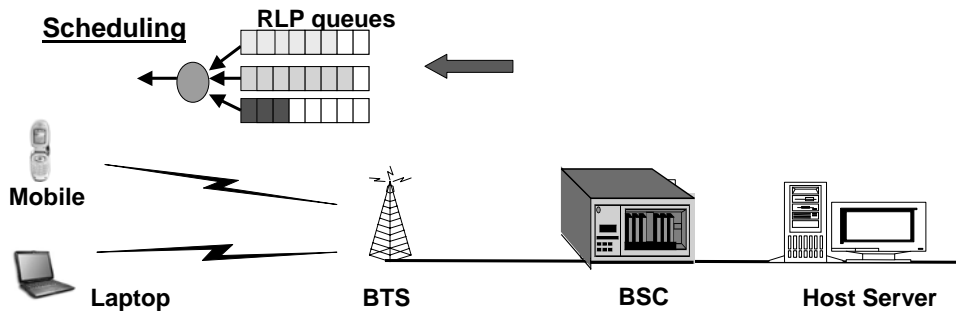


Figure 10. RLP Queues for one user with different classes of traffic, where each RLP queue carries a separate class of traffic.

Secondly, once the identity of the user to receive data during the next slot is determined using a modified PF algorithm (such as the one suggested above to differentiate between users), the class of traffic that should use this slot should be determined as well. This will provide service differentiation among traffic destined to the same user. When data to be sent to a user is received at the BSC on multiple GRE tunnels (each carrying one class of traffic), they will be segmented into RLP frames and should be queued in different queues as shown in Figure 10. Once this is done, at each slot provided to the user, a RLP frame from one of the queues can be selected based on some priority-based algorithm where the priority can be based on a number of factors including delay and jitter requirements.

Thus providing service differentiation between users as well as different classes of traffic destined to the same user requires a *hierarchical* scheduling algorithm to be implemented at the BSC.

6. CONCLUSION

In this paper, we discuss a CDMA2000 standards compliant architecture that supports service differentiation within the RAN. In particular, we examine the drawbacks of existing CDMA2000 network architecture that prevent supporting multiple classes of traffic. We propose changes in RAN within the standards that enable support for multiple traffic classes. In addition, we discuss how SIP can be augmented with QoS signaling for supporting end-to-end QoS. We also review base station scheduling algorithms and provide possible extensions to these algorithms to support different classes of traffic and different classes of users.

REFERENCES

1. 3rd Generation Partnership Project. <http://www.3gpp.org>.
2. 3rd Generation Partnership Project, "IP Multimedia Subsystem (IMS); Stage 2 (Release 5)." <http://www.3gpp.org>, Dec 2002. TS 23.228 v5.7.0.
3. 3rd Generation Partnership Project 2, "Developing the Next Generation CDMA 2000 Wireless Communications." <http://www.3gpp2.org>.
4. Real Networks, "RealPlayer." <http://www.realnetworks.com>.
5. Apple Computers, "Apple QuickTime Player." <http://www.apple.com/quicktime>.
6. Microsoft Corp., "Microsoft Windows Media Player." <http://www.microsoft.com>.
7. G. Armitage, "Lag over 150 Milliseconds is Unacceptable," May 2001. URL: http://www.geocities.com/gj_armitage/things/quake3-latency-051701.html.
8. 3rd Generation Partnership Project 2, "Wireless IP Network Standard ." <http://www.3gpp2.org>, Oct 2002. 3GPP2 P.S0001.
9. 3rd Generation Partnership Project 2, "Interoperability Specification (IOS) for CDMA 2000 Access Network Interfaces." <http://www.3gpp2.org>, May 2002. 3GPP2 A.S0001.
10. W. Simpson, "The Point-to-Point Protocol (PPP)," *RFC-1661* , July 1994.
11. 3rd Generation Partnership Project 2, "Rm interface ." <http://www.3gpp2.org>, 2003.
12. 3rd Generation Partnership Project 2, "BTS-BSC Inter-operability (Abis Interface)." <http://www.3gpp2.org>, Jul 2001. 3GPP2 A.S0003.
13. 3rd Generation Partnership Project 2, "Data Service Options for Spread Spectrum Systems - Addendum 2." <http://www.3gpp2.org>, Aug 2000. 3GPP2 C.S0017-0-2 v2.0.
14. S. Hanks, T. Li, D. Farinacci, and P. Tranina, "Generic Routing Encapsulation," *RFC-1701* , Oct 1994.
15. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," *RFC-3261* , Jun 2002.
16. 3rd Generation Partnership Project 2, "Quality of Service, Stage 1 Requirements." <http://www.3gpp2.org>, Sept 2002. 3GPP2 S.R0035.
17. M. Handley and V. Jacobson, "SDP: Session Description Protocol," *RFC-2327* , Apr 1998.
18. E. G. Camarillo, E. W. Marshall, and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)," *RFC-3312* , Oct 2002.
19. J. Rosenberg and H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP) ," *RFC-3264* , Jun 2002.
20. J. Rosenberg, "The Session Initiation Protocol (SIP) UPDATE Method," *RFC-3311* , Sept 2002.
21. J. Rosenberg, "Requirements for Session Policy for the Session Initiation Protocol (SIP)," *IETF Internet Draft*, <http://www.ietf.org/internet-drafts/draft-ietf-sipping-session-policy-req-00.txt> , June 2003.
22. V. Hilt and J. Rosenberg, "Supporting Intermediate Session Policies in SIP," *Work In Progress, IETF Internet Draft* , Oct 2003.
23. W. Simpson, "PPP in HDLC-like Framing." RFC-1662, Jul 1994.
24. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *RFC-2475* , Dec 1998.
25. R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," *RFC-1633* , June 1994.
26. 3rd Generation Partnership Project 2, "Interoperability Specification (IOS) for CDMA 2000 Access Network Interfaces – Part 1 Overview." <http://www.3gpp2.org>, May 2002. 3GPP2 A.S0011-0 v2.0.
27. 3rd Generation Partnership Project 2, "Inter-Operability Specification (IOS) for High rate Packet Data (HRPD) Access Network Interfaces ." <http://www.3gpp2.org>, Nov 2001. 3GPP2 A.S0007-0 v2.0.
28. P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users," *IEEE Communications Magazine* , pp. 70–77, Jul 2000.