DYNAMIC BANDWIDTH MANAGEMENT FOR TCP FLOWS IN A DIFFSERV-ENABLED MOBILE WIRELESS ACCESS NETWORK

by

Li Huang
B. Eng., Huazhong University of Science and Technology,
China, 1991

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2005

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

Li Huang

Master of Applied Science, 2005

Department of Electrical and Computer Engineering,

Ryerson University

With the soaring demand to provide global mobility for wide range of services, a growing trend for wireless access networks is to support multiple radio technologies that can be achieved efficiently by using TCP/IP protocols in the access network. In the thesis, we consider a mobile wireless access network where DiffServ is deployed as the QoS solution and Mobile IP is employed as the handover protocol. We first conducted a study on the impact of handover on DiffServ flows. Then we introduced a transient service level for handover flows and propose our QoS scheme and admission control algorithms for handover flows, which protect local flows from losing bandwidth to handover flows by separating the two flows into different service classes. We also proposed a service upgrade algorithm to upgrade the service level of handover flows based on the dynamic information of bandwidth utilization and different service upgrade priorities. To guarantee the proper provisioning for each service class, we proposed a dynamic bandwidth-provisioning algorithm that allows dynamic adjustment of bandwidth allocations to different service classes by adjusting their respective weights configured at the scheduler. We evaluated the feasibility of our QoS scheme and algorithm by simulating different handover situations and results show that the proposed scheme is viable under variety of provisioning scenarios.

Acknowledgements

I would like to thank my advisor, Dr. Muhammud Jaseemuddin, for supporting me throughout my time at Ryerson University. His hard working and seriously researching attitude set up role model for all his students including me. I always appreciate his cheerful encouragement during my studies. He always inspired me to pursue active ideas in our research area. He also makes chance for the members of the WINCORE lab to discuss, exchange ideas and learn from each other so that our knowledge is not only spot on the research area we each focusing on but also broad and include a big picture of relative areas. I am particularly grateful for his insightful comments on my work and helping me develop many needed soft skills.

I would also like to thank the members to my defense committee including Professor Alagan Anpalagan, Professor Bobby Ma for kindly taking time out from their busy schedules to serve.

I am so grateful to Ganesha Bhaskara who is now a graduate assistant in the Network Design and Test Lab of Department of Electrical Engineering of University of South California. He gave me many good suggestions on how to start a simulation on NS2. I am also grateful to all the members of our WINCORE lab for their friendly encouragement during my hard work on the research.

This thesis is dedicated to my parents, my husband and two sons for their love, support, and understanding, without which this thesis would not have been possible.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AF: Assured Forwarding

ATM: Asynchronous transfer Mode

BA: Behavior Aggregate

BE: Best Effort

BER: Bit Error Rate

BU: Binding Update

CBS: Committed Burst Size

CBR: Constant Bit Rate

CIR: Committed Information Rate

CN: Correspondent Node

CoA: Care of Address

CTR: Committed Target Rate

DiffServ: Differentiated Services

DSCP: DiffServ Code Points

EBS: Excess Burst Size

EF: Expedited Forwarding

FA: Foreign Agent

FIFO: First In First Out

GRE: Generic Routing Encapsulation

HA: Home Agent

ICMP: Internet Control Message Protocol

IRDP: ICMP Router Discovery Protocol

ISP: Internet Service Provider

IntServ: Integrated Services

LAN: Local Area Network

MAST: Multiple Average Single Threshold

MAMT: Multiple Average Multiple Threshold

MF: Multi Field

MN: Mobile Node

MRED: Multiple Random Early Detect

PBS: Peak Burst Size

PHB: Per Hop Behavior

PIR: Peak Information Rate

PRI: Priority Queuing

PTR: Peak Target Rate

QoS: Quality of Service

RED: Random Early Detect

RER: Radio Edge Router

RIO: Random Early Detection with In and Out

RIO-C: RIO Coupled Mode

RIO-D: RIO Decoupled Mode

RR: Round Robin

RTT: Round Trip Time

SAST: Single Average Single Threshold

SAMT: Single Average Multiple Threshold

SLA: Service Level Agreement

TCA: Traffic Conditioning Agreement

TCP: Transmission control Protocol

UDP: User Datagram Protocol

UMTS: Universal Mobile Telecom. System

WIRR: Weighted Interleaved Round Robin

WFQ: Weighted Fair Queuing

WRED: Weighted Random Early Detect

WRR: Weighted Round Robin

Chapter 1
Introduction

1.1 Overview

Increasing demand for bringing high-speed data and multimedia applications to mobile users pushes Third-generation (3G) Wireless Systems to reality. 3G Systems are intended to provide a global mobility with wide range of services such as web browsing, e-commerce, interactive games (audio, video and data), etc. To satisfy variety of user applications, a growing trend for wireless access networks is to support multiple radio technologies by using TCP/IP protocols. Mobile IP [1] is the standard mobility management protocol in the IP network and it provides the primary framework for mobility in the IP layer. Fast handover enhancements to Mobile IP are also proposed and are in the draft stage to reduce latency and packet drops during the handover [2].

Quality of Service (QoS) is a major concern for mobile wireless users. The wireless access network is evolving towards an All-IP network [3], which we consider in this work. New applications such as audio and video streaming require high data throughput and low latency. Since traditional IP networks are designed for data applications with a single service class, best effort, where all user packets compete equally for network resources. It is not sufficient to support various requirements of multimedia applications, which placed a significant burden on limited network resources, such as bandwidth and buffer space, and caused congestion due to limited resources. Such congestion does not encourage mass adoption of IP networks as transport mechanisms for real time and mission-critical applications [3]. Therefore, IP QoS models are developed to offer different levels of treatment to user packets. Differentiated Services [4], or DiffServ, is an IP QoS architecture based on differential treatment of packets that are marked at the network edge for the packets to receive priority according to user requirements. The DiffServ architecture provides QoS by aggregating flows into different traffic classes, marking each packet with a code point that indicates its class, and scheduling packet transmission according to their code points. Regardless of how the QoS is set up on the radio link, DiffServ is the most scalable and expected QoS model in the access network. Introducing DiffServ in the mobile wireless access network introduces some unique challenges and issues related to the impact of handover on DiffServ QoS assurance.

1.2 Quality of Service in Mobile Access Network

In this thesis, we study a DiffServ enabled mobile wireless network as shown in Figure 1.1.

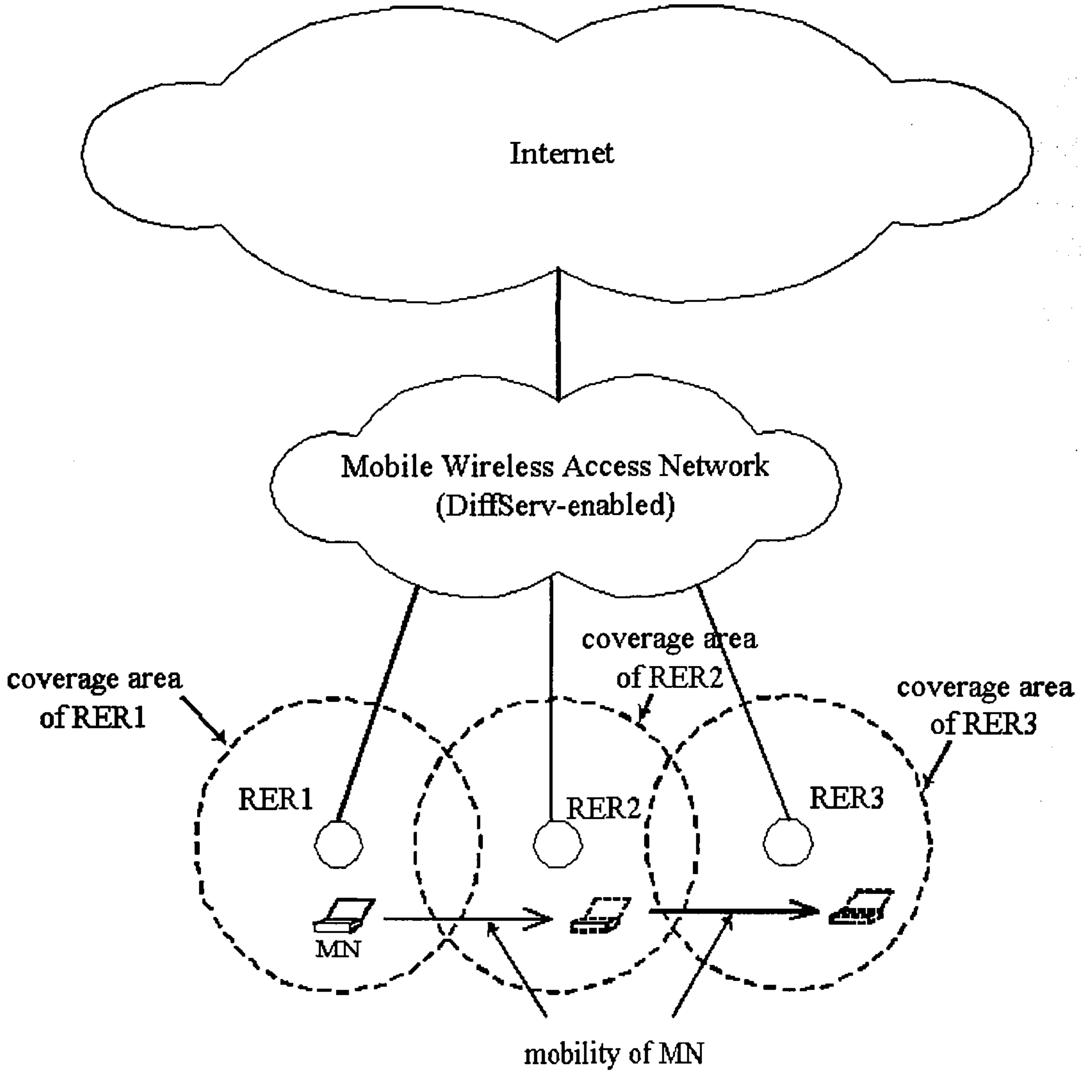


Figure 1.1 The DiffServ-enabled Mobile Wireless Access Network

The DiffServ-enabled mobile wireless access network is composed of Border Routers, Core routers and Radio Edge Routers (RER). Border Routers provide connectivity with the Internet (Packet Data Network). Core Routers form the wireline network that connects the radio edge with the core routers. The RERs have multiple interfaces: one is a wired interface connected with a core router and the other one is connected to an access point (AP) or base station (BS), which provides radio interface to mobile nodes. Each RER provides connectivity to mobile nodes within its coverage area, thus the

Mobile Nodes' (MN) mobility is from the coverage area of a RER to the coverage area of another RER.

In the DiffServ-enabled mobile wireless network, we consider two service levels: Assured Forwarding (AF) and Best Effort (BE). AF is a qualitative service with assured differences, while BE is a service without any QoS guarantee. Therefore, we consider AF a service level higher than BE. We also consider the priority of a service to be upgraded from its current service level to a higher service level and called it service upgrade priority. Then the Service Level Agreement (SLA) for a user should include information of service level, service upgrade priority, etc.

1.3 Problems for Providing Service Differentiation in Mobile IP Networks

In a DiffServ-enabled mobile wireless access network, we consider that all the flows generated and received by mobile nodes are marked with the appropriate DiffServ code points to become DiffServ flows. The impact of handover on the performance of DiffServ flows is a main factor for QoS guarantee. When a mobile node moves to a new cell with one or more active DiffServ flows, the flows compete for bandwidth resources along the new path with those existing active flows. On the other hand, at the time when a mobile node is leaving the old cell it releases the bandwidth resources that were assigned to its DiffServ flows. The handover flows can cause jitter in the performance of the existing flows, for example it may introduce an uncontrolled packet loss, delay, etc. The situation gets worse when the number of handover flows doesn't follow a predictable distribution. Mechanisms are needed to protect the existing flows from getting performance degradation while at the same time provide some QoS support to the handover flows so that their performance degradation can be controlled.

In this thesis we address the issue of the impact of handover on DiffServ flows when the bandwidth is not adequate. If we define the original qualitative service class as AF1, a transient class AF2 is introduced for handover flows, which is a service level lower than AF1 but higher than BE. We designed an admission control algorithm for handover flows. The algorithm provides schemes for the separation of handover flows from the existing flows. When the bandwidth is not enough to meet all the target rate of existing and handover AF1 flows, overall performance degradation is compared at the situation when the handover flow is marked as AF1 with the situation when it is marked as AF2. Our study shows that remarking the handover flows to the lower service class AF2 results in degradation of the handover flows but provides protection to the existing flows. We also propose algorithms of upgrading the handover flows to achieve the same level of service as they were receiving before the handover. A scheme for dynamically monitoring the bandwidth utilization and managing the queue for handover AF2 flows is proposed to realize service upgrading. Based on the idea of the above two algorithms, we further propose dynamic bandwidth provisioning schemes by adjusting weights of our scheduler according to the bandwidth utilization of different service levels. Our algorithms and schemes are presented and verified by different simulation scenarios. Results show that the proposed scheme is viable under variety of provisioning scenarios.

1.4 Thesis Outline

The thesis is organized in five chapters. In Chapter 1, we give an overview of the problem with some motivations and outline our approach for the proposed solution. The background information about Mobile IP, DiffServ model and mechanisms, and the issues of Quality of service in the mobile wireless access network is given in Chapter 2. In Chapter 3, we first discuss with the help of a simple simulation the impact of handover on DiffServ flows, and then present our proposed scheme and algorithms for admission control of handover flows to the new cell, dynamic service level adjustment, and adjustment of bandwidth allocation for different service classes. In Chapter 4 we present the results of our simulations to study the viability of the proposed scheme in handling different scenarios. Results show our algorithm is efficient and feasible. Finally we conclude the thesis in Chapter 5 with directions for future work.

Chapter 2
Background

A wireless access network is structured as a radio access network connected to PSTN and Internet through a wireline core network. The demand for data applications has pushed the focus of the third-generation (3G) systems and beyond to optimize the network for data communication. The 3G systems (e.g. UMTS [5] and CDMA2000 [6]) have already introduced IP transport in the core network with SIP, Mobile IP and other IP protocols. This trend will continue to grow with the advent of integration of radio technologies such as WLAN, WPAN and Cellular. In this thesis we consider a DiffServ enabled wireless core network that employs Mobile IP handover protocol. We first discuss briefly Mobile IP protocol, and then DiffServ QoS model and mechanisms in detail.

2.1 Mobile IP

In an IP network, the location of a node is identified by its IP address. According to the IP address that is assigned on the network, called home network, the node is reachable through normal IP routing. However, when the node roams away from its home network, it is no longer reachable using normal IP routing and its active sessions are terminated. Mobile IP [7] was designed to enable users to keep the same IP address while moving to a different network (which may even be in a different wireless operator domain), thus ensuring that a roaming individual could continue communication without sessions or connections disruption. Since Mobile IP is based on IP, any media that can support IP can support Mobile IP.

There are three components in Mobile IP: Mobile Node (MN), Home Agent (HA) and Foreign Agent (FA), as shown in figure 2.1.

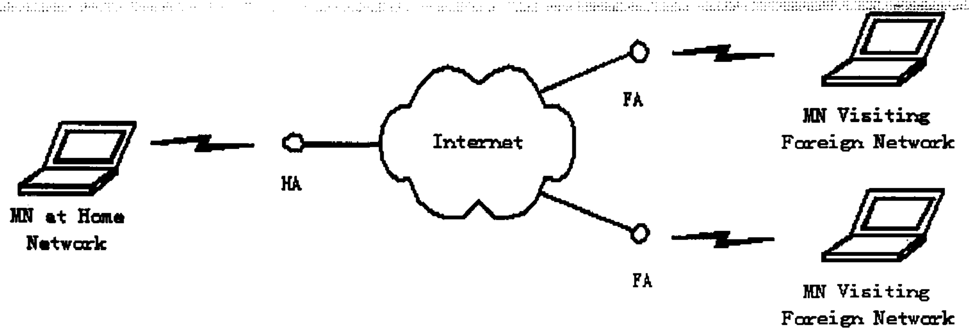


Figure 2.1 Mobile IP Components

A Mobile Node (MN) is an Internet-connected device whose location and point of attachment to the Internet may frequently be changed. This kind of node is often a cellular telephone or handheld or laptop computer, or even a router. A node or device on the Internet that communicates with the mobile node is called a Correspondent Node (CN). Mobile IP defines two IP addresses for mobile nodes that are visiting foreign network domains (networks). The home address identifies the mobile node's location in the home domain, which is known to all correspondent nodes in the Internet. A care-of-address (CoA) is used to identify the mobile node's current location (i.e. subnet) in the foreign network. It is the termination point of the tunnels toward the Mobile Node when it is in a foreign network. The Home Agent (HA) is a device (typically a router) in the home network serving as the anchor point for communication with the mobile node. It maintains an association between the home address of the mobile node and its care-of-address, which is the current location of the mobile node on the foreign or visited network. It intercepts the packets from a CN destined to the MN's home address on the home subnet, and tunnels them to CoA. The Foreign Agent (FA) is a router that provides CoA to the mobile node when the MN roams into its network, and then it detunnels the packets from the home agent and delivers them to the mobile node.

In Mobile IP v4 [7], there are three main phases:

1. Agent discovery.

During this phase, a mobile node discovers its home agent and foreign agent. Usually this can be done in two ways. One is that the Home Agent and Foreign Agent advertise their services on the network by using the ICMP Router Discovery Protocol (IRDP). The Mobile Node listens to these advertisements to determine if it is connected to its home network or foreign network. The IRDP advertisements carry Mobile IP extensions that specify whether an agent is a Home Agent, Foreign Agent, or both, its care-of address, the types of services it will provide such as reverse tunneling and generic routing encapsulation (GRE), and the allowed registration lifetime or roaming period for visiting Mobile Nodes. Another way is that a Mobile Node can send out an agent solicitation, rather than waiting for agent advertisements. This solicitation forces any agents on the link to send an agent advertisement immediately.

If a Mobile Node determines that it is connected to a foreign network, it acquires a care-of address. There are two types of care-of addresses: Care-of address acquired from a Foreign Agent, and colocated care-of address. A Foreign Agent care-of address is an IP address of a Foreign Agent that has an interface on the foreign network being visited by a Mobile Node. A Mobile Node that acquires this type of care-of address can share the address with other Mobile Nodes. A colocated care-of address is an IP address temporarily assigned to the interface of the Mobile Node itself. A colocated care-of address represents the current position of the Mobile Node on the foreign network and can be used by only one Mobile Node at a time.

When the Mobile Node receives a Foreign Agent advertisement and detects that it has moved outside of its home network, it begins the Mobile IP registration process.

2.Registration

In the registration phase, the Mobile Node registers its CoA with the Foreign Agent and Home Agent.

Usually, a Mobile Node is configured with an IP address and mobility security association (which includes the shared key) of its Home Agent. It is also configured with either its home IP address, or another user identifier, such as a Network Access Identifier. According to this information and the information it learns from the Foreign Agent advertisements, the Mobile Node generates a Mobile IP registration request. It adds the registration request to its pending list and sends the registration request to its Home Agent either through the Foreign Agent or directly depending on which care-of address it is using. Since in our research we use the care-of address acquired from a Foreign Agent, we consider the registration process in which the Mobile Node sends the registration request to its Home Agent through the Foreign Agent. The Foreign Agent checks the validity of the registration request, which includes checking whether the requested lifetime exceeds its limitations, and the requested tunnel encapsulation is available and the reverse tunnel is supported. If the registration request is valid, the Foreign Agent adds the visiting Mobile Node to its pending list before relaying the request to the Home Agent. If the registration request is not valid, the Foreign Agent sends a registration reply with appropriate error code to the Mobile Node. The Home Agent checks the validity of the registration request, which includes authentication of the Mobile Node. If the registration request is valid, the Home Agent creates a mobility binding (an association of the Mobile Node with its care-of address), a tunnel to the care-of address, and a routing entry for forwarding packets to the home address through the tunnel.

The Mobile Node renews its registration before the registration lifetime expires. If the registration is denied, the Mobile Node makes the necessary adjustments and attempts to register again. During the renewal of registration, the Home Agent and Foreign Agent update their mobility binding and visitor entry respectively. Thus, a successful Mobile IP registration sets up the routing mechanism for transporting packets to and from the Mobile Node as it roams.

3.Tunneling

The Mobile Node's movement is transparent to correspondent nodes. When the Mobile Node sends packets, it uses its home IP address; so that it seems that the Mobile Node is always on its home network. Packets addressed to the Mobile Node are routed to its home agent first. Then the Home Agent intercepts and tunnels them to the care-of address toward the Mobile Node. The tunneling overhead includes the time spent in encapsulation and decapsulation of the packets and the transmission of the tunnel header. IPinIP Encapsulation is the default tunnel mode, while GRE and minimal encapsulation within IP are optional.

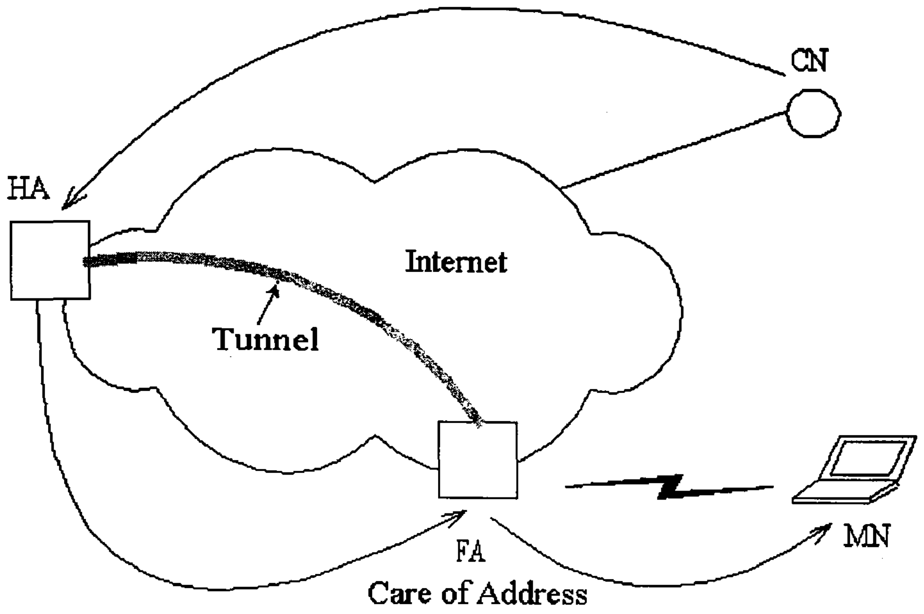


Figure 2.2 Mobile IP Tunnel

Mobile IP packet forwarding is shown in figure 2.2. The MN can send packet directly to the Correspondent Node (CN), but it receives packets via HA from the CN. This creates a triangle routing MN-CN-HA, which can be prevented by the proposed route optimization wherein the MN updates the CN with its CoA and thus packets traverse through the optimal route between MN-CN. In Mobile IP v6 [8] the MN can directly send the registration message, called Binding Update (BU), to the HA, hence there is no provision for FA in the network.

2.2 Wireless Access Network

2.2.1 Infrastructure

In a wireless access network, edge routers connected to one or more base stations are called **Radio** Edge Router (RER). They provide connectivity to mobile nodes. For each RER, there is a certain geographic coverage area within which mobile nodes can communicate to it. We call this coverage area as a cell. Usually neighboring RERs overlap with each other's coverage area so that consistency of communications is ensured when mobile nodes move from one cell to another. The infrastructure of wireless access network and the handover process is illustrated in Fig2.3.

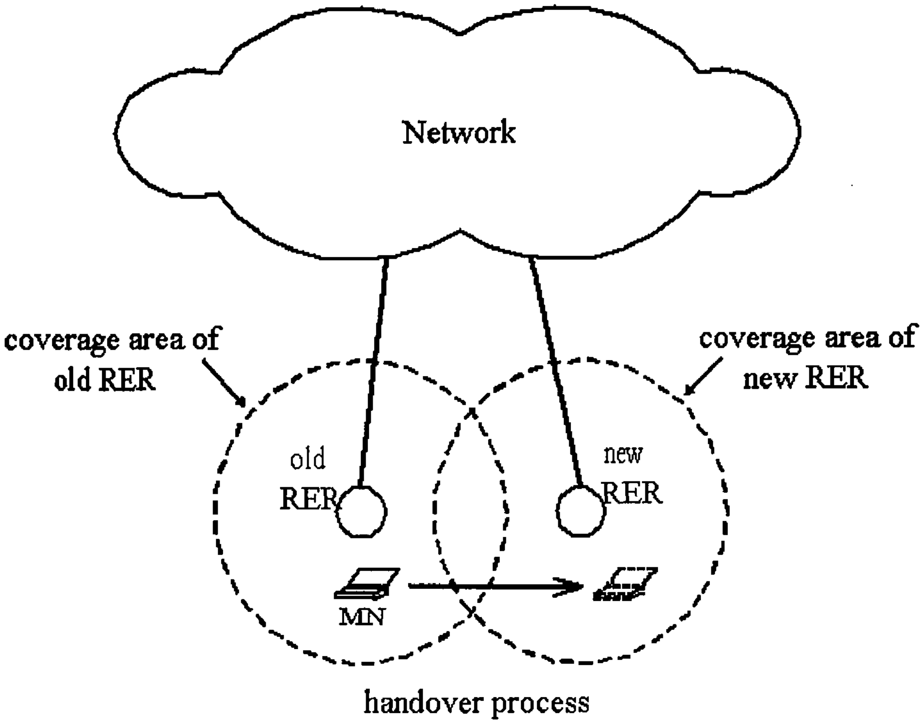


Figure 2.3 The Infrastructure of Wireless Network and Handover Process

2.2.2 Handover

Handover is the transition for a given mobile node from the coverage area of one RER to the coverage area of a geographically adjacent RER as the mobile node moves around. In this procedure, data transfer session is maintained while mobile nodes moving from one cell to another. Each time a mobile node moves from one cell into another, the network automatically switches coverage responsibility from one RER to another. In a smooth handover process, the communication disruption should be minimal.

2.3 DiffServ

The Integrated Services (IntServ) and Differentiated Services (DiffServ) are two main Quality of Service architectures for the IP network. Since IntServ needs to explicitly signal and dynamically allocate resources at each intermediate node along the path for each flow, it is less scalable and abandoned in ISP networks in favor of more scalable DiffServ.

DiffServ is a class-based QoS system that provides qualitative assurance to aggregate flows [4]. No per flow state needs to be maintained in the core routers, neither is there an explicit connection setup phase. It recognizes the boundaries of Internet Service Providers, and assumes that ISPs define the service classes for their own networks. It defines standard behavior (treatment) of flows, called Per Hop Behavior (PHB), which is used by the ISPs to construct their own services.

2.3.1 DiffServ Domain and Architecture

A DiffServ domain is a set of DiffServ nodes, which operates with a common service provisioning policy and set of PHB groups executed on each node. Devices in a DiffServ domain are illustrated in Figure 2.4.

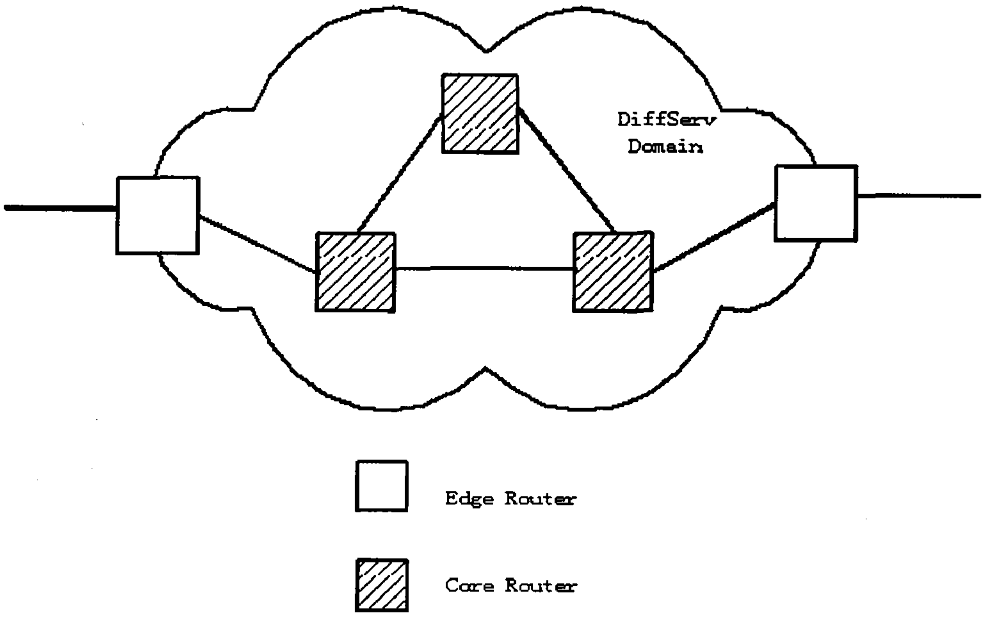


Figure 2.4 Components in a DiffServ Domain

DiffServ distinguishes between the edge and the core of a network. In a DiffServ domain, per flow traffic conditioning is limited to the boundaries of the network (edge routers), whereas the core of the network provides treatment to only aggregate flows. As traffic enters the network it is classified and conditioned at the edge router and assigned to a behavior aggregate. DiffServ defines standard forwarding treatments of packets inside the network, which are called Per Hop Behavior (PHB). A PHB is a description of the externally observable forwarding behavior of a DiffServ node applied to a particular DiffServ behavior aggregate. It defines a forwarding treatment of a single packet in a router. These PHBs do not offer any quantitative guarantees e.g. fixed bandwidth, and bounds on packet delay or jitter. They provide qualitative assurance and means to implement service classes. All the edge and core routers perform Per Hop Behavior (PHB) along the path. Packets are identified for a particular treatment (PHB) through DiffServ Code Points (DSCP), which are standard bit combinations of the DSCP in the IP header. Packets marked with the same DSCP are treated the same way. Flows are aggregated based on desired behavior, and marked with the same DSCP at the edge, which in turn get the given differentiated treatment within the network.

The DiffServ architecture has three major components. One is the policy and resource manager, which handles the creation of network policies and distribution of those policies to the DiffServ routers. The other components are edge routers and core routers. DiffServ attempts to restrict complexity to only the edge routers of a domain.

A policy and resource manager is a necessary component of a DiffServ network that allows an administrator to communicate policies to the edge and core devices. A policy specifies which traffic receives a particular level of service in the network.

Edge routers are used between the hosts and the core network. They classify and possibly condition the incoming traffic using Traffic Conditioning Agreement (TCA). Edge routers are responsible for examining incoming packets and classifying them according to policy specified by the network administrator. Figure 2.5 shows the building blocks of an edge router. Its main functions are the following:

1. It classifies packets for appropriate DSCP marking. It performs detailed packet classification, called MF (Multi Field) classification, based on the following IP-header fields: source address, destination address, traffic type or any combination of these.

2. It marks packets with the appropriate DSCP based on the SLA and the policies.

3. It ensures that user traffic adheres to its policy specifications, by shaping and policing traffic. Shaping includes measuring transmission rates of the given traffic aggregation and conforming it to pre-defined values.

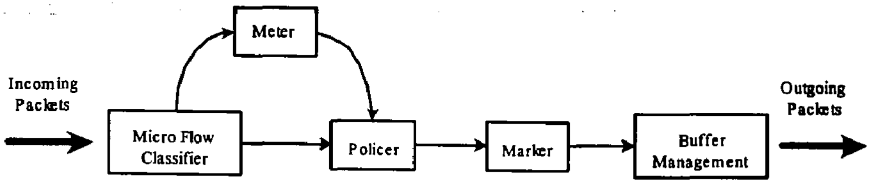


Figure 2.5 Functions of Edge Routers

Core routers are routers that are only connected to other routers inside the network. They only perform PHB on incoming packets. Their main responsibilities are:

1. Classifying incoming packets based on the DSCP marking done on the packet by the edge routers, called BA (Behavior Aggregate) classification.

2. Performing PHB while forwarding incoming packets according to the DSCP markings. (Core routers' behaviors follow the marking done by edge routers).

2.3.2 Assured Forwarding

Assured Forwarding (AF) and Expedited Forwarding (EF) are the two DiffServ PHBs that provide different quality assurances. AF [10] PHB is suggested for applications that require a better reliability than the best-effort service. It offers the same delay characteristics as of the best effort class, however the firmness of its guarantee on differential treatment depends on how well the individual links are provisioned for bursts of assured packets. Since AF can provide flexible QoS with relative qualitative assurance, we use this PHB in our research, and discuss it in detail.

AF provides differential treatment of traffic by discarding more low priority packets during times of congestion than high priority packets. AF defines four classes of service. In each DiffServ node, a certain amount of forwarding resources such as buffer space and bandwidth is allocated for each AF class. All packets from a certain class are put into an assured queue and the queue is managed by a queue management scheme called Random Early Detection with In and Out (RIO). Within each AF class, there are three drop precedences. The different drop precedence levels are also referred in terms of their colors. For example, Green – for the lowest drop precedence level, Yellow – for the medium, and Red – for the highest drop precedence level. During congestion period, the drop precedence of a packet determines the relative importance of the packet within each class and packets with lower priority is more likely to be dropped than other packets. In other words, The Assured Forwarding mechanism is a group of code points that can be used in a DiffServ network to define four classes of traffic, each of which has three drop precedences. The drop precedence enables differential treatment of traffic within a single class.

2.3.3 Buffer Management

Packets belonging to different DiffServ classes are enqueued in different queues. Within each queue, buffer management is one of our main concerns. The basic algorithm for buffer management in routers is Drop Tail. Drop tail queues with FIFO simply accept any packet that arrives when there is sufficient buffer space and drop any packet that arrives when the buffer space is insufficient. This algorithm is simple and easy to implement, but usually causes problems with multiple constant packet flows and global synchronization of TCP connections.

2.3.3.1 Overview of RED Algorithm

Random Early Detection (RED) [11] is a congestion avoidance algorithm that can be implemented in routers. RED gateways are designed to detect incipient congestion by computing a weighted average queue size because a sustained long queue is a sign of network congestion. When a packet arrives, a RED gateway checks the weighted average queue size and compares it with the specified minimum and maximum thresholds. If there is congestion, it notifies, either by dropping a packet or by setting a bit in a header field of the packet, probabilistically. A RED is in one of the following three phases in determining about the packet drop:

1. Normal Operation

If the average queue size is less than the minimum threshold, no packets are dropped.

2. Congestion Avoidance

If the average queue size is between the minimum and maximum thresholds, packets are dropped with a certain probability. The probability is a function of the average queue size. It is increasing linearly as the buffer begins filling up, so that larger queues lead to higher drop probabilities.

3. Congestion Control

If the average queue size is greater than the maximum threshold, all incoming packets are dropped.

2.3.3.2 Multiple RED Mechanism

As we have discussed before, Assured Forwarding is a group of code points that can be used to define four classes of traffic, and each class has three drop precedence that enable differential treatment of traffic within a single class. Assured Forwarding uses the RED mechanism by enqueuing all packets for a single class of traffic into a certain physical queue in which there are three virtual queues (one for each drop precedence), shown in Fig 2.6. We call the RED mechanism used by Assured Forwarding as Multiple RED mechanism (MRED) [12]. In MRED, Different virtual queues have different RED parameters thus packets from some virtual queues are dropped more frequently than packets from other virtual queues. A packet assigned a code point that corresponds to a virtual queue with lower drop precedence, in other words, with relatively lenient RED parameters, is given better treatment when congestion happens.

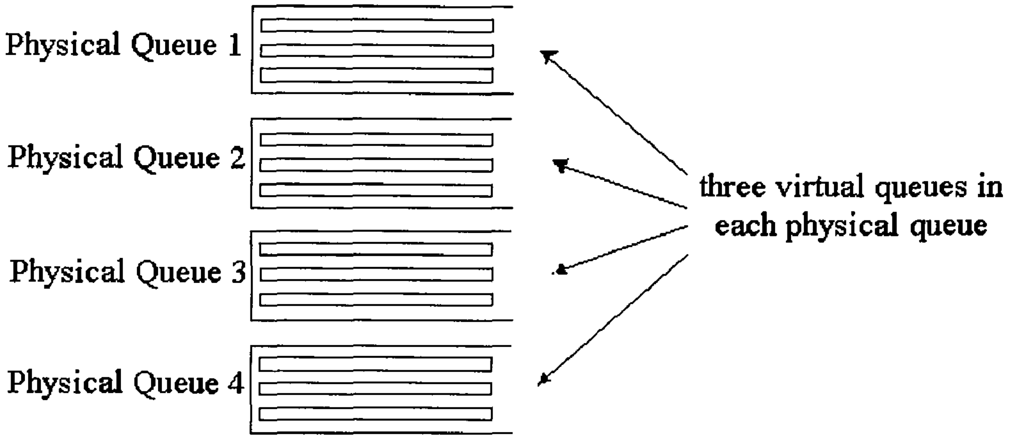


Figure 2.6 DiffServ Queues for AF

In MRED, drop probability for packets with different drop precedence (packet color) have to be calculated independently for each drop precedence, thus multiple sets of RED thresholds need to be maintained – one for each drop precedence. There are different schemes applied for calculating the average queue used for the drop decision, which are classified into four categories [12], as showed in Table 2.1:

RED Variants	Single Average Single Threshold (SAST)	Single Average Multiple Thresholds (SAMT)	Multiple Average Single Threshold (MAST)	Multiple Average Multiple Threshold (MAMT)
Examples	RED	WRED	-	RIO-C, RIO-D
Availability for MRED	No	Yes	No	Yes

Table 2.1 Taxonomy of RED Mechanism

▪ Single Average Single Threshold (SAST): This RED mechanism does not distinguish between packets of different colors and maintains a single average queue length and the same min_th and max_th thresholds for the packets of all colors. In fact, SAST is simply plain RED.

▪ Single Average Multiple Thresholds (SAMT): The average queue length is based on total number of packets in the queue regardless of their color, but packets of different color have different drop thresholds.

▪ Multiple Average Single Threshold (MAST): Average queue length for packets of different colors is calculated differently. For example, average queue length for a color can be calculated using number of packets in the queue with same or better color. However, packets of all color have the same min_th and max_th thresholds.

▪ Multiple Average Multiple Threshold (MAMT): Average queue length for packets of different colors is calculated differently, and packets of different color have different min_th and max_th thresholds.

Among above RED variants, SAMT and MAMT are used to implement MRED. An example of SAMT is Weighted RED, while RIO-C and RIO-D are examples for MAMT. Several popular MRED modes are list below:

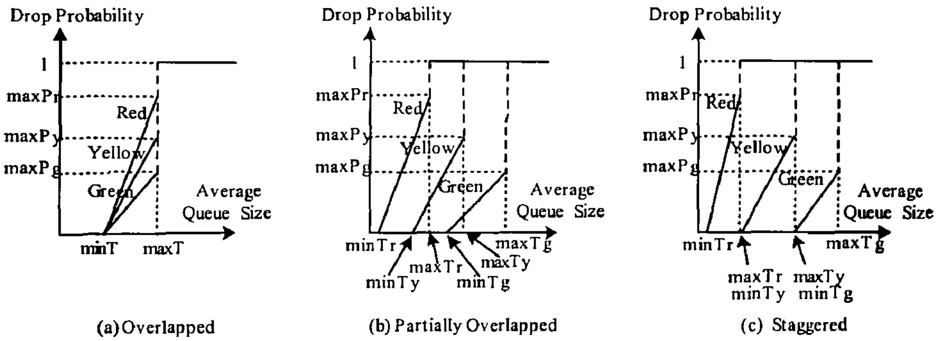
▪ RIO Coupled mode (RIO-C) [13]: For one physical queue, the probability of dropping an out-of-profile packet is based on the weighted average lengths of all its virtual queues; while the probability of dropping an in-profile packet is based solely on the weighted average length of its virtual queue. Moreover, multiple RED threshold parameters are maintained – one for each color.

- RIO De-coupled mode (RIO-D) [13]: The probability of dropping an out-of-profile packet is based on the size of its virtual queue only, while the probability of dropping an in-profile packet is the same as RIO-C. Moreover, multiple RED threshold parameters are maintained – one for each color.

- Weighted RED mode (WRED) [14]: A single average queue length that includes packets of all colors is calculated. For any arrival or departure of green, yellow or red packets, WRED updates the single average queue length based on the total number of packets of green, yellow and red color if they are all applicable. However, multiple RED threshold parameters are maintained – one for each color.

- DROP mode: As soon as the queue size reaches the minimum threshold, all packets are dropped regardless of marking.

Generally, for all RED variants, the threshold parameters for packets of different colors could be set in three ways: overlapped, partially overlapped and staggered, as showed in figure 2.7. According to the above taxonomy of RED variants, (a) is only suitable for SAST and MAST because in these mechanisms packets of all color have the same minT and maxT thresholds. (b) and (c) are suitable for SAMT and MAMT, thus they are used to implement MRED. In fact, (b) is more popular than (c).



(a) Overlapped (b) Partially Overlapped (c) Staggered

Figure 2.7 Multiple RED Threshold Parameter Settings.

2.3.4 Queue Management

The scheduler is a key component in a router that distributes the link capacity to multiple queues. The scheduling mode can be Weighted Round Robin (WRR) [15], Weighted Interleaved Round Robin (WIRR) [16], Round Robin (RR) [17], Priority Queuing (PRI) [18], and other scheduling modes. Two popular schedulers related to our research are discussed below in details.

1. WRR scheduler

Weighted Round Robin scheduler is a variant of the Weighted Fair Queuing (WFQ) algorithm [19]. It shares link access circularly, one buffer at a time. Each buffer has a weight that refers to the relative share of link access time compared to other buffers. Any buffer with 0 weight will be skipped and its link access time will be shared by other buffers according to their weights.

By properly configuring the WRR scheduler, we can arrange different link access time for different buffers so that service differentiated from class to class and different portions of bandwidth are assigned to different classes. In contrast with PRI scheduler, high priority traffic doesn't exhaust lower priority traffic in WRR.

2. PRI scheduler

Priority queuing scheduler sets different priorities to different buffers. The non-empty buffer with highest priority always gets link access. This scheduler can guarantee small packet delay and jitter, but on the other hand high priority traffic can easily exhaust lower priority traffic. To avoid the starvation, the buffer bandwidth limits are configured properly.

2.3.5 Policy and Marking Schemes

Before traffic enter a DiffServ domain, users have to agree with a certain traffic profile so that there won't be any unforeseen congestion. The profile usually includes a Committed Information Rate (CIR) and allowed Peak Information Rate (PIR). When traffic comes into a DiffServ domain, it is metered, shaped and marked with certain code point. The purpose of metering and policing is to smooth the bursts over time, thus to ensure that there is no violation even though some flows may be misbehaving and violating agreed profile. The purpose of marking is to indicate whether current packet violates the profile or not. For example, in a three color marking scheme, the green indicates that the packet has arrived below CIR, the yellow shows that the packet has exceed committed profile but still falls within the peak rate, and the red shows that the packet has arrived at or above PIR.. Colors are coded by the drop precedence of AF class [10]. Some commonly used policers for AF are discussed below:

- Token Bucket: The Token Bucket algorithm is illustrated in fig 2.8. In this algorithm, a token bucket of depth Committed Burst Size (CBS) is filled at the rate of the Committed Information Rate (CIR). CBS and CIR are important parameters in the traffic profile committed by users. When a packet comes, its length is compared with the occupancy of the token bucket b. It is marked *in* if there are enough bytes of token in the bucket. Otherwise, it is marked *out*. If a packet is marked *in*, a number of bytes (referred to as tokens) equal to the packet length is subtracted or taken from the bucket. *In* refers to a lower drop precedence (or green in color), while *out* refers to a higher one (or red in color) [20].

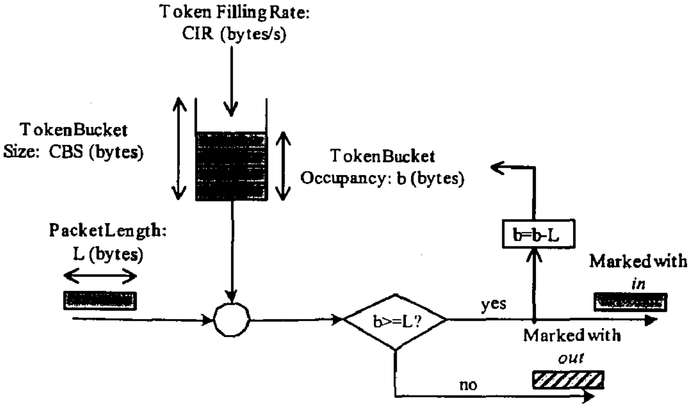


Figure 2.8 The Token Bucket Policy

▪ Single Rate Three Color: According to this policy, incoming packet stream is metered and marked based on a Committed Information Rate (CIR) and two associated burst size, a Committed Burst size (CBS) and an Excess Burst Size (EBS). Token bucket C (with maximum size CBS) and token bucket E (with maximum size EBS) is filled at the same rate of CIR and EBS is assumed to be bigger than CBS. A packet is marked green if there are enough token in token bucket C, yellow if the packet's length exceeds CBS, but not EBS; otherwise it is marked red [21].

▪ Two Rate Three Color: Similar with Single Rate Three Color, this policy has two token bucket. Token bucket C's maximum size is CBS and it is filled at the rate of CIR. Token bucket P's maximum size is Peak Burst Size (PBS) and it is filled at the rate of Peak Information Rate (CIR). . A packet is marked green if tokens are enough in token bucket C, yellow if the packet's length exceeds CBS, but not PBS; otherwise it is marked red [22].

▪ Time Sliding window: In this policy, metering and marking is done based on the Committed Target Rate (CTR) and the Peak Target Rate (PTR). A rate estimator provides an estimate of the traffic stream's arrival rate, which approximates the running average bandwidth of the traffic over a specific period of time (length of the sliding window). If the estimated average rate is no more than the CTR, the packets of the stream are marked as green. If the estimated average rate exceeds CTR but is no more than PTR, packets of the stream are marked yellow with probability P0 and green with probability (1-P0). If the estimated rate exceeds PTR, packets are marked red with probability P1, marked yellow with probability P2 and marked green with probability (1-(P1+P2)) [23].

2.4 Quality of Service Issues

In the 3G wireless systems, all service is carried directly on IP so its QoS model is quite different from traditional models.

2.4.1 Mobility and QoS

Usually, QoS parameters include bandwidth, packet delay, packet loss rate, and jitter. Some of these parameters are quite different between mobile and fixed worlds. In the mobile environment, bandwidth is usually lower than in the fixed environment, while delay is usually longer. There are some unique challenges to provide and maintain QoS in a mobile environment. Besides the challenges of maintaining wired part network level QoS, handover between access routers, frequent changes of IP address, and competition for resources among mobile users all increase the complexity of QoS provisioning in the mobile environment.

In a mobile network environment, a mobile node may change its point of attachment to the network many times during a session. There is also disruption during the handover period. These may lead to changes of routing within the access network and possibly other changes in the end-to-end session. To maintain the active sessions on the mobile nodes, the network should negotiate QoS along the new path during the handover process. After the negotiation with the network, the QoS contract should be maintained if it is possible, or downgraded to a lower service level that is still acceptable by users.

As we have mentioned in previous sections that Mobile IP mechanism use tunnels during handover to forward packets between the old and new access routers for the mobile node. The mechanism can help eliminate packet loss, but the packets should be treat very carefully. They must be allocated an appropriate QoS to ensure that they reach the mobile node within the confines of the QoS contract.

Considering the issue of disruption during the handover process, there is a period of time during which the end-to-end connection data path is incomplete. How much this disruption can affect the application performance depends in the nature of the application and the period of the disruption. For example, a voice application can only tolerate very short disruptions.

2.4.2 Traffic Classes and QoS Classes Mapping

In 3G wireless systems, traffic types are divided into four different service classes based on their individual requirement of bit rate, bit error rate, delay, and etc [24]. The four defined classes are:

- Conversational class: The conversational class services are mainly for conversational real-time applications such as voice, videoconference, video gaming and so on. The critical performance requirement for this class is consistency in time relations, including low delay and low delay jitter requirements. The main idea of these requirements is to preserve source data rate according to human perception for this kind of applications. The conversational class services also resemble Constant Bit Rate (CBR) services in Asynchronous Transfer Mode (ATM) and can be supported by fixed resource allocation in the network [25].

- Streaming class: The streaming class services include streaming media applications such as streaming audio, streaming video (video on demand), webcast, etc. They also require low delay and low delay jitter but the requirements is not as strict as those for the conversational class services. Streaming class services can be considered as real-time variable bit rate services or variants of the constant bit rate services.

- Interactive class: Examples of the interactive class services are web browsing, network gaming, e-commerce, remote Local Area Network (LAN) access, etc. They all require high throughput, and low loss rate. Traffic flow prioritization is also considered important within this service class.

- Background class: The background class services are for traditional best-effort services such as non real-time download of emails, file transfers and so on. They don't have special requirement for delay, delay jitter or throughput, even though low loss rate is still critical for minimizing retransmissions. Background services have the lowest priorities among services of all other classes.

Considering DiffServ as our QoS model for the core network, we can do some mapping between the traffic classes and the QoS classes, as illustrated in Fig 2.9. In DiffServ, there are three service classes: Expedited Forwarding (EF), Assured Forwarding (AF) and Best Effort (BE). Since EF class can give guarantees on parameters such as delay and throughput, we consider mapping the conversational class to EF. The streaming class services have similar but looser requirements compared to the conversational class services, so that stream class can be mapped to either EF or AF. To ensure that the interactive class services receive assured throughput, it is better to map the interactive class to AF. It is also okay for some researchers to map it to BE [26], if the requirement is not so strict. The background class is usually mapped as BE because it has the lowest priority among all the classes.

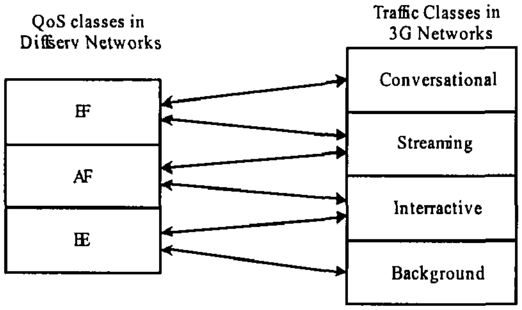


Figure 2.9 Mapping Between 3G Traffic Classes and DiffServ Classes

2.4.3 QoS Parameters

Based on the analysis of traffic classes in former sections, we consider the expected service performance parameters for 3G wireless networks as throughput, delay, delay variation, drop probability, and bit error rate.

- **Throughput**: Throughput is also known as bit rate. The bit rate between two communicating end-systems is the number of binary digits that network is capable of accepting and delivering per unit time. Practical bit rate is limited to the capability of the network and the destination in accepting and processing information. Considering our core network as a DiffServ-enabled IP network, then traffic aggregation is an important factor affecting the throughput parameter. We can divide traffic aggregate in an IP network into congestion sensitive TCP flows and congestion insensitive UDP flows because TCP reduce their traffic rate if packets are lost while UDP show no response to losses. Seddigh et al [27] have conducted a detailed experimental study of five different factors that impact throughput assurances for TCP and UDP flows in an AF based DiffServ-capable IP network. The study demonstrated that the factors could cause different throughput rates for end-users even though they have contracted identical service agreements. We will discuss these factors in detail in following sections.

- **Delay**: Delay can be defined as the time period between the emission of the first bit of a data block by the transmitting end system and its reception by the receiving end-system. In the 3G wireless networks, both conversational class and streaming class have strict requirements for transfer delay. To provide guarantees on end-to-end delay in a DiffServ-capable IP network, we consider controlling specification in relative packet forwarding urgency among the classes by appropriate scheduling mechanism among the queues within a particular router.

- **Delay variation**: The variation in delay is defined as delay variation or jitter. When a stream of packets traverses a network, each packet may experience different delay due to buffering in routers. For real-time applications, delay variation is an essential performance parameter. An applicable way to overcome delay variation is to make the receiving system waiting a sufficient time (called delay of set) before the play out, so that most delayed packets have a chance to act like arrive in time.

- **Drop probability**: In a DiffServ-capable network, drop probability is a critical parameter for different treatment to packets. It is no doubt that certain drop assurance is possible in a single domain. Since the DiffServ framework is intended to operate on bilateral agreements between two neighboring domains, the owner of a domain can also obtain a service level agreement with its neighbors on drop probability. However, it is still not clear how to ensure the end-to-end drop probabilities specified in the contract for aggregate flows going across multiple domains in the Internet [28].

- **Bit Error Rate**: Bit Error Rate (BER) is defined as the percentage of bits that have errors to the total number of bits received and it is expressed as ten to a negative power. For example, if BER is 10^{-6} in a transmission process, we know that out of 1,000,000 transferred bits one bit is not correct. The BER is an indication of how often a data segment (in packets or other units) has to be retransmitted due to an error.

Considering AF PHB as our differentiated drop mechanism to provide IP QoS in the core network, different user traffic classes obtain different treatments based on their requirements, but the AF PHB does not focus on delay. Nevertheless, QoS parameters have inherent relations. For example, if the throughput is sufficient for a certain traffic flow, there will be low loss rate thus we have low retransmission rate and low delay. Therefore, it is probable that end-users use throughput assurance as a measure of good or bad network performance [27].

2.4.4 Throughput Assurance Issues

Some studies have shown a different list of factors that impact throughput issues for TCP and UDP flows in an AF based DiffServ-capable IP network [27] [29]. These factors can cause different throughput for equal paying end-users who have committed identical service level agreements (SLA), which includes values for target rates (CIR) set for the TCP connections. The main factors of our concern are:

- **Round Trip Time (RTT)**: RTT is defined as the time between the transmission of a packet and the receipt of its acknowledgment or reply. TCP uses a self-clocked sliding window based mechanism and adjusts its rate based on RTT. Thus bandwidth assurance can be regarded as a function of RTT. Aggregate TCP flows with different RTTs get different shares of bandwidth in spite of identical target rates. In an over-provisioned network, flow aggregates will achieve their target rate irrespective of their RTTs, but there is an unfair sharing of the excess bandwidth since flows with lower RTT will get more portion of the excess bandwidth than those with higher RTT. In a under-provisioned network, neither of the aggregated flows will achieve their target rate but flows with lower RTT is close to their target rate compared with those with higher RTT. The phenomenon can be explained by the following equation by Mathis et al [30]:

$$BW < \frac{MSS}{RTT}\frac{1}{\sqrt{p}}$$

The equation reveals the relationship between TCP bandwidth, BW, and the factors of Round Trip Time (RTT), packet size (MSS) and packet loss rate (p). Flows with different values of any of the three factors will have different distribution of excess bandwidth in the over-provisioned network and different degrees of degradation in the under-provisioned network. The impact of packet size and packet loss rate on bandwidth is discussed in detail in following passages.

- **Packet Size**: In an over-provisioned network, flows with the same RTT will achieve their target rate in spite of their different packet size. But the there will be an unfair sharing of the excess bandwidth in favor of the target aggregates with larger packet size. Similarly, in an under-provisioned network, no aggregated flow will achieve its target rate but those with larger packet size will be closer to their target rates.

- **Size of Target Rate**: Based on the common sense, in an over-provisioned network, flows with larger packet size should get more excess bandwidth. But it is not the case according to the detailed study of [27]. Their results show that the excess bandwidth is distributed almost equally among flows with different values of target rate. In a under-provisioned network, still no target is achieved, but there is fair degradation of bandwidth for flows with different target rate.

- **Number of Active Flows**: The number of active flows in the core of the network is an important factor that impacts the TCP throughput for both individual flows and flow aggregates [31]. As the number of active flows in the core network increases, the queue length is more prone to cross the maximum threshold thus causes packet drop and retransmission and will further cause unfair sharing of TCP bandwidth. So the effectiveness of RED parameters is partially dependant on the number of active flows in the core network. For a given set of RED parameters, the end-to end TCP flow behavior will change as the number of active flows changes with time.

- **Impact of Non-responsible flows**: If we let the responsive TCP flows and Non-responsible UDP flows share the same service class with the same drop precedence, non-responsible flow will starve the responsive flows in the under-provisioning case; while both will achieve their target rate in the over-provisioning case. Alternatively, if we let the responsive TCP flows and Non-responsible UDP flows share the same service class with different drop precedence, TCP flows can be protected from the impact of the non-responsible UDP flows.

- **Number of Micro-flows in an Aggregate**: Study shows that in the over-provisioned scenario, aggregates with larger number of micro-flows will get more share of the excess bandwidth. In the under-provisioned scenario, situation is similar, but the difference is not so obvious.

- **Different TCP Stacks**: Currently existing TCP stacks such as Reno, new Reno, SACK, etc have different mechanisms of handling packet drops, which causes different levels of aggressiveness to maintain throughput in case of packets drop. For example, even though a Reno user and a SACK user have the same drop probability, they still get different throughputs due to the different congestion avoidance mechanisms they use when packet drop happens.

Chapter 3

Dynamic Bandwldth Management Algorithm

3.1 Impact of Mobile IP Handover on the Performance of DiffServ Flows

3.1.1 Overview

Our research is focused on QoS provisioning in a DiffServ-enabled mobile wireless access network. It is well known that DiffServ can provide efficient and scalable service differentiation for the wired Internet infrastructure. However when it is used in the mobile wireless access network, situation becomes dynamic and several challenging issues arise. For example, in a DiffServ network, packets are marked and assigned a DiffServ Code Point (DSCP) at ingress edge routers and get special treatment along the path pointing accordance with the PHB identified by the DSCP. However in a mobile IP network, traffics destined to a mobile node's home address are redirected to the Foreign Agent form the Home Agent through the Mobile IP tunnel. Then the question arises where shall the packets be marked and is there any need for remarking the redirected traffic? Moreover, when a mobile node moves to the coverage area of a new cell competes for resources with the existing users, then how does the network allocate the resources to achieve the performance objective? In this thesis we study the above issues related to QoS provisioning and management for a DiffServ-enabled Mobile IP network. We consider that all the flows generated and received by mobile nodes are marked with the appropriate DSCPs, which are called DiffServ flows. We first present the impact of Mobile IP handover on the performance of DiffServ flows. Then, we present the proposed admission control algorithms.

For the simulation study we present in this thesis we only consider TCP flows because TCP flows constitute majority of the Internet traffic today. To investigate the impact of Mobile IP handover on the performance of DiffServ flows, we simulated a number of TCP connections experiencing variety of handover situations.

3.1.2 Problem Simulation and Analysis

We simulated an all-IP wireless access network with three base stations forming three cells, as illustrated in Figure 3.1. We assume a base station to be the Radio Edge Router (RER) implementing Mobile IP mobility agents. For example, three base stations, *HA*, *FA1* and *FA2* are edge routers that also serve as Home Agent (HA), and Foreign Agents (FA) respectively. The base stations are connected through a network of core routers *c0*, *c1*, *c2*, and *c3*. We simulated traffic from correspondent nodes *CN0*, *CN1* and *CN2* to three mobile nodes *MH0*, *MH1* and *MH2* respectively. The correspondent nodes also act as edge routers and connected to the network through the core router *c0*, while the mobile nodes *MH0*, *MH1* and *MH2* are connected to *HA*, *FA1* and *FA2* respectively. During the simulation, *MH1*and *MH2* remain connected with their respective base station, whereas *MH0* moves from *HA* to *FA1* and then to *FA2*. The links *l0*, *l1*, *l2*, *l3*, *l6*, *l7*and *l8* are configured as links whose bandwidth is 10 Mbps and propagation delay is 5 ms. Links *l4* and *l5* are configured as links whose bandwidth is 300 Kbps and propagation delay is 5 ms. The links *l4* and *l5* are the bottleneck links carrying handover traffic and are the focus of our investigation.

In this simulation we want to study the impact of mobile IP handover on the adaptive TCP connections. Hence, we generated three FTP traffic – one each from *CN0* to *MH0*, from *CN1* to *MH1* and from *CN2* to *MH2*. The corresponding TCP flows are named as flow0, flow1 and flow2, respectively. The rate adjustment algorithm at the TCP sender continuously adjusts the sending rate to the available bandwidth in the network throughout the connection lifetime. Details of configuration for the access network and the DiffServ domain is given in Chapter 4.

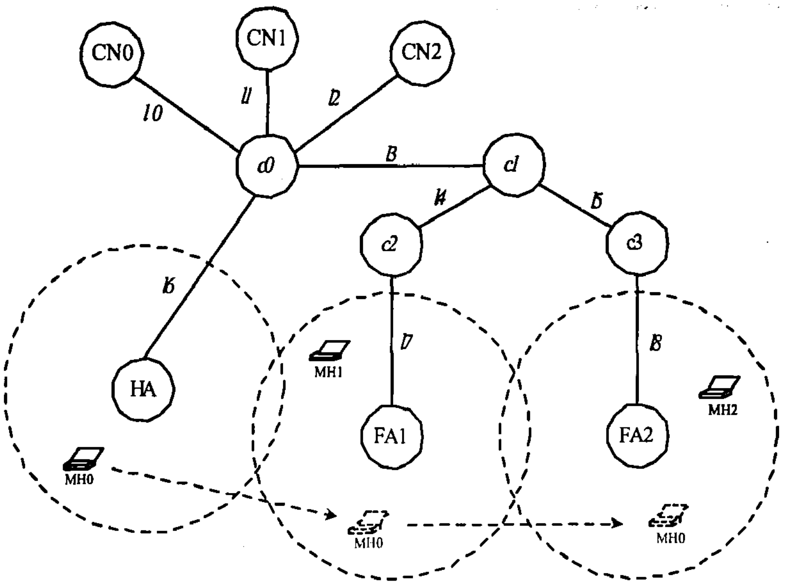


Figure 3.1 Basic Simulation Scenario with Two CN and Two Mobile Nodes

When mobile nodes move from one base station to another the traffic load varies on the links in the communication path. We call the flows that experience handover as *handover flows*, while the flow from a correspondent node to a mobile node that remains connected with its original base station as a *local flow* of the base station. Since the number of handover flows is usually unpredictable at any given time, the handover flows are expected to cause performance degradation in the existing flows. We organized the simulation into two cases. In case 1, flow1 from CN1 to MH1 is a local flow in the cell FA1 and flow2 from CN2 to MH2 is a local flow in the cell FA2. Flow0 from CN0 to MH0 is a handover flow from the cell HA to the cell FA1 and then to FA2. To keep focus on the main issues we only consider AF1, for the simulation of case1. Hence, the handover flow and the local flows are all marked as AF1. The bandwidth of each flow on the bottleneck link is show in Figure 3.2; Figure 3.2(a) shows the interaction of the handover and local flows along the path to FA1, and Figure 3.2(b) shows the interaction of the handover and local flows along the path to FA2. The bandwidth values presented in the above figures are computed as the percentage of the total bandwidth of the bottleneck link.

The simulation result shows that the effect on the existing flows in the new cell is in proportion to the bandwidth demand of the handover flows when both flows share the same DiffServ class. We found that the local flows in the new cell experience bandwidth shortfall due to sharing the available bandwidth on the bottleneck links with the handover flows.

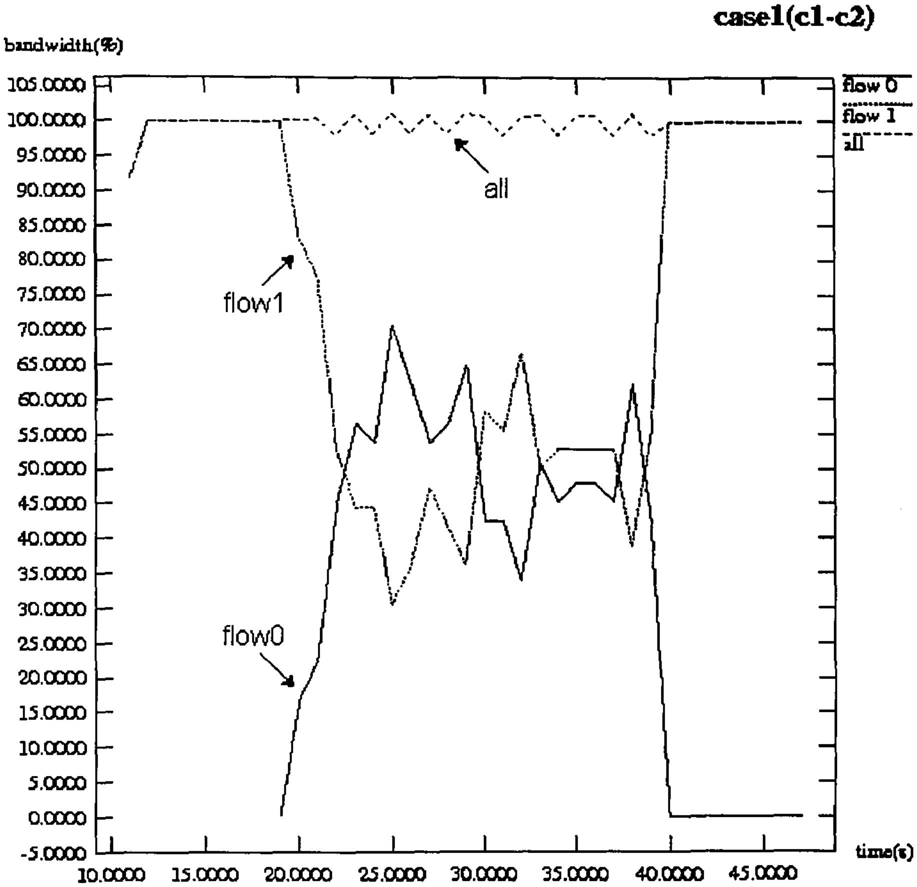


Figure 3.2(a) Interaction of Handover and Local Flows Along the Path to FA1

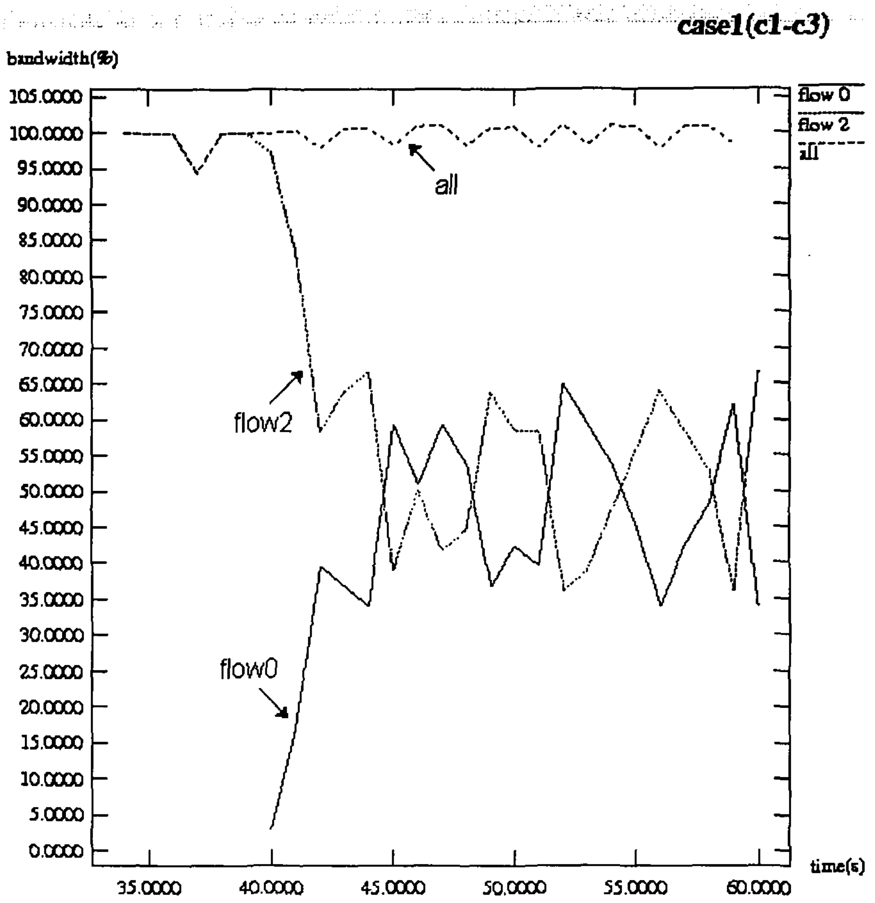


Figure 3.2(b) Interaction of Handover and Local Flows Along the Path to FA2

Figure 3.2 Result of Case1: Handover and Local Flows Sharing AF1

It is obvious from the simulation that handover flow competes for bandwidth on the bottleneck link with the local flows after the handover in the new cell. This causes the service degradation of the local flows. To avoid this situation, a possible solution is to remark the handover flow to a lower service level to protect the bandwidth share of the local flows. We study the effects of this scheme by simulating case 2 whose result is shown in Figure 3.3. The simulation settings for case 2 are almost the same as those for case1, except that we introduced remarking of the handover flow as AF2, a lower service level as compared to the local flow. Remarking is done at the home agent. A Weighted Round Robin (WRR) scheduler is used to schedule different services along the bottleneck link between core routers. We assigned twice the bandwidth of AF2 to AF1 by setting the weight of AF1 twice as much as that of AF2. Figure 3.3 shows that the bandwidth assignment for the local flow can be protected if the handover flows are remarked to a different traffic class and the weight for the WRR scheduler is set properly after the handover.

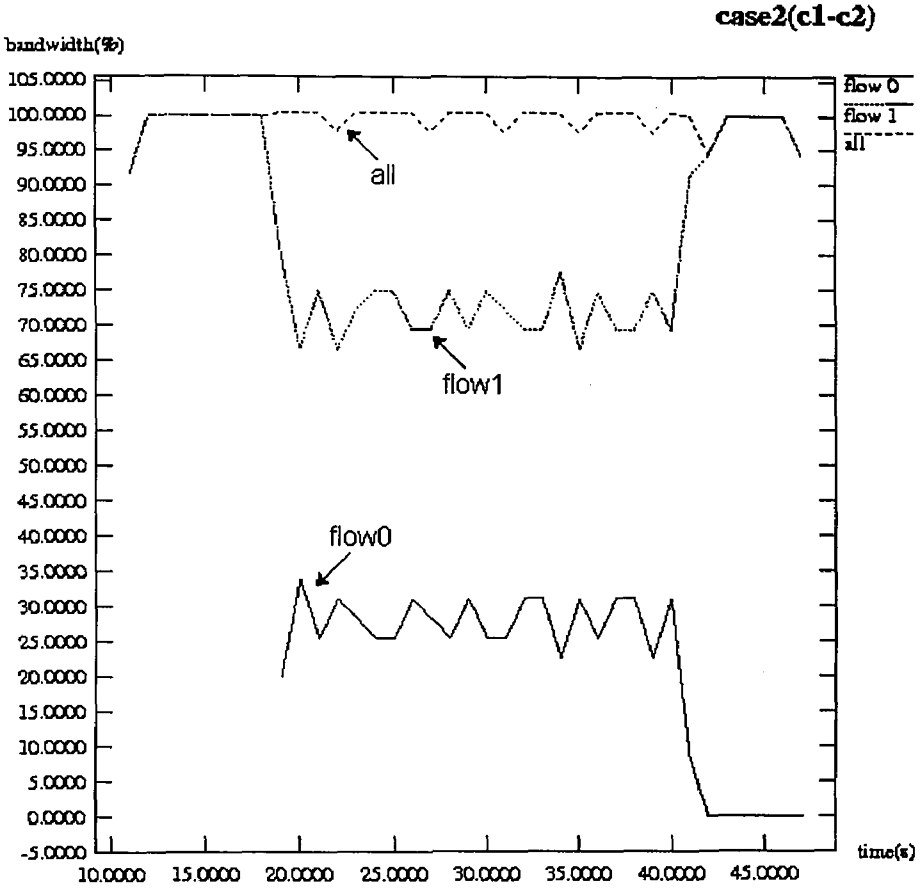


Figure 3.3(a) Interaction of Handover and Local Flows Along the Path to FA1

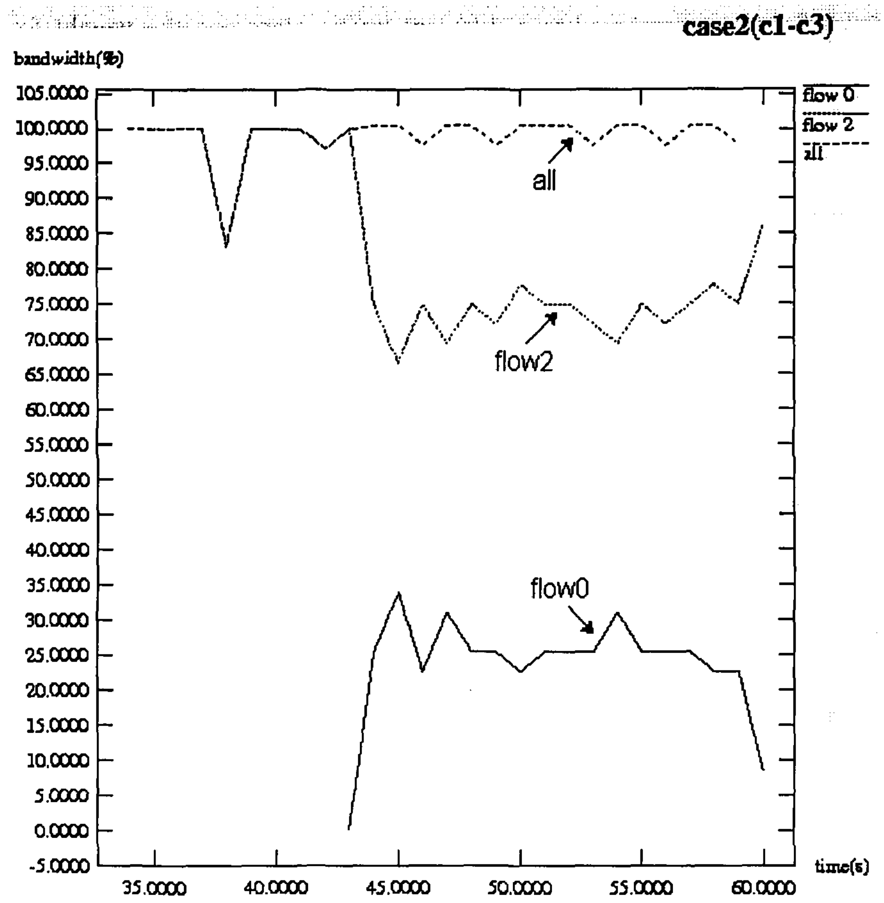


Figure 3.3(b) Interaction of Handover and Local Flows Along the Path to FA2

Figure 3.3 Result of Case2: Handover and Local Flows in AF2 and AF1 Respectively

This simple simulation illustrates the impact of handover on DiffServ flows and the benefit of employing a protection scheme by isolating handover and local flows in different AF classes. In the following sections we discuss a dynamic bandwidth management scheme based on the above principle.

3.2 Dynamic Bandwidth Management Algorithm

3.2.1 Overview

We consider two service classes implemented as AF1 and BE in the DiffServ-enabled mobile wireless access network. The reason for considering a single high priority AF1 service is to develop the idea that can be extended in the future for multiple service class, in addition AF1 provides qualitative assurance that can be used for a wide variety of applications and is easy to manage with less stringent requirements. We also assume that AF2 is also implemented as an auxiliary service class for AF1 to accommodate AF1 handover flows when those flows cannot be marked as AF1 in the new cell. As we have analyzed in the last section, handover flows compete with local flows for bandwidth, which causes the local flows to lose some of their assigned bandwidth. In this section we develop the bandwidth management algorithms for handover flows to achieve the following two goals. First, an AF1 flown should either continue with the same service level after handover, or in case of lack of available bandwidth, should receive better than best-effort service. In other words, the AF1 service is either continued or downgraded gracefully after the handover. Second, handover flows should not cause huge disruption to the service level of local flows. The second objective can be achieved to some extent with over provisioning because of unpredictably large number of flows handover caused by either crowd movement or by the movement of few users with huge application requirements.

The main idea of our algorithm is to assign traffic to different service classes so that local flows are protected and at the same time handover flows are still admitted in the new cell. The tradeoff of this idea is that some of the flows will lose their service guarantee to an affordable extent temporarily. We developed admission control algorithm for handover flows that outlines how to admit handover flows in a new cell with or without remarking to AF2. We also developed a service upgrade algorithm that is periodically invoked to decide about upgrading an AF2 flow. The algorithm selects the flows for service upgrade (restoring AF1 marking) by using an AF2 flow. The algorithm selects the flows for service upgrade (restoring AF1 marking) by using a priority system. We also propose a weight adjustment algorithm, including weight stealing and weight returning, for dynamic bandwidth assignment to AF2 service class by taking bandwidth from BE so that AF2 is assured higher service level than BE. The algorithm is invoked on a larger time scale by following the trend of AF2 operating under high utilization at lower than BE bandwidth. This algorithm allows the operators to provision smaller bandwidth to AF2 initially and then increase its share as the demand grows over a long period, which indicate higher level of sustained utilization.

3.2.2 Admission Control for Handover Flows

In the mobile wireless access network, admission control includes the admission control for originating flows and the admission control for handover flows. In this thesis we focus on the admission control of handover flows. Admission control for both originating and handover flows deal with both radio and network resources. In our study we assume that radio resource is available for the handover flow. This means that either the radio data rate is significantly more than the bottleneck link data rate (which is the case for WLAN) or the admission control on radio resource precedes the admission control on the network resources. Since we consider two service classes AF and BE here, whereas AF is the qualitative service while BE is not, our admission control algorithm is applicable only for the handover AF flows. In this context handover flows refer to handover AF flows. The aim for admission control is to provide as much as possible the contracted QoS guarantees to handover flows in the new cell; and at the same time maintain QoS guarantees for local flows. Further, in case of lack of resources along the new path, the handover flow QoS should be degraded gracefully, that is it receives no worse than best-effort service. The algorithm tries to meet the objectives over large time scale, while it makes necessary tradeoff in small time scale.

Considering the Mobile IP handover of the DiffServ flows scenario in Section 3.1, we now introduce a service class AF2 lower than AF1 but better than BE. When bandwidth available to AF1 flows is not sufficient to meet the demands of handover flow, then the handover AF flows can be assigned to this service class so that local AF1 flows can be protected and handover flows can still be admitted even though they get a downgraded service. We initially assign traffic to AF1 and BE and during handover assign traffic to AF2 only if AF1 lacks the required bandwidth, AF2 serves as a transient class or a backup class for AF1 flows. AF2 is configured as a service level lower than AF1, but higher than BE. Thus we can define the QoS contract for the AF user as: the standard service level is AF1 and the bottom line or the worst-case service level is AF2. When enough bandwidth is available for the handover AF flows to a new cell for AF1, they are marked as AF1 and named AF1 flows; while when bandwidth provisioning is limited and not enough in the new cell some handover AF flows are marked as AF2 and consigned to AF2 queue. When more bandwidth becomes available for AF1, one or more handover AF2 flows can be selected for upgrading to AF1. In that case the chosen AF2 flows are remarked as AF1. In our algorithm we address the issues that related to the situation of downgrading and upgrading of handover flows.

For upgrading handover flows we consider two types of users: those with less tolerance to service downgrade – called as G1 users and the other with more tolerance to service downgrade called as G2 users. We assume two priority levels. The G1 users get higher upgrade priority in the admission process than G2 users.

Let us review the situation on the bottleneck link along the path to the new cell. Assume that the total bandwidth of the bottleneck link is BW_{total}, and the bandwidth allocation for AF1, AF2 and BE flows are BW_{AF1}, BW_{AF2} and BW_{BE} respectively, then:

$$BW_{total} = BW_{AF1} + BW_{AF2} + BW_{BE}$$

Let BWA_{total} be the available bandwidth on the bottleneck link, BWA_{AF1}, BWA_{AF2} and BWA_{BE} be the available bandwidth for AF1, AF2 and BE respectively, then:

$$BWA_{total} = BWA_{AF1} + BWA_{AF2} + BWA_{BE}$$

In this research we only consider TCP flows that go through traffic conditioning at the entrance to the DiffServ domain. The traffic conditioning process includes metering and policing using token bucket policer. We assume that a target rate is defined for each of the TCP flow as following:

T_i: Target rate for the i th AF1 flow;

T_j: Target rate for the j th AF2 flow;

Let the number of AF1 and AF2 flows served in the new cell be p and q respectively. Then we can have following expressions of the available bandwidth for AF1 and AF2:

$$BWA_{AF1} = BW_{AF1} - \sum_{i=1}^{p} T_i$$

$$BWA_{AF2} = BW_{AF2} - \sum_{j=1}^{q} T_j$$

Let us assume that at time t_0 a handover AF flow, f_{HO}, makes a request for handover into the new cell. Further assume that the flow f_{HO} belongs to G1 user. The marking of f_{HO} in the old cell is out of concern of the admission control algorithm. If the target rate of the handover AF flow is denoted by

T_{HO}, and the DiffServ marking for the flow is denoted by CP_{HO}, then admission control problem can be stated as: Given BWA_{AF1} and BWA_{AF2}, determine CP_{HO} when the flow target rate is T_{HO}. We find the following three cases for the admission control algorithm.

- Case-1. Handover flow can be marked as AF1.

  If $T_{HO} \leq BWA_{AF1}$

  Then $CP_{HO} = DSCP\ for\ AF1$

  and $BWA_{AF1} = BWA_{AF1} - T_{HO}$ (1)

  This is the best case where the available bandwidth for AF1 is no less than the target rate of the handover AF flow. The handover flow is marked as AF1 and the available bandwidth for AF1 is updated with Eq.1.

- Case-2. Handover flow can be marked as AF2.

  If $T_{HO} > BWA_{AF1}\ and\ T_{HO} \leq BWA_{AF2}$

  Then $CP_{HO} = DSCP\ for\ AF2$

  and $BWA_{AF2} = BWA_{AF2} - T_{HO}$ (2)

  In this case, the available bandwidth for AF1 is not enough for the handover flow. If the algorithm marks the handover flow as AF1 it will steal bandwidth from the local AF1 flows causing their service level to decrease. Since the available bandwidth for AF2 is enough for the handover flow the algorithm should mark the handover flow as AF2 and update the available bandwidth for AF2 as given in Eq.2. The benefit is quite obvious here. First, local AF1 flow is fully protected. Second, local AF2 flows (if there is any) are not affected either. Third, the handover flow has a better chance to achieve its target rate after the handover even though it gets worse service than AF1 during congestion.

Case-3: Handover flow can be marked as either AF1 or AF2

If $T_{HO} > BWA_{AF1}\ and\ T_{HO} > BWA_{AF2}$ then the flow can be considered for marking either as AF1 or AF2.

In this case we introduce the concept of *penalty* to evaluate the relative merit of marking either as AF1 or AF2. We discuss the penalty and its evaluation in case 3 in the following section.

3.2.3 Discussion for Admission Control Case-3

Before discussing our solution for case-3, we explain the dynamics of bandwidth sharing of TCP flows. In [27], Seddigh et al presented through a performance study they conducted on an experimental testbed that the TCP flows get fair degradation in their target rates in the under-provisioned situation. Here fair degradation means proportional degradation. The situation depicted in case 3 resembles the under-provisioning situation; hence their conclusion is applicable to the flows in this case.

We conducted a simulation to understand the nature of downgrade caused by handover to local flows. The topology for this simulation is similar to that given in Figure3.1. We have three AF flows in the test: two local flows flow1 (from CN1 to MH1) and flow2 (from CN2 to MH2) in the cell FA1, and one handover flow – flow0 (from CN0 to MH0) moving from the cell HA to the cell FA1. Traffic settings are given in Table3.1.

Flow number	Source	Destination	Flow Class	DiffServ marking	Target rate (kbps)
0	CN0	MH0	Handover	AF1	x
1	CN1	MH1	Local	AF1	100
2	CN2	MH2	Local	AF1	100

Table 3.1 Traffic Settings for Test1.

In the test, the total bandwidth of the link is 300 kbps and the bandwidth allocation for AF1 class is 70% of the link bandwidth that is 210 kbps. Local flows –flow1 and flow2 have fixed target rate of 100 kbps each, while handover flow –flow0 makes handover to the new cell with a variable target rate x that we can set to different rates. We call a flow with small target rate as thin flow and a flow with large target rate as a fat flow. We set the target rate of handover flows to 15, 30, 45, 60, 75, 90 and 105 kbps simulating from thin to a fat flow situation. The target rates for the handover flow are expressed as 5%, 10%, 15%, 20%, 25%, 30% and 35% of the bottleneck link bandwidth. For every simulation run, we monitor and record the average throughput of each flow (flow 0, 1 and 2) on the bottleneck link between the core routers c1 and c2 along the path to FA during the time period when mobile node MH0 is registered with the base station FA1. The result is shown in Figure3.4.

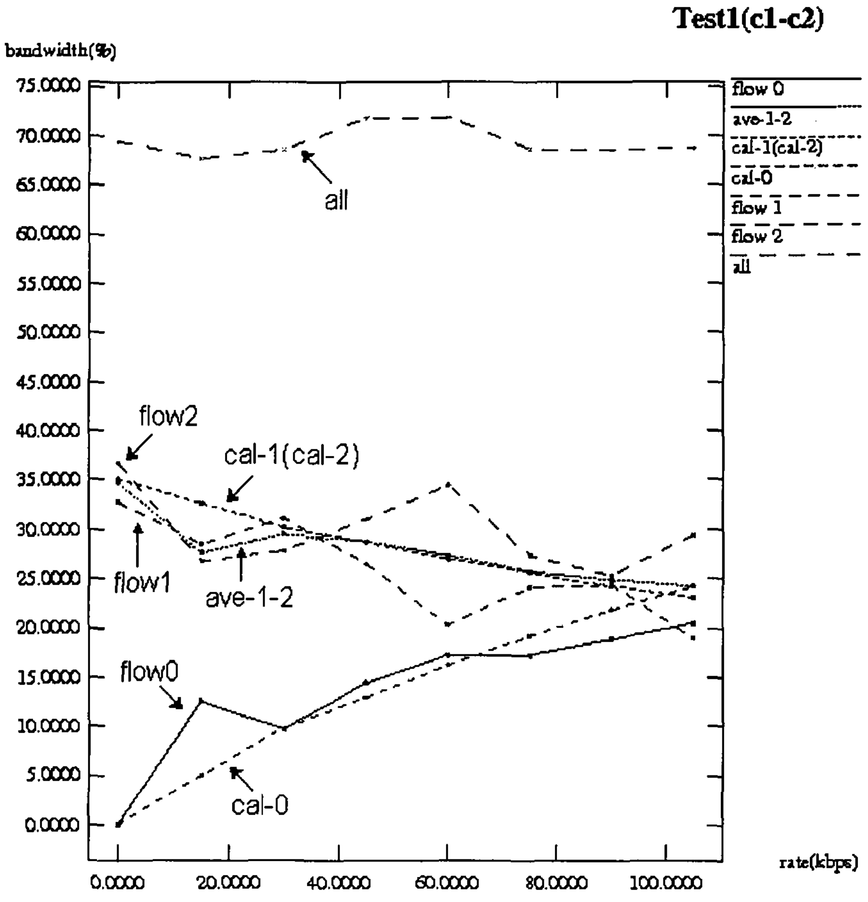


Figure 3.4 Results for Testl

Figure3.4 shows the throughput expressed in terms of the percentage of the bottleneck link bandwidth. We use $B_0(x)$, $B_1(x)$, and $B_2(x)$ to denote the throughput of flow 0, 1 and 2 respectively when the target rate of flow 0 is set to x kbps. Ave-1-2 is the average bandwidth for the local flows, which is calculated as:

$$B_{ave\text{-}1\text{-}2}(x) = \frac{1}{2}[B_1(x) + B_2(x)]$$

We also calculate cal-1, cal-2 and cal-0 under the assumption that flows get proportional degradation under the under-provisioning scenarios, using following equations:

$$B_{cal\text{-}1}(x) = B_{cal\text{-}2}(x) = \frac{100}{100+100+x} \times 70\%$$

$$B_{cal-0}(x) = \frac{x}{100 + 100 + x} \times 70\%$$

We can see from Figure 3.4 that except the very thin flow (target rate is set to 15 kbps) that causes extreme bandwidth shortfall of local flows, other sizes of flows all go approximately with the assumption. The exceptional case is left for further investigation. Our simulation corroborates the results of [27] and shows that flows get proportional degradation in the under-provisioning situation.

Since neither available bandwidth for AF1 or AF2 is enough to meet the target rate of the handover flow in case 3 (an under-provisioning situation), regardless whether the handover flow is assigned to AF1 or AF2 all flows including handover and local flows within that class will be penalized by missing their target rates. We define the penalty of a flow as the amount by which it misses the target rate, which is calculated as the difference between the target and the estimated bandwidth it is expected to receive based on the calculation of the proportional bandwidth using above formula. We calculate the total penalties for both AF1 and AF2 assuming the handover flow is assigned to each of them respectively. Hence, we decide about the assignment of the handover flow to either AF1 or AF2 based on whichever assignment gives the total minimum penalty. We define the algorithm formally below.

Assume P_i is the percentage by which the i-th local AF1 flow misses its target rate, P_{HO} is the percentage by which the handover flow misses its target rate, and A_i is the simulated average bandwidth the i-th local AF1 flow and A_{HO} were expected to receive if the handover flow was marked as AF1. We further assume that P_{AF1} is the total penalty of all local AF1 flows and the handover flow if the handover flow is marked as AF1. Then we can derive the following formulae if the handover flow is marked as AF1:

$$A_i = \frac{T_i}{\sum_{i=1}^{p} T_i + T_{HO}} \times BW_{AF1} \tag{3}$$

$$P_i = \frac{T_i - A_i}{T_i} = 1 - \frac{A_i}{T_i} \tag{4}$$

Substituting A_i by (3) in (4) and simplifying it, we can get

$$P_i = 1 - \frac{BW_{AF1}}{\sum_{i=1}^{p} T_i + T_{HO}} \tag{5}$$

Let $\alpha_{AF1} = \frac{BW_{AF1}}{\sum_{i=1}^{p} T_i + T_{HO}}$,

Then Eq.5 can be simplified as $P_i = 1 - \alpha_{AF1}$

Similarly $A_{HO} = \frac{T_{HO}}{\sum_{i=1}^{p} T_i + T_{HO}} \times BW_{AF1}$

and $P_{HO} = \frac{T_{HO} - A_{HO}}{T_{HO}} = 1 - \frac{A_{HO}}{T_{HO}}$

$$= 1 - \frac{BW_{AF1}}{\sum_{i=1}^{p} T_i + T_{HO}} = 1 - \alpha_{AF1}$$

So $P_{AF1} = \sum_{i=1}^{p} P_i + P_{HO} = (p+1)(1 - \alpha_{AF1})$ (6)

Similarly, assume that P_j is the percentage by which the j-th local AF2 flow misses its target rate, A_j is the estimated average bandwidth the j-th local AF2 flow is expected to receive, and P_{AF2} is the total penalty of all local AF2 flows and the handover flow if the handover flow is marked as AF2. Then we can derive the following formulae if the handover flow is marked as AF2:

$$A_j = \frac{T_j}{\sum_{j=1}^{q} T_j + T_{HO}} \times BW_{AF2}$$

$$P_j = \frac{T_j - A_j}{T_j} = 1 - \frac{A_j}{T_j}$$

$$= 1 - \frac{BW_{AF2}}{\sum_{j=1}^{q} T_j + T_{HO}} \qquad (7)$$

Let $\alpha_{AF2} = \frac{BW_{AF2}}{\sum_{j=1}^{q} T_j + T_{HO}}$,

Then Eq.7 can be simplified as $P_j = 1 - \alpha_{AF2}$

Similarly $A_{HO} = \frac{T_{HO}}{\sum_{j=1}^{q} T_j + T_{HO}} \times BW_{AF2}$

and $P_{HO} = \frac{T_{HO} - A_{HO}}{T_{HO}} = 1 - \frac{A_{HO}}{T_{HO}}$

$$= 1 - \frac{BW_{AF2}}{\sum_{j=1}^{q} T_j + T_{HO}} = 1 - \alpha_{AF2}$$

So $P_{AF2} = \sum_{j=1}^{q} P_j + P_{HO} = (q+1)(1 - \alpha_{AF2})$ (8)

We can use Eq.6 and Eq.8 to compute the total penalty if the handover flow is assigned to either AF1 or AF2 respectively. To safeguard AF1 flows being penalized with a too close margin to the penalty of an AF2 flow, we introduce a factor β to be the difference between P_{AF1} and P_{AF2}. Hence we decide that if $P_{AF1} \le \beta \cdot P_{AF2}$, then the handover flow is assigned to AF1, and the available bandwidth for AF1 is updated to be zero since there is no more bandwidth remains available for AF1 after the assignment. Otherwise it is assigned to AF2 and the available bandwidth for AF2 is updated to be zero since there is no more remaining available bandwidth for AF2 after the assignment. In any case, the local flows are adversely affected by the handover flow and none of the local and the handover flows could achieve their target rates. However, this approach minimizes the total penalty within a class.

Let us discuss the effect of factor β on the bandwidth distribution. It is an important parameter that provides control in achieving the level of control for af1 flows. When β=1, we it is equally likely that the handover flow is assigned to AF1 or to AF2. When β<1, it is more likely that the handover flow will be assigned to AF2, which gives more protection to local AF1 flows and consequently shift more punishment to AF2 flows. On the other hand, when β>1, it gives more protection to local AF2 flows and shift more punishment to AF1 flows. Hence, we suggest that $\beta \leq 1$ for normal operation. Figure 3.5 shows β*P_{AF2} curves for different values of β (0.25, 0.5, 1.0, 1.5) when PAF2 varies from 0 to 2.5. The region under the curve are the values of β*P_{AF2} (P_{AF1}) for which the handover flow will be assigned to AF1, and the region above the curve is where the handover flow will be assigned to AF2. The Figure shows that as we increase the value of β the region where the handover flow should be marked as AF1 expands, which indicates that the protection for AF1 will decrease.

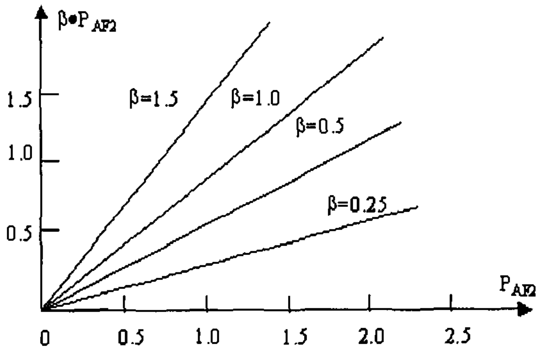


Figure 3.5 Effect of factor β

3.2.4 Summary of the Admission Control Algorithm

Figure3.6 shows the admission control algorithm pseudo code.

Admission control algorithm

```
upon the arrival of the handover AF flow
IF available bandwidth for AF1 of the link $BWA_{AF1}$ is no less than the target rate of the handover AF flow (case-A)
        mark the handover AF flow as AF1
        use Eq. 1 to update the available bandwidth for AF1
ELSE IF available bandwidth for AF2 of the link $BWA_{AF2}$is no less than the target rate of
```

```
  the    handover AF flow (case-B)
        mark the handover flow as AF2
        use Eq.2 to update the available bandwidth for AF2
        ELSE  (case –C)
          use Eq.6 to calculate the total penalty $P_{AF1}$ for local AF1 flows and the handover
             flow if the handover flow is marked as AF1
          use Eq.8 to calculate the total penalty $P_{AF2}$ for local AF2 flows and the handover
             flow if the handover flow is marked as AF2
          IF $P_{AF1} \le \beta \cdot P_{AF2}$
             mark the handover AF flow as AF1
             $BWA_{AF1} = 0$
           ELSE
             mark the handover AF flow as AF2
             $BWA_{AF2} = 0$
            ENDIF
        ENDIF
  ENDIF
RETURN
```

Figure 3.6 Admission Control Algorithm Pseudo-code.

During the admission control process, if there are more than one handover AF1 flows making request for handover to a cell, then they should get equal chance to be admitted as AF1 flow. However, they are queued in the request buffer and served in FIFO order. When bandwidth is limited on the bottleneck link only some of the flows are assigned to AF1, while others are assigned to AF2. We provide a mechanism for upgrading a handover flow from AF2 to AF1 whenever AF1 has residual bandwidth available. We discuss the service-upgrading algorithm in detail in the next section.

3.2.5 Service Level Upgrade Algorithm

Although some handover AF1 flows can be admitted at downgraded service level by being assigned to AF2 when congestion happens, they should be able to upgrade to the standard service level whenever there is enough bandwidth available for AF1 on the bottleneck link. We introduce Service Level Upgrade algorithm in this section to achieve the above objective.

We need to consider several issues in designing service upgrade algorithm. For example, which flow to select for upgrade, when to check for upgrade, etc. We assume an upgrade priority associated with every flow. Flows with G1 priority level have stricter requirement for their service level than flows with G2 priority. For flows with the same priority but with different target rates, we suggest that the flow with the highest target rate should be considered first. The reason for this is that a flow with high target rate is more likely to wait in the service upgrade queue, which is further discussed below.

When some local AF1 flows leave a cell, they release their bandwidth, which causes the increase of the available. Assume that the number of AF1 flows in the cell at any time *t* is *p*, and the target rate of the flow leaving the cell is T_{HO}, then

$$BWA_{AF1} = BWA_{AF1} + T_{HO}$$

We consider the flows in AF2 available for service upgrade are arranged in a priority queue. There are two independent First In first Out (FIFO) queues corresponding to LOW and HIGH upgrade priority, which are served by a priority scheduler, as illustrated in Figure 3.7.

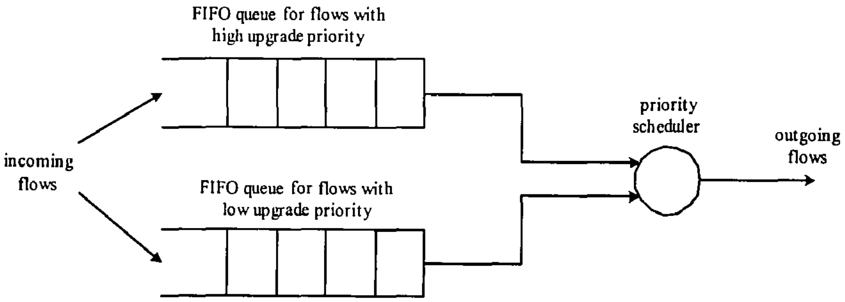

FIFO queue for flows with high upgrade priority
incoming flows
FIFO queue for flows with low upgrade priority
priority scheduler
outgoing flows

Figure 3.7 Schedule Schemes for Flow Upgrading

Let T_i be the target of the *i*-th outgoing flow of the priority scheduler where *i* starts from 1, details of the upgrading process can be expressed as the pseudo-code in Figure3.8:

```
(1)WHILE $BWA_{AF1} \geq T_i$
(2) DO remark flow $i$ as AF1; $BWA_{AF1} = BWA_{AF1} - T_i$; $i = i + 1$
(3) RETURN
```

Figure 3.8 Pseudo-code of Service Upgrade Algorithm

A potential problem for the service upgrade algorithm concerns with the different target rates of the AF2 flows in the cell. When a AF2 flow with high upgrade priority goes to the front of its corresponding queue such that BWA_{AF1} is less than its target rate T_i, then all the AF2 flows are blocked, even though some of them may have smaller target rates less than BWA_{AF1}. This situation can adversly affect the efficiency of the service upgrade algorithm. In order to avoid this situation, the flows in a FIFO queue are sorted in the ascending order of their target rates. Thus flows with low target rates having more chance for upgrade precede the flows with high target rates having less chance for upgrade. We need further mechanism to deal with starvation of high rate flows that is beyond the scope of this thesis.

If the FIFO queue for high upgrade priority is always busy with the flow, then the flows with low upgrade priority never get a chance to be served as AF1. A possible remedy is to also consider sojourn time in the upgrade decision. We define sojourn time of a flow in a cell as the time elapsed since the flow enters the cell. To avoid the above situation, a threshold for the sojourn time can be set up for the flows in the queue corresponding to LOW upgrade priority. When a flow's sojourn time exceeds the threshold, its priority is changed from low to high and it is moved to the high upgrade FIFO queue.

We propose a mechanism to periodically invoke the service upgrade algorithm. We set up a timer with the time period of t_u to periodically measure the available bandwidth of AF1, which can be estimated by measuring the bandwidth consumed by green packet and subtracting that from the bandwidth provisioned for AF1. We can maintain only a single timer for a cell. Figure 3.9 shows the time line of service upgrade invocation. The time t_0 is the beginning of the upgrade process. After every time t_u the service upgrade algorithm is invoked, which computes the available bandwidth of AF1 along the bottleneck links and upgrade the eligible flows from AF2 to AF1 following the algorithm given in Figure 3.8. It then resets the timer to expire after t_u. If a handover occurs before the timer expires, for example at time t_1 in Figure 3.9, then the service upgrade algorithm can be executed as a part of the handover process, hence the timer can be reset. This is because the time difference to the next t_u as was originally set may not be enough to create significant change in the bandwidth availability of AF1.

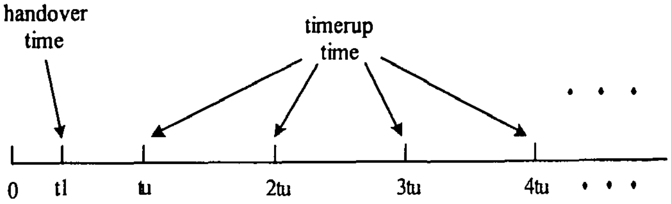


Figure 3.9 Timer Arrangements and Handover Time

The trigger for the service upgrade algorithm can be described as follows:

```
Trigger
IF (HO_flag)
    upgrade ( )
    HO_flage =false
ENDIF
RETURN
```

3.3 System Overview

In this section we give an overview of the system implementing that identifies the issues related to the admission control algorithm for handover flows that includes a discussion on relevant issues. When a mobile node comes to a new cell, it first discovers its FA and HA and then registers with them and then the Mobile IP tunnel is established. In the registration process, when MN sends registration to its FA, it also triggers the admission control algorithm that will be performed by the FA. According to the algorithm, FA should keep the information of the available bandwidth for AF1 and AF2 along the bottleneck link. To get this information FA needs to know the bandwidth allocation for AF1 and AF2 along the bottleneck link and the information of bandwidth utilization of each service class. This information comes from the core network periodically and saves in the FA. The FA can find the bottleneck link that is beyond the scope of this thesis. When MN registers with FA, it also sends information of its traffic source, target rate, etc. The FA performs the proposed admission control algorithm and decides whether the flow needs to be remarked. If it decides for remarking, then it will signal the HA, remarking DSCP to AF2. On contrast, no remarking is needed and flows remain AF1. Then FA updates the information for available bandwidth according to the marking of the handover flow. If it is marked as AF1, then the available bandwidth for AF1 is decreased by the target rate of the handover flow and the available bandwidth for AF2 remains the same. Otherwise, if the handover flow is remarked as AF2, then the available bandwidth for AF2 is decreased by its target rate and the available bandwidth for AF1 remains the same.

For the service upgrade algorithm, a timer is installed in FA; every time the timer expires FA sends request to the core network to update the information of available bandwidth for AF1 and AF2. Alternatively, when handover happens, the service upgrade algorithm is also triggered. After the upgrading process, FA also upgrades its information for available bandwidth according to the upgrading information.

3.4 Provisioning

We introduced AF2 as a transient class that accommodates handover flows that cannot be admitted in AF1 and tries to provide better than best-effort service for handover flows when bandwidth is not sufficient for AF1. Under a rightly provisioned system AF2 should have few handover flows. Hence, we suggest that initially AF2 should be provisioned with small amount of bandwidth, even smaller than BE, so that AF1 can be provisioned with adequate amount of bandwidth. However, when utilization of AF2 is high and utilization of BE is low, AF2 flows may get bandwidth much less than their target rate while BE flows may get bandwidth quite close to their target rate. This situation may result in AF2 flows getting worse than best-effort service, which contradicts with the goal of our dynamic bandwidth management scheme. To avoid this situation, we introduce weight adjustment algorithm to dynamically adjust bandwidth allocation for different service class according to the bandwidth utilization of different service class.

3.4.1 Weight Adjustment Algorithm

We use Weighted Round Robin (WRR) scheduler in our simulations to implement the DiffServ classes. Given BW_{totoal} as the total link bandwidth, W_{AF1}, W_{AF2} and W_{BE} as weights for AF1, AF2 and BE respectively, then the bandwidth allocation for corresponding service is given as:

$$BW_{AF1} = BW_{total} \times \frac{W_{AF1}}{W_{AF1} + W_{AF2} + W_{BE}}$$

$$BW_{AF2} = BW_{total} \times \frac{W_{AF2}}{W_{AF1} + W_{AF2} + W_{BE}}$$

$$BW_{BE} = BW_{total} \times \frac{W_{BE}}{W_{AF1} + W_{AF2} + W_{BE}}$$

The mobile wireless access network only provides two class of services, AF1 and BE. AF2 is a transient class for AF1, which is implemented to accept more handover AF1 flows when bandwidth is limited. When a handover AF1 flow cannot be admitted as AF1 in the new cell, we assign it to AF2. In the DiffServ provisioning scheme AF2 can be allocated a small amount of bandwidth, even lower than BE. Since AF2 is designed to provide better than BE service to handover AF1 flows, we need to define a mechanism to enhance bandwidth allocation for AF2 to surpass the BE allocation. Alternatively, AF2 can be allocated higher than BE bandwidth but it makes the bandwidth allocation less efficient because AF2 utilization can be low at the beginning. Furthermore, it does not preclude the need for future enhancement to AF2 bandwidth. In order to understand the need for bandwidth enhancement of AF2 consider the situation when a crowd moves into a cell with AF1 flows. In this case a large number of flows may be assigned to AF2 service class, which may bring down the bandwidth available to AF2 flows even lower than BE causing a risk for AF2 flows to receive lower than BE service. This situation can be handled if we define a bandwidth enhancement scheme for AF2 class. However, any bandwidth enhancement scheme for AF2 class, including the one proposed below, must be carefully evaluated for their overhead and should be judiciously invoked to avoid instability in service provisioning.

We propose a weight adjustment scheme for AF2 and BE service classes to achieve bandwidth adjustment to those classes. For example, when AF2 is crowded with the handover flows running the risk of AF2 flows getting lower than BE service, then the bandwidth can be stolen from BE and allocated to AF2. This can be achieved by increasing the weight of AF2 and simultaneously decreasing the weight of BE by the same amount.

We also need to define a bandwidth return procedure from AF2 to BE whenever the utilization of AF2 bandwidth is significantly lower than the BE utilization. To guarantee basic services for BE and provide some level of protection for AF2, we give AF2 and BE minimum weight limit as WL_{AF2} and WL_{BE} respectively. The process here is to take the weight from BE and give it to AF2, until there is enough bandwidth for the handover AF flows or until the minimum weight for BE is reached. In this process AF2 steals bandwidth from BE. On the other hand, when the utilization of the bandwidth of AF2 is significantly lower than BE, it returns some weights to BE.

For the basic quantum of weight moved from AF2 to BE and vice versa, we consider a quantum δ here in our algorithm. Then the weight adjustment algorithm can be expressed as a combination of the following weight stealing and weight-returning algorithm:

- **Weight stealing algorithm**

```
IF (WeightStealing _flag)
    IF $W_{BE} > WL_{BE} + \delta$
        $W_{BE} = W_{BE} - \delta$;
        $W_{AF2} = W_{AF2} + \delta$;
    ENDIF
ENDIF
RETURN
```

- **Weight returning algorithm**

```
IF (WeightReturning _flag)
    IF $W_{AF2} > WL_{AF2} + \delta$
        $W_{AF2} = W_{AF2} - \delta$;
        $W_{BE} = W_{BE} + \delta$;
    ENDIF
ENDIF
RETURN
```

Chapter 4

Simulation Setup and Result Analysis

We have developed a simulation model of a mobile wireless access network configured with three base stations. We performed our simulation in network simulator ns-2 [32]. In this chapter we will discuss the results of our simulation of the admission control algorithm, as described in Chapter 3. We will first discuss the simulation setup in Section 4.1, and then we will present the results and their analyses in Section 4.2.

4.1 Simulation Setup

4.1.1 Topology

We used the network topology as shown in Figure 4.1 in our simulation. We used this topology to create four scenarios. In the following we describe main component of the network.

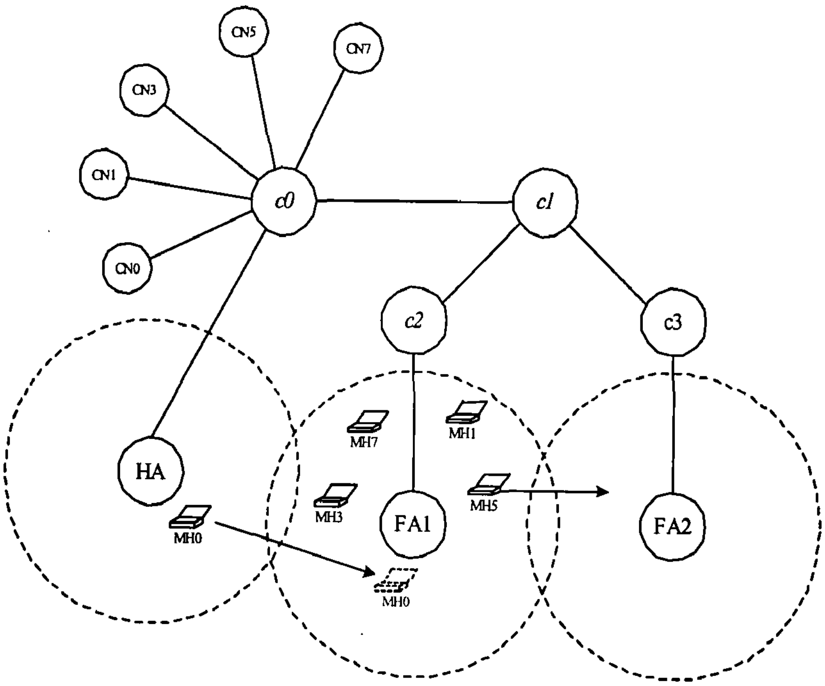


Figure 4.1 Network Topology in Our Simulation

In our simulation topology, CN0, CN1, … , are correspondent nodes. c0, c1, c2 and c3 are core routers in the DiffServ domain. HA, FA1, FA2 are base station nodes and MH0, MH1, … , are mobile hosts. All the correspondent nodes and the base station nodes are also configured as DiffServ edge nodes.

We have several kinds of links in this simulation topology, including core links, edge links and other links. To identify the links, we define link (x, y) as the link from node x to node y. Then link (r1, r2) and link (r1, r3) are our bottleneck links. The bandwidths of the links are both set as 300k bps and the packet processing delay is 5ms. The queue used for both these links are DiffServ type core queues using RED buffer type.

To simplify our simulation, we set the link from each correspondent node to its access router as a DiffServ-capable edge link. Then there is only one physical queue on the simplex edge link since we only configured one traffic flow from each correspondent node to its related mobile host. The bandwidth of the link is 10Mbps and the packet processing delay is 5ms. The queues used for the edge link are DiffServ type edge queues using RED buffer type. On the other direction we define a simplex link from the access router to every correspondent node, which is configured to have 10 Mbps bandwidth and 5ms packet processing delay and they use drop-tail queues. Similar with the links between the base stations and their correspondent access routers, we set the link from each base station to its access router as a DiffServ-capable edge link with the bandwidth set as 10Mbps and the packet processing delay as 5ms and using RED buffer type. On the other direction we define a simplex link from the access router to every base station, which is configured to have 10 Mbps bandwidth and 5ms packet processing delay and they use drop-tail queues. Link (c0, c1) is also a link between core routers. However, since it is not the bottleneck link, we simply configure it as a duplex link with 10Mbps bandwidth and 5ms packet processing delay and using drop-tail queues.

4.1.2 Traffic Generation

Unlike UDP, TCP is a transport protocol adaptive to the network congestion state. It explores the availability of bandwidth along the communication path and tries to greedily grab that bandwidth. For this characteristic of TCP we used it in our simulation to measure how does TCP react to the service upgrade and downgrade. We selected FTP as a representative of long flows, because as compared to other traffic types such as HTTP, FTP has larger object size that can be adjusted to generate longer TCP-connections. Further, it spends less time and packets on control and connection management than the time for the transmission of data. The long data transmission provides TCP's congestion control algorithm enough time to adapt to the available bandwidth in the network. The packet size we used for our simulation is 1040 KB. We establish TCP flows denoted by $flow_j$ between CN_j and MH_j.

4.1.3 DiffServ Configuration

To distinguish traffic with different Per Hop Behaviors (PHB) s, DiffServ code point (DSCP) is used and attached to a packet's IP-header. DiffServ PHBs are designed to provide qualitative service differentiation with no quantitative guarantees. As mentioned before, we have implemented three PHBs: AF1, AF2 and BE. The critical components in implementing these PHBs are scheduler and buffer management schemes, which we discuss in detail below.

4.1.3.1 Buffer Management

In a DiffServ domain, traffic goes to different queues based on their DSCPs within a router. Buffer management decides how to manage packets inside a single queue. Two buffer types used in our simulation are RED and drop-tail.

Core queue and edge queue are two DiffServ-capable queue types in our simulation. The core queue is usually deployed at the congestion point in the network. Since congestion only happen in the bottleneck link in our topology, only the bottleneck link is configured as a core queue. The edge queue is capable of performing policing, packet marking, etc. All queues between a host (e.g. CN or MH) and its access router in the direction of host-to-router are edge queues.

We implemented MRED [12] scheme for buffer management in AF1 and AF2. We implemented only two drop precedences for AF1 anAF2 PHBs. Each drop precedence is identified by a distinct DSCP – the packet with low drop precedence is called the green packet and the one with high drop precedence is called the red packet. Each PHB is implemented using a separate physical queue; while both AF1 and AF2 physical queues consist of two virtual queues each one for the in-profile (green) packets and the other is for the out -file (red) packets, as illustrated in Figure4.2. Different RED parameters are used for the virtual queues; causing red packets to be dropped more frequently than green packets. Similarly we configure AF1 MRED with higher minimum and maximum thresholds than AF2 MRED to drive AF2 packets to the congestion point before AF1 packets.

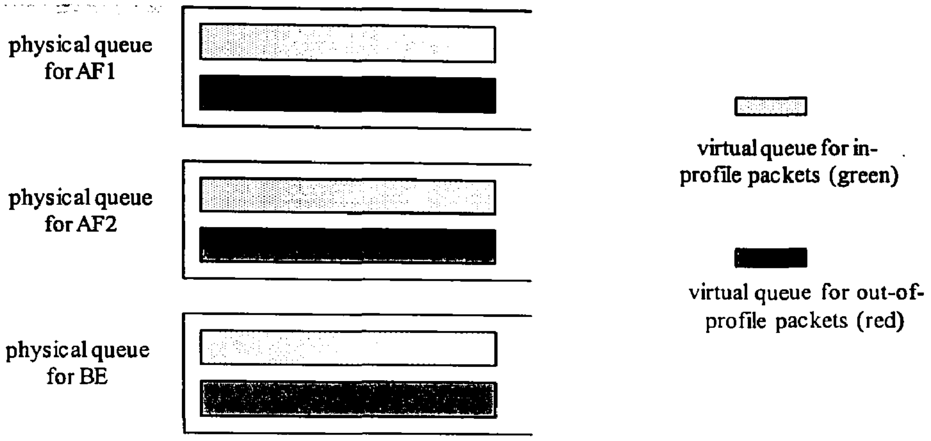

physical queue for AF1
physical queue for AF2
physical queue for BE
virtual queue for in-profile packets (green)
virtual queue for out-of-profile packets (red)

Figure 4.2 Queue Structure in Simulation

In MRED, drop probability for packets with different drop precedence (or described as packet color) have to be calculated independently for each drop precedence, thus multiple sets of RED thresholds need to be maintained – one for each drop precedence. Multiple Average Multiple Threshold (MAMT) scheme including RIO Coupled mode (RIO-C) and RIO de-coupled mode (RIO-D) is applied for calculating the average queue used for the drop decision. In the MRED scheme we implemented threshold parameters for packets of different colors are set to partially overlap. For example, following parameters are defined for MRED supporting two colored packets: in_min, in_max and in_prob to be the minimum and maximum thresholds of average queue length and drop probability of green packets respectively, out_min, out_max and out_prob to be the minimum and maximum thresholds of average queue length and drop probability of red packets respectively, and qlimit to be the maximum buffer size:

Service class Parameter	AF1	AF2	BE
In_min (in packets)	20	10	15
In_max (in packets)	40	20	15
In_prob	0.02	0.08	1.00
Out_min (in packets)	10	5	0
Out_max (in packets)	20	20	0
Out_prob	0.10	0.15	1.00
qlimit (in packets)	40	20	15

Table 4.1 Parameters for MRED in the Simulation

Except core and edge queues (at edge and bottleneck links) all other queues are configured as simple FIFO queues with drop tail buffer management policy, because they do not experience congestion and do not contribute to the QoS performance of DiffServ flows.

4.1.3.2 Queue Management

Queue Management deals with how to control scheduling between multiple queues. For proportional bandwidth distribution across multiple PHBs, Weighted Fair Queuing (WFQ) is quite popular. It is a flow-based scheduling algorithm that provides fair sharing of bandwidth among multiple flows in proportion to the assigned weights. For example, flows with higher weights will get more bandwidth than flows with lower weights. Similarly, in Weighted Round robin (WRR) scheduling scheme weights are also assigned to buffers to indicate their relative share of link access time, and buffers are given opportunity for link access in a round robin fashion one buffer at a time. Thus, WRR can be regarded as a variant of WFQ.

For convenience, we employed WRR as our scheduling mechanism to implement the three PHBs. Each physical queue is assigned a weight determined by the DiffServ provisioning scheme. The WRR scheduler is work-conserving which means that if a queue has no packets to send its turn will be handed over to the next queue. Thus, if a service class lacks packets, then other service classes will share the link capacity according to their weights.

4.1.3.3 Core Routers

The core routers in DiffServ are quite simple because the only function they perform is to assign incoming packets to the relevant virtual queue and physical queue as identified by the DSCP. In each core router, we have three physical queues refer to the AF1, AF2 and BE services and each of them has two virtual queues for the in-profile (green) packets and the out -profile (red) packets. Each physical queue and virtual queue combination correspond to a unique DSCP, thus we need five DSCP to implement the three PHBs – two each for AF1 and AF2 and one for BE. For each virtual queue, RED parameters are properly set, including minimum and maximum thresholds and drop probability.

4.1.3.4 Edge Routers

Edge routers perform more QoS functions than core routers in DiffServ. Besides the implementation of PHBs using schedulers and buffer management similar to core routers, Edge routers also perform traffic conditioning functions - policing, metering, and marking packets.

Edge routers deal with the individual user flows. They classify flows based on: source address, destination address and traffic type or any combination of them. After classification they mark the packets with the appropriate DSCPs. We configured edge routers with Token Bucket policers, which use Committed Information Rate (CIR), Committed Burst Size (CBS) and two drop precedences as the parameters. An arriving packet is marked with the lower drop precedence (green) if the token bucket has enough tokens for the packet; otherwise it is marked for higher drop precedence (red).

4.1.4 Statistics and Monitoring

To monitor and gather statistics of our simulation, a number of different approaches are used including monitoring parts of the topology or tracing events as the simulation progresses and writing them to files. The actual simulation scale statistics are collected by post-processing the trace and monitor files with awk-scripts, perl-scripts, etc.

4.1.4.1 Flow Monitoring

To observe and analyze the bottleneck links, we created and attached flow monitor to them. A flow monitor is used to collect the statistics of a flow for packet arrivals, departures and drops in either number of bytes or packets. We collect every second the number of bytes departs the bottleneck link in our simulation, which is a good estimation of the number of bytes transmitted within the measurement window (of 1 second). To track the statistics for each flow that traverses the bottleneck link, a classifier is also defined so that it selects the flow based on its flow id. With the statistics of packet departure, it is easy to get the instantaneous throughput for each flow along the bottleneck link. Details will be given in following sections.

4.1.4.2 Tracing

Tracing is the most basic method of collecting simulation information. We could set tracing on individual links or for all links in the network. However, tracing all links will induce considerable performance penalty and large amounts of output data including some useless data. Therefore, selectively tracing links or sources we can perform simulation more efficiently. For example, we chose to trace some TCP parameters such as congestion window, ssthresh, etc.

4.1.4.3 Analysis

Together statistics from various tracing and monitoring files, we use perl and awk scripts in our post-processing process. Xgraph is also a useful graphing tool to present our results. With the help of these tools we have been able to analyze the simulation data that we present below.

4.2 Results and Analysis

We divide our results into three cases for admission control in addition to a service upgrade scenario.

4.2.1 Admission Control Case-1: Scenario 1

Scenario 1 is set as the over-provisioning scenario, whose details of traffic settings are given in Table 4.2. Originally, the mobile node MH0 is registered with the base station HA, while MH1, MH3 and MH7 are in the cell of FA1. The local flows in the cell of FA1 namely flow1 (from CN1 to MH1) is marked as AF1, flow3 (from CN3 to MH3) is marked as AF2 and flow7 (from CN7 to MH7) is marked as BE. WRR weights for AF1, AF2 and BE is 7, 1 and 2 respectively which indicate that they share the bandwidth of the bottleneck link in proportion to 70%, 10% and 20% respectively.

Flow number	Source	Destination	Flow class	Target rate (kbp)	DiffServ marking	WRR weight	Bandwidth*
0	CN0	MH0	HO#	100	AF1	7	70%
1	CN1	MH1	local	100			
3	CN3	MH3	local	100	AF2	1	10%
7	CN7	MH7	local	100	BE	2	20%

* Bandwidth refers to the bandwidth of the bottleneck link in percentages;

# HO stands for handover .

Table 4.2 Traffic Settings for Scenario 1

At simulation time 35s MH0 starts moving from the cell of HA to the cell of FA1, its flow0 is subject to admission control in the new cell. Since flow0 is originally an AF1 flow, for the admission control first checks if it can be assigned to the AF1 class at the bottleneck link. The bandwidth allocated to AF1 is 70% of 300 kbps, that is 210 kbps. As the target rates of flow1 and flow0 are 100 kbps, the available bandwidth for AF1 is:

$$BWA_{AF1} = 210 - 100 = 110 > T_{HO} = 100\,(\text{kbp})$$

This indicates the over provisioning Case-1 of the admission control algorithm where bandwidth available to AF1 is sufficient to accommodate the handover flow. The handover flow into AF1 class and remains marked as AF1 after the handover.

After running the simulation of Scenario1, we get the instantaneous throughput of each flow in Figure 4.3. We can see that after handover the handover flow – flow0 and the local AF1 flow – flow1 share the bandwidth allocation of the AF1 class which is 210 kbps, occupying 70% of the total bandwidth of the bottleneck link and the local AF2 and BE flows are as usual and share the rest of the bandwidth as 10% and 20% of the total bandwidth of the bottleneck link respectively.

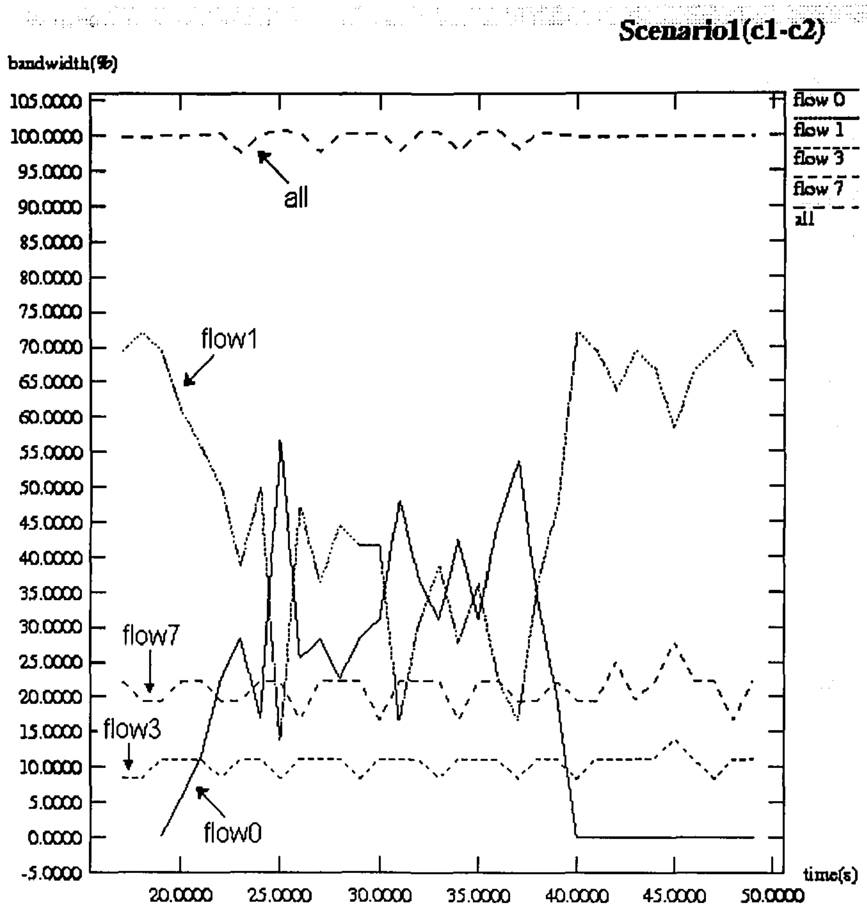


Figure 4.3 Instantaneous Throughput for flows in Scenario 1

We also traced packet numbers and congestion windows for the handover flow at the sender side, which are shown in Figures 4.4 and 4.5.

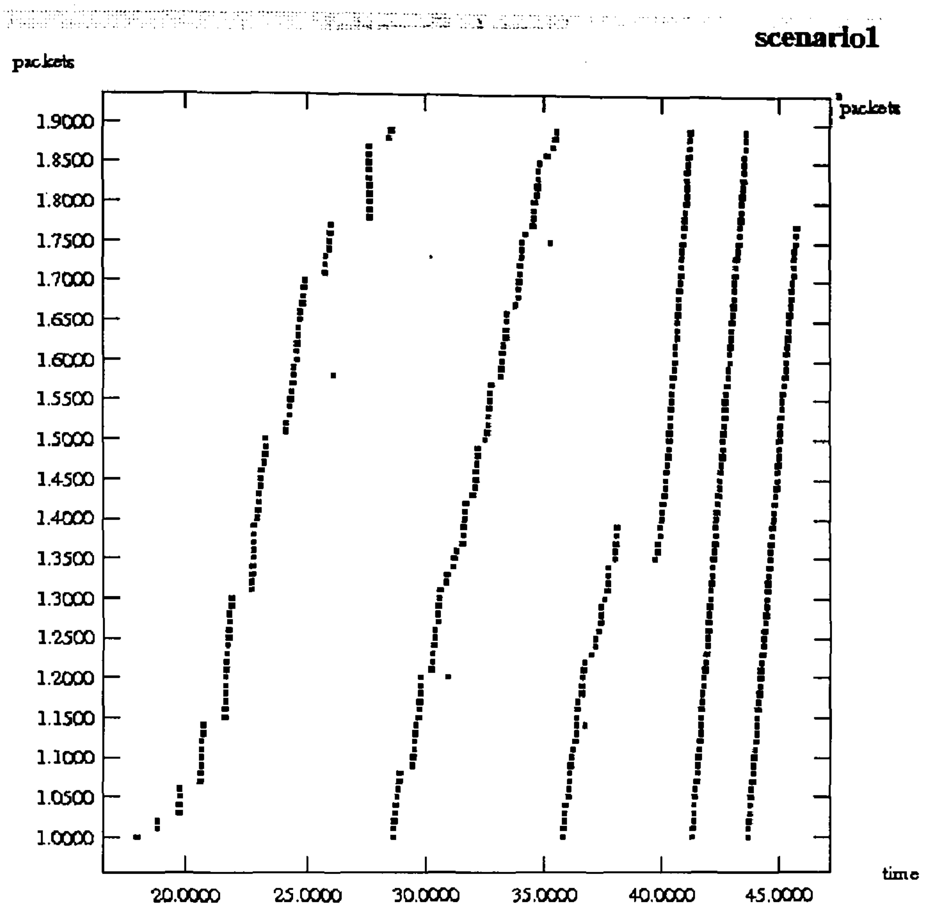


Figure 4.4 Packet Sequence Numbers vs. Time of flow0 in Scenario 1

In Figure4.4, packet numbers are scaled within the range of 1.0 – 1.9. During the period of around 20 – 40s correspondent node CN0 is connected to FA1 and flow0 starts traversing the links along the path to FA1. Flow0 has five retransmissions during the period and they happen at 26.1s, 30.9s, 35.3s, 36.7s and 39.8s respectively.

Figure 4.5 shows that every retransmission causes the congestion window to be reduced by half. The reason for this phenomenon is that we use TCP reno as our source.

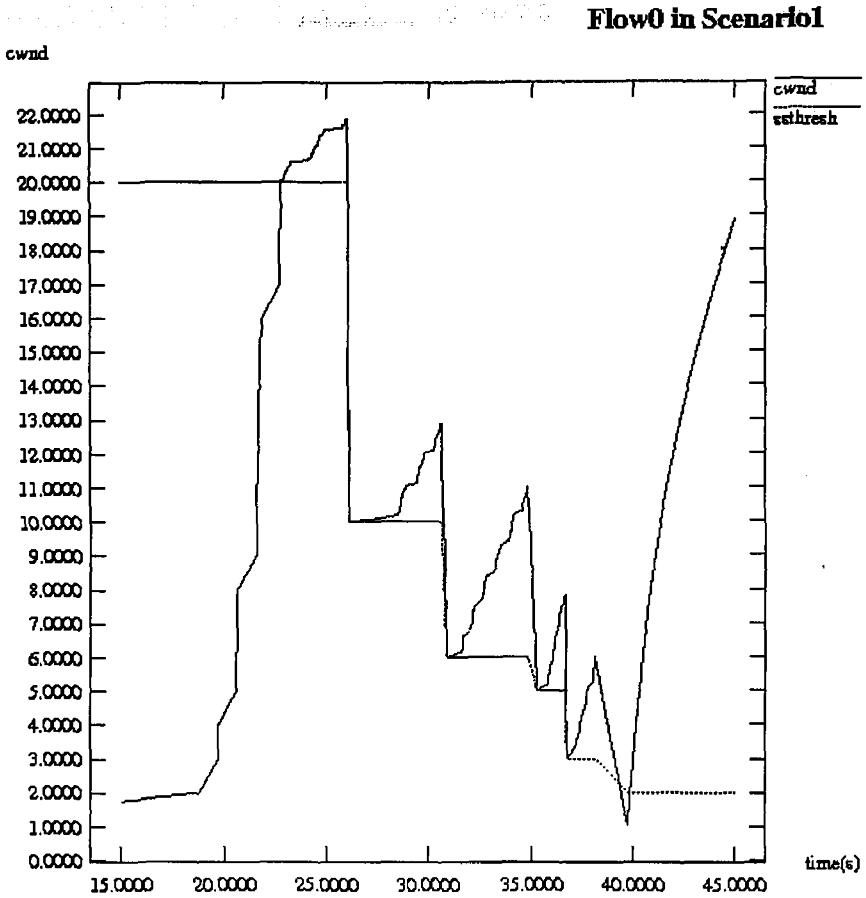


Figure 4.5 Congestion Window vs. Time of Flow0 in Scenario 1

4.2.2 Admission Control Case-2: Scenario 2

Traffic settings for Scenario2 are given in Table 4.3. The mobile node MH0 is originally registered with the base station HA. We have four local flows in the cell of FA1: flow1 (from CN1 to MH1) and flow5 (from CN5 to MH5) are both marked as AF1, while flow3 (from CN3 to MH3) is marked as AF2 and flow7 (from CN7 to MH7) is marked as BE. WRR weights for AF1, AF2 and BE are set to 7, 2 and 1 respectively which indicates that they share the bandwidth of the bottleneck link in proportion to 70%, 20% and 10% respectively.

Flow number	Source	Destination	Flow class	Target rate (kbp)	DiffServ marking	WRR weight	Bandwidth*
1	CN1	MH1	local	100	AF1	7	70%
5	CN5	MH5	local	100			
0	CN0	MH0	HO#	30	AF2	2	20%
3	CN3	MH3	local	30			
7	CN7	MH7	local	100	BE	1	10%

* Bandwidth refers to the bandwidth of the bottleneck link in percentages;

# HO stands for handover .

Table 4.3 Traffic Settings for Scenario 2

In scenario2, bandwidth allocations for AF1 and AF2 are 70% and 20% respectively, which correspond to $300 \times 70\% = 210$ kbps and $300 \times 20\% = 60$ kbps respectively. Then available bandwidths for AF1 and AF2 are:

$$BWA_{AF1} = 210 - (100 + 100) = 10 \text{ (kbp)}$$

$$BWA_{AF2} = 60 - 30 = 30 \text{ (kbp)}$$

Since the target rate for the handover flow in this scenario is T_{HO} = 30 kbps, we have the relation: $T_{HO} > BWA_{AF1}$ and $T_{HO} \leq BWA_{AF2}$ which goes along with Case-2 of our admission control algorithm. Therefore, the algorithm decides to admit the handover flow to AF2 and it is remarked to AF2 when it makes handover to the new cell.

We simulated this scenario and the instantaneous throughput for each flow after the handover is shown in Figure 4.6. The figure shows that local AF1 flows – flow1 and flow5 are not affected by the handover flow, while local AF2 flow – flow3 shares the 20% of the bandwidth of the bottleneck link with the handover flow. Since their target rates are both 30 kbps, they share the bandwidth allocation for AF2 that is 60 kbp. The local BE flow remains the same as before handover.

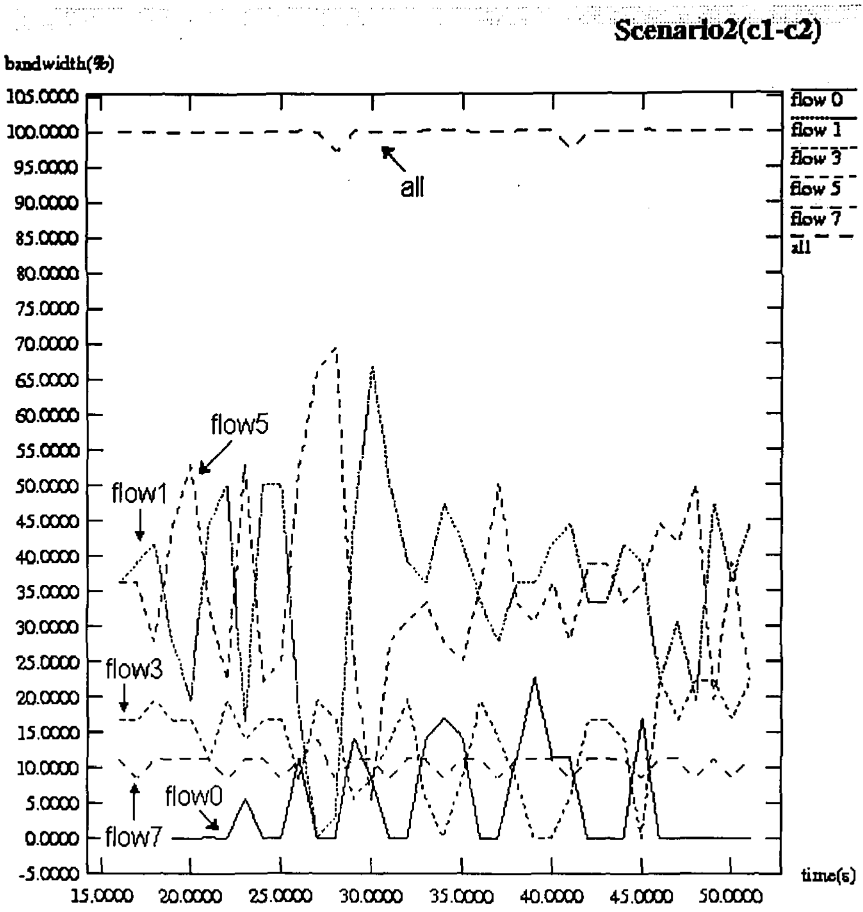


Figure 4.6 Instantaneous Throughput for flows in Scenario 2

We also traced packet numbers and congestion windows for the handover flow at the sender side, which are shown in Figure 4.7 and 4.8.

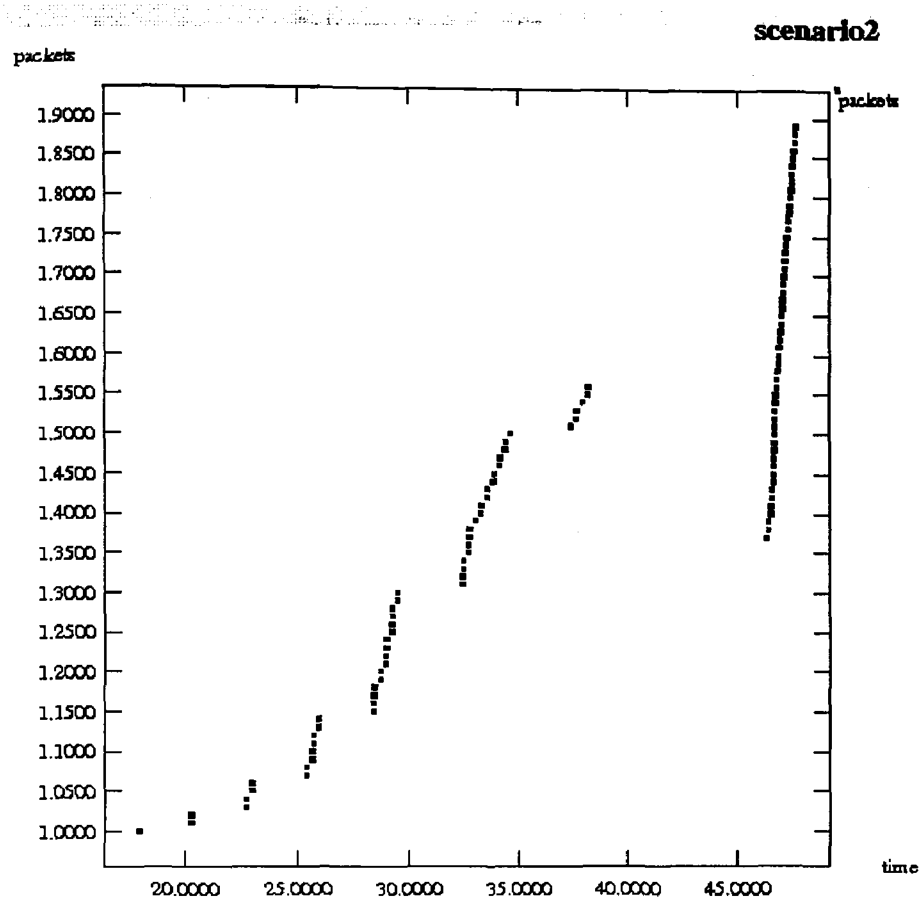


Figure 4.7 Packet Sequence Numbers vs. Time of flow0 in Scenario 2

In Figure 4.7 we don't see any retransmission. In combination with Figure 4.8, we see that during the time when the mobile node MH0 is registered with FA1, flow0 has never reached the congestion avoidance phase since the congestion window remains less than ssthresh for the whole period.

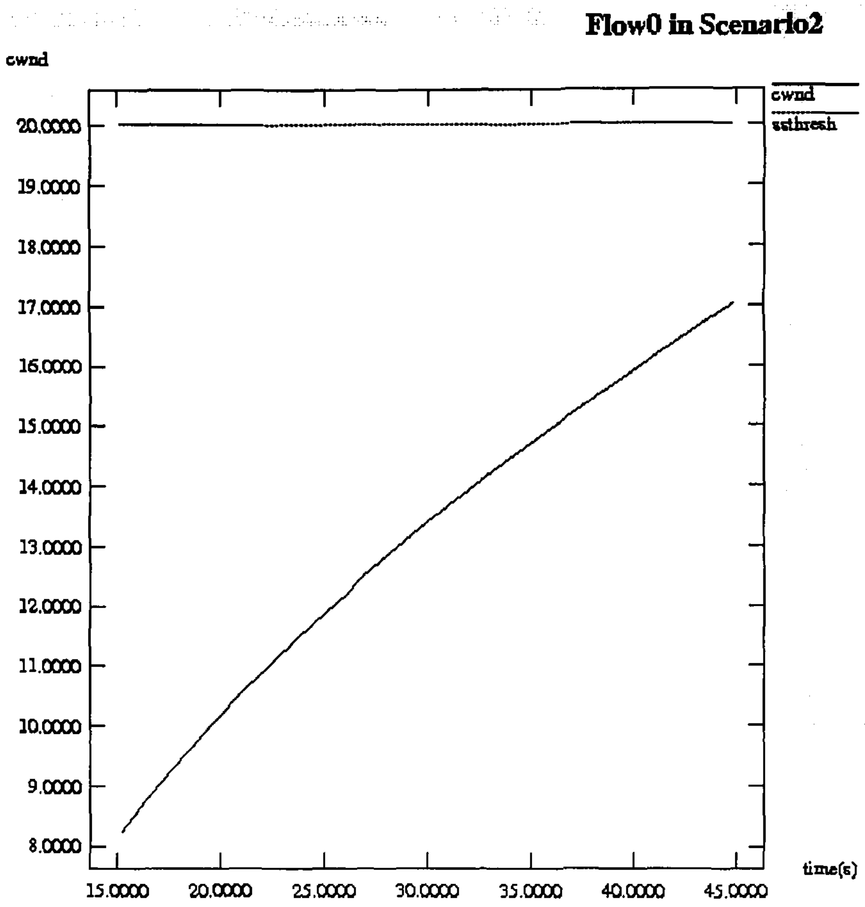


Figure4.8 Congestion Window vs. Time of Flow0 in Scenario 2

4.2.3 Admission Controi Case – 3: Scenario 3

When the available bandwidth of neither AF1 nor AF2 is sufficient to meet the target rate of the handover flow, the proposed admission control algorithm applies penalty scheme of Case-3 to determine the assignment of the handover flow to the appropriate class. We simulated this case in the following scenario.

4.2.3.1 Scenario 3a

Traffic settings for this scenario are given in Table 4.4. Before the handover flow comes to the new cell available bandwidth for AF1 and AF2 are:

$$BWA_{AF1} = 210 - (100 + 100) = 10 \text{ (kbps)}$$

$$BWA_{AF2} = 0$$

So neither of them is enough to meet the target rate of the handover flow and this scenario falls into the situation of case – C of our admission control algorithm.

Flow number	Source	Destination	Flow class	Target rate (kbp)	DiffServ marking	WRR weight	Bandwidth*
0	CN0	MH0	HO#	100	AF1	7	70%
1	CN1	MH1	local	100			
5	CN5	MH5	local	100			
3	CN3	MH3	local	100	AF2	1	10%
7	CN7	MH7	local	100	BE	2	20%

* Bandwidth refers to the bandwidth of the bottleneck link in percentages;

# HO stands for handover .

Table 4.4 Traffic Settings for Scenario 3a

Let's calculate the penalty for AF1 and AF2 if the handover flow is assigned to the respective class.

$$\text{Since } \alpha_{AF1} = \frac{BW_{AF1}}{\sum_{i=1}^{p} T_i + T_{HO}} = \frac{300 \times 70\%}{100+100+100} = 0.7$$

$$P_{AF1} = (p+1)\cdot(1-\alpha_{AF1}) = 3 \times (1-0.7) = 0.9$$

$$\alpha_{AF2} = \frac{BW_{AF2}}{\sum_{j=1}^{q} T_j + T_{HO}} = \frac{300 \times 10\%}{100+100} = 0.15$$

$$P_{AF2} = (q+1)\cdot(1-\alpha_{AF2}) = 2 \times (1-0.15) = 1.7$$

if β=1, we have the relation $P_{AF1} < \beta \cdot P_{AF2}$, so flow0 should be marked as AF1 according to our admission control algorithm. We simulate this scenario and the instantaneous throughput is given in Figure 4.9.

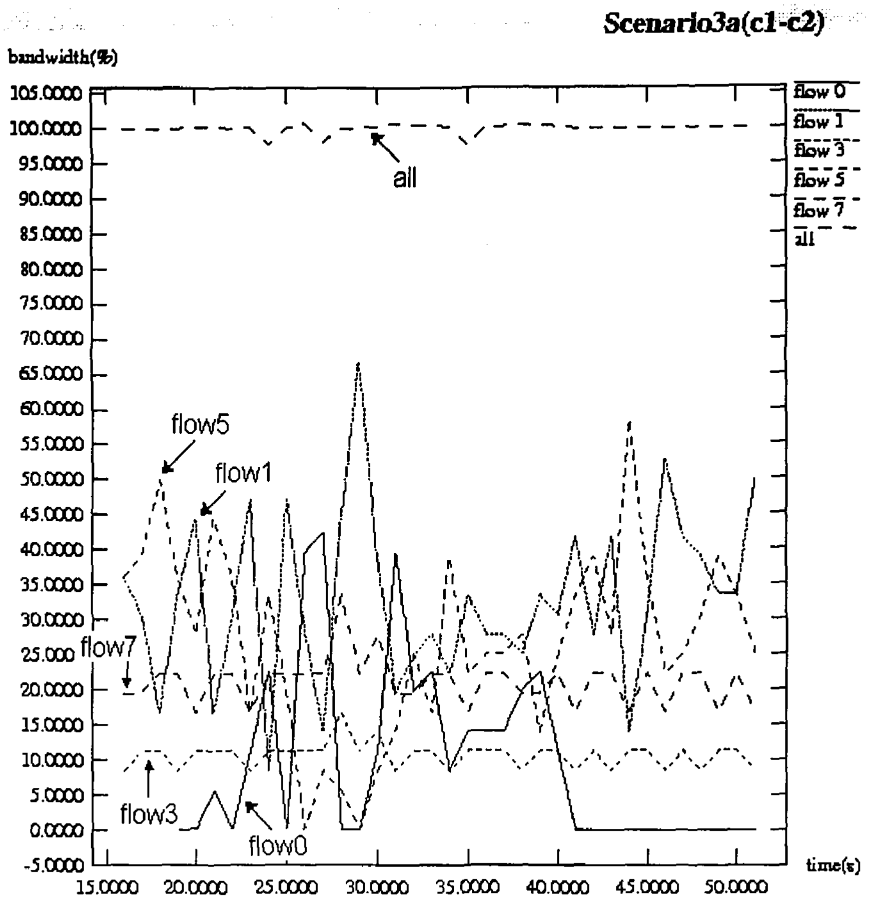


Figure 4.9 Instantaneous Throughput for flows in Scenario 3a

Figure 4.9 shows that when the handover flow, flow0, is marked as AF1 it competes for bandwidth with local AF1 flows, flowl and flow5, and cause their bandwidth to drop. Local AF2 flow, flow3, and local BE flow, flow7, are not affected by the handover. This scenario presents the situation where AF1 and AF2 flows get equal protection. However, if we want to give more protection to local AF1 flows, we can increase the value of β. We discuss this case in the following section.

4.2.3.2 Scenario 3b

In scenario 3b, traffic settings are exactly the same as that of scenario 3a. Thus the penalties are calculated to the same values as for Scenario 3a, which are given below:

$P_{AF1} = 0.9$ and $P_{AF2} = 1.7$

if we choose $\beta = 0.5$,

then $\beta \cdot P_{AF2} = 0.5 \times 1.7 = 0.85 < P_{AF1}$

so the handover flow - flow0 should be marked as AF2 according to our admission control algorithm. We simulated this scenario and result is given in Figure 4.10.

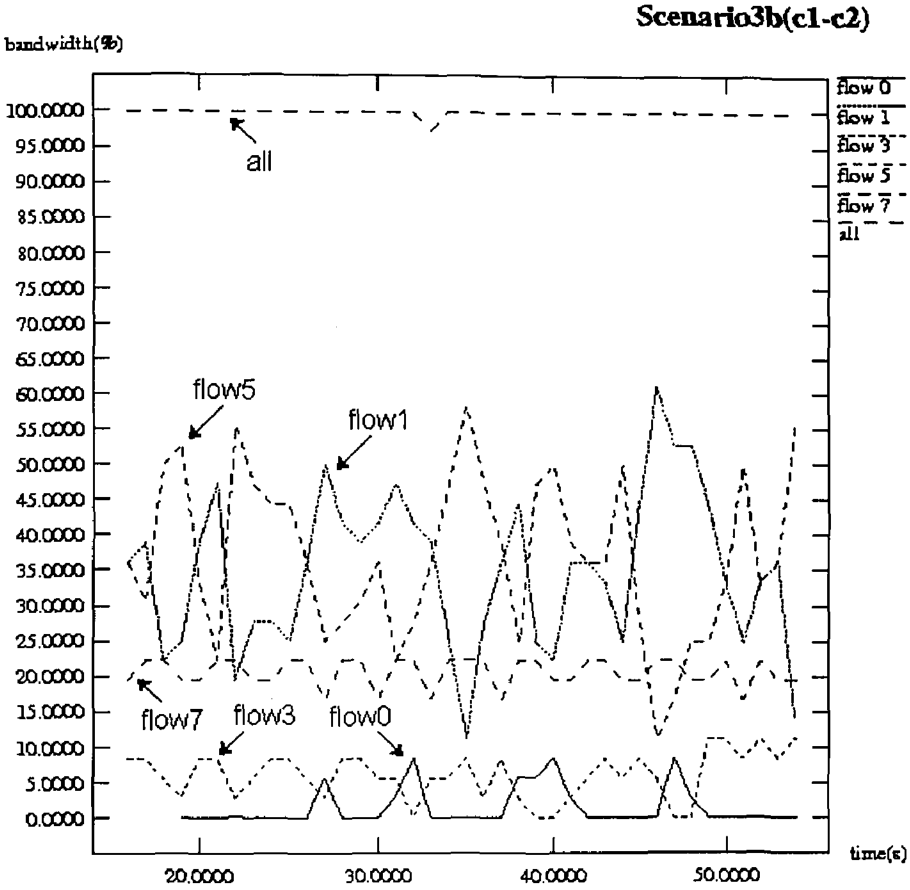


Figure 4.10 Instantaneous Throughput for flows in Scenario 3b

From Figure 4.10, we see that local AF1 flow, flow1 and flow5, are well protected and continue receiving the same bandwidth as what they have had before the handover. Local AF2 flow, flow3, shares the bandwidth with the handover flow. Since bandwidth allocation for AF2 is 10% and the two flows have the same target rate of 100 kbps, each gets 5% of the bandwidth of the bottleneck link after the handover, which is equivalent to $300 \times 10\% = 30$ kbps and much less than their target rates. Local BE flow, flow7, remains unaffected.

4.2.3.3 Impact of Weight Adjustment on Factor β

In senario3b, we want to give more protection to local AF1 flows and assigned the handover flow to AF2, which consequently results in AF2 flows missing their target rates. In this situation we can employ the weight adjustment algorithm, to steal some bandwidth from the lower service class level – BE.

Assume that we steal 10% on the bandwidth of the bottleneck link from BE and add that to AF2, thus the bandwidth allocation for AF2 will be increased to:

$$BW_{AF2} = 300 \times 20\% = 60 \text{ (kbp)}$$

and $$\alpha_{AF2} = \frac{BW_{AF2}}{\sum_{j=1}^{q} T_j + T_{HO}} = \frac{60}{100+100} = 0.3$$

then $$P_{AF2} = (q+1)\cdot(1-\alpha_{AF2}) = 2 \times (1-0.3) = 1.4$$

Since penalty for the AF2 flows if the handover flow is marked as AF2 decreases as a result of increase in bandwidth allocation through weight adjustment, β should be raised proportionally in order to mark the handover flow as AF2 and give more protection to local AF1 flows. Consider the situation in scenario 3b, when $\beta \geq \frac{P_{AF1}}{P_{AF2}} = \frac{0.9}{1.7} \approx 0.53$, the handover flow is marked as AF1, otherwise it is marked as AF2. Now when weight is added to AF2, we have the new decision bound as when $\beta \geq \frac{P_{AF1}}{P_{AF2}} = \frac{0.9}{1.4} \approx 0.64$, the handover flow is marked as AF1, otherwise it is marked as AF2.

Therefore, when AF2 gets more weight, the decision bound (β) for marking of the handover flow also increases to provide the same level of protection to AF1 flows as it was receiving before the weight adjustment. Thus, the bandwidth management algorithm needs to be modified to take into consideration the situation that when weight adjustment algorithm is triggered, the β factor should be adjusted accordingly other wise the protection level of AF1 flows will be changed. We leave for further research to explore the space of β adjustments.

4.2.4 Service Upgrading: Scenario 4

To understand the dynamic of service upgrade algorithm, we still investigate the upgrade situation on the bottleneck link between nodes c1 and c2. Traffic settings for this scenario are given in Table 4.5.

Flow number	Source	Destination	Flow class	Target rate	DiffServ marking		WRR weights		Bandwidth*	
					Old	New	Old	New	Old	New
5	CN5	MH5	HO#	100	AF1	-	7	-	70%	-
1	CN1	MH1	local	100		AF1		7		70%
0	CN0	MH0	local	100	AF2		2		20%	
3	CN3	MH3	local	100		AF2		2		20%
7	CN7	MH7	local	100	BE		1		10%	

* Bandwidth refers to the bandwidth of the bottleneck link in percentages;

# HO stands for handover .

- Old means the value before the handover happens;
- New means the value after the handover happens.

Table 4.5 Traffic Settings for Scenario 4 of Service Upgrading

This scenario presents the dynamic situation in the cell of FA1. Originally, there are five mobile nodes in the coverage area of FA1, which are MH0, MH1, MH3, MH5 and MH7. Flow1 (from CN1 to MH1) and flow5 (from CN5 to MH5) are local AF1 flows, flow0 (from CN0 to MH0) and flow3 (from CN3 to MH3) are local AF2 flows, and flow7 (from CN7 to MH7) is a local BE flow. The provisioning scheme sets WRR weights to 7, 1, 2 (corresponding to 70%, 10% and 20% of the bottleneck link bandwidth) for AF1, AF2 and BE respectively. We assume at simulation time 25s mobile node MH5 moves from the cell of FA1 to the cell of FA2. We also assume that flow0 (from CN0 to MH0) has higher service upgrade priority than flow3 (from CN3 to MH3). Therefore, according to our service upgrade algorithm, flow0 is upgraded and remarked as AF1 after the handover of MH5.

Simulation result of this scenario is shown in Figure 4.11. We can see that as a result of handover flow5 releases its bandwidth on the link, and flow0 is upgraded to AF1 and share the AF1 bandwidth

(70% of the bottleneck link bandwidth) with flow3, which is another local AF1 flow on the same link. Since flow0 is upgraded to AF1, there is only one AF2 flow, flow3, along the link after the handover of MH5, which gets the whole 20% allocation for AF2 that it was sharing before handover with flow0. Local BE flow remains unaffected in this scenario.

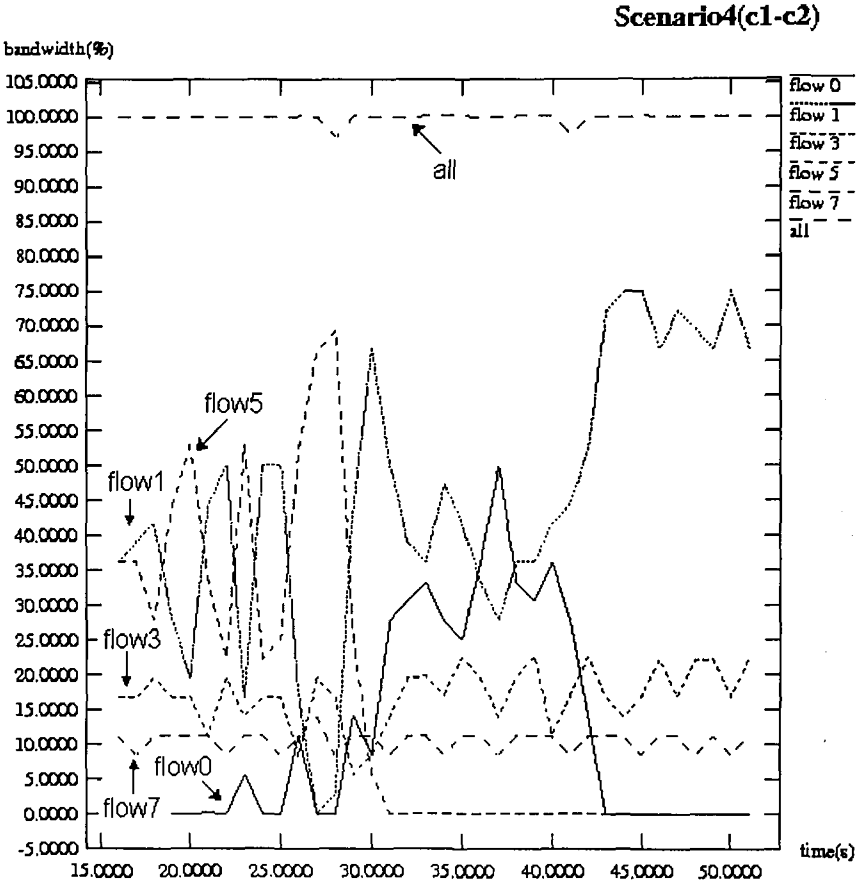


Figure 4.11 Result of the Service-upgrading Scenario

4.3 Summary

In this chapter, we setup a simulation model of a mobile wireless access network using the network simulator ns-2. The simulation topology consists of correspondent nodes, core router nodes, mobile nodes and three base stations acting as Radio Edge Routers. We used FTP to generate long TCP-connections. We employed MRED buffer management scheme to implement differential treatment for AF1, AF2 and BE, and selected the MRED parameters such that AF1 gets lower drop precedence than AF2. WRR scheduler is used to schedule different queues and its weights for AF1, AF2 and BE are selected to create different provisioning scenarios.

Based on the mobile wireless access model, we first simulated three scenarios for the admission control algorithm for handover flows including one over-provisioning scenario and two under-provisioning scenarios. Then the forth scenario is given for the service-upgrading algorithm. In the over-provisioning scenario, when the available bandwidth for AF1 service is no less than the target rate of the handover flow, the handover flow can still be marked as AF1. Results show that both the handover flow and existing AF1 flows can meet their target rate in this situation and they also share the excess bandwidth for AF1 flows.

In the under-provisioned scenarios, we first studied the situation when the target rate of the handover flow is bigger than the available bandwidth for AF1 service class but less than that for AF2 service class. Then the handover flow is remarked as AF2. Results show that existing AF1 flows are well protected and receive the same bandwidth as before handover, while the handover flow can also meet their target rate at the handover time but may suffer worst treatment in congestion due to the lower service level marking.

The situation when the target rate of the handover flow is bigger than both the available bandwidth for AF1 and AF2 service class is studied in Scenario 3. To decide whether to remark the handover flow or not, we calculated the penalty for the situation when the handover flow is marked as AF1 and when the handover flow is marked as AF2. We also introduce a factor β to the comparison process. Results in Scenario 3a shows that when β=1, we get equal compare for penalty of the respective situations when the handover flow is marked as AF1 and when the handover flow is marked as AF2. On the other hand, when β is less than 1, results in Scenario 3B show that more protection is given to existing AF1 flows.

In Scenario 4, we simulated the situation when an existing AF1 flow ends and releases bandwidth to AF1 flows. Then AF2 flows are considered to be upgraded to AF1. Results show that the flow with the highest upgrade priority is upgraded first and release bandwidth to AF2 flows.

Chapter 5
Conclusion and Future Work

5.1 Concluding Remarks

Providing feasible dynamic bandwidth management for TCP flows in a DiffServ-capable mobile wireless access network is a challenging problem. This thesis investigates the problem in some level of detail. In this thesis we focus on the issue of admission control for handover flows. We developed an admission control and a bandwidth management algorithm. Our algorithms contribute to a better understanding of the problem, and provide some possible solutions. We validated the feasibility of our algorithms through a set of simulations.

In the thesis, we consider a mobile wireless access network where DiffServ is deployed as the QoS solution and Mobile IP as employed as the handover protocol. We also consider two service classes: AF1 and BE, where AF1 is used to provide qualitative service assurance and BE to provide the best effort service. We first presented a study on the impact of handover on DiffServ flows by simulating a simple scenario. Results show that handover contributes significantly to the service level degradation to the local AF1 flows in a cell. Further, separating the handover and local flows can protect the local AF1 flows. This provides motivation for our QoS scheme and admission control algorithms. In this thesis we made four contributions to the QoS management in a DiffServ- enabled mobile wireless access network. First, we propose a QoS scheme that protects local flows from losing bandwidth to handover flows by separating the two flows into different service classes. The separation allows graceful degradation of service level of handover flows. Second, we propose an admission control algorithm for handover flows. Third, we propose a service upgrade algorithm to upgrade the service level of handover flows. Fourth, we propose a dynamic bandwidth-provisioning algorithm that allows dynamic adjustment of bandwidth allocations to AF2 and BE flows by adjusting their respective weights configured at the scheduler.

We propose AF2 class to be used as a transient class for handover flows. When a handover flow is expected to cause significant degradation to the bandwidth of local AF1 flows then it is assigned to AF2 class. The AF2 flows are expected to receive better than best effort service. Our admission control algorithm contributes mechanisms to protect the service level of local flows, at the same time admit handover flows and gracefully downgrade their service when bandwidth provisioning on the bottleneck link to the new cell is not sufficient. A mechanism is also provided to enable the admission control algorithm to make necessary tradeoff between maintaining the service level of existing flows and admitting as much incoming handover flows as possible.

Service level upgrading algorithm provides an important mechanism for dynamic service level adjustment. The load of each service class, AF1 and AF2, are monitored periodically. Whenever load of AF1 decreases up to a point where it can accommodate some AF2 flows, then the AF2 flows are upgraded to AF1. The selection of handover flows for upgrade is controlled by their upgrade priority. This algorithm tries to mitigate the effect of handover on the service of handover flows.

When a large number of flows handover to AF2 service class or some fat flows move to the new cell, then the risk for AF2 flows to receive lower than BE bandwidth grows. We proposed a dynamic provisioning scheme whereby new weights are computed for AF2 and BE to move bandwidth from BE to AF2. We also provide a mechanism of returning the bandwidth to BE when the AF2 utilization goes low.

We evaluated the feasibility of our QoS scheme and algorithm by simulating different handover situations. Our simulation shows that the weight adjustment has an impact on the β factor we introduced to control algorithm is capable of protecting local flows under variety of handover situations.

5.2 Future Work

QoS Provisioning in DiffServ-capable mobile IP networks is a promising area of research and there are many issues that warrant further investigation.

In this thesis we focus on the bandwidth assurance to TCP flows. In fact in our background analysis, we introduced five factors that in some way or another affect the throughput rates achieved by TCP flows of end users. These factors include Round Trip time, number of micro flows, size of target rate, packet size and interaction with non-responsible (UDP) flows. Besides target rate that we have studied in this thesis, other factors could be included in future studies would provide better understanding on the bandwidth assurance issues and may improve our algorithms.

Concerning different factors that affect bandwidth assurance issues for TCP flows in DiffServ networks; another related research area is on DiffServ marking schemes. Besides the simple Token

Bucket policy we used in this thesis, some more complicated intelligent marking scheme could be considered in future studies.

Related to our research, there are many interesting problems to be explored. For example, in the core network how to discover the bottleneck link. Will bottleneck link change by the frequent moving of mobile nodes? In the access network, the implementation of dynamic bandwidth provisioning scheme requires provisioning protocol. Moreover, the effect of handover on additional QoS parameters such as delay, packet loss rate, etc is also an interesting topic for future studies.

References

[1] C. Perkins, "IP Mobility Support", RFC 2002, Internet Engineering Task Force, October 1996.

[2] R. Koodli, Nokia Research center, "Fast Handovers for Mobile Ipv6", Internet Draft, Oct 2004.

[3] Bos, L. and Leroy, S. "Toward an ALL-IP-based UMTS System Architecture". IEEE Network, Vol. 15(1), pp.36-45, 2001.

[4] S. Blake, et al, "An Architecture for Differentiated Services", RFC 2475, 1998.

[5] The UMTS Forum, "3G/UMTS Towards Mobile Broadband and Personal Internet", White Paper, Feb, 2005.

[6] L. McLoughlin, "cdmaOne ™& cdma2000 White Paper", Dec 2000.

[7] C. Perkins, "IP Mobility Support for Ipv4", RFC 3344, August 2002.

[8] D. John, C. Perkins, J. Arkko, "Mobility Support in Ipv6", RFC 3775, June 2004.

[9] V. Jacobson, et al, "An Expedited Forwarding PHB", RFC 2598, 1999.

[10] J. Heinanen, et al, "Assured Forwarding PHB Group", RFC 2597, 1999.

[11] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 1(4), pp.397-413, August 1993.

[12] M. Goyal, et al, "Effect of Number of Drop Precedences in Assured Forwarding", GLOBECOM 99, Rio De Janeiro, December 99. Conference Record / IEEE Global Telecommunications Conference, V 1 (B), pp. 188–193, 1999.

[13] D. Clark, et al, "Explicit Allocation of Best-Effort Packet Delivery Service", IEEE/ACM Transactions on Networking, Vol. 6(4), pp. 362-373, August 1998.

[14] Technical Specification from Cisco, "Distributed weighted Random Early Detection", *http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.pdf.*

[15] A. Silberschatz and P. galvin, Operating system Concepts. Reading, MA, USA: Addison-Wesley, 5th ed., 1998

[16] H. M. Chaskar and U. Madhow, "Fair Scheduling with Tunable Latency: A Round Robin Approach", IEEE/ACM Transactions on Networking, Vol. 11(4) pp. 592-601, August 2003.

[17] John B. Nagle, "On packet Switches with Infinite Storage", IEEE Transactions On Communication, Vol. 35(4), pp.435-438, Apr 1987.

[18] Technical Specification from Cisco, "Congestion Management Overview", *http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt2/qcdconmg.pdf.*

[19] R. Liao, et al, "Dynamic Core Provisioning for Quantitative Differentiated Services", IEEE/ACM Transactions on Networking, Vol. 12(3), pp.429-442, June 2004.

[20] S. Tartarelli, A. Banchs, "Random Early Marking: Improving TCP Performance in DiffServ Assured Forwarding", ICC 2002 –IEEE International Conference on Communications, Vol. 25, No. 1, pp. 970-975, April 2002.

[21] J. Heinanen, et al, "A Single Rate Three Color Marker", RFC 2697, 1999.

[22] J. Heinanen, et al, "A Two Rate Three Color Marker", RFC 2698, 1999.

[23] W. Fang, N. Seddigh, "Time Sliding Window Three Color Marker", RFC 2859, 2000.

[24] 3G TS 23.107 v5.3.0 (2002-01), "QoS Concept and Architecture", http://www.3gpp.org.

[25] R. Feng, J. Song, "Some QoS Issues in 3G Wireless Networks", TENCON'02, Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, control and Power Engineering. Vol 2, 28-31 pp.724-727, Oct 2002.

[26] Y. Cheng and W. Zhuang, "DiffServ Resource Allocation for Fast Handoff in Wireless Mobile Internet", IEEE Communication Magazine, vol. 40(5), pp.130-136, May 2002.

[27] N. Seddigh, B. Nandy, P. Pieda, "Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network", Proceedings of Global Internet symposium, Globlecom 99, Rio De Janeiro, December 1999.
http://www.sce.carleton.ca/~nseddigh/publications/globe9806.pdf

[28] B. Nandy, N. Seddigh, P. Pieda, "DiffServ's Assured Forwarding PHB: What Assurance does the Customer Have?" in Proceedings of Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)'99, July 1999.
http://www.nossdav.org/1999/papers/82-1232701015.pdf

[29] M.Goyal, et al, "Performance Analysis of Assured Forwarding", Internet Draft, Feb 2000

[30] M. Mathis, et al, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communication Review, vol. 27(3), pp. 67-82, July 1997.

[31] R. Morris, "TCP Behavior with Many Flows", IEEE International Conference on Network Protocols, Atlanta, October 1997, pp. 205-211.

[32] UCB/LBNL/VINT. Network Simulator – ns, DiffServ Module. [On –line]. Available: http://www.isi.edu/nsnam/ns.