Supporting Integrated Services

In a Cellular Packet Switched Network

Min Seok Kang

A dissertation submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

of the

University of London

Department of Computer Science
University College London

1999
Abstract

The third-generation cellular network should be capable of accommodating the integration of data sources with various QoS requirements, along with voice service. There has been no lack of research effort on the integrated services for fixed networks, but these methods cannot be directly applied to a cellular packet network since it has different characteristics such as the handovers and varying available bandwidth due to mobile hosts. This thesis presents an integrated service architecture for the cellular packet switched network and deals with its design issues. It provides uplink and downlink connections with diverse QoS keeping an identical link service for both link directions whilst maintaining fairness among connections regardless of handovers.

The service architecture we propose in this thesis can accommodate multiple scheduling disciplines to serve various traffic according to their traffic characteristics. For the service architecture, we developed two scheduling disciplines, PEDQ (Packet Endurable Delay Queuing) and PCAWFQ (Packet-by-Packet Combined Absolute and Weighted Fair Queuing). The PEDQ is designed for delay-intolerant traffic and schedules packets according to the remaining lifetime of a packet to minimise packet drops. The PCAWFQ is a general packet scheduling discipline designed for scheduling delay-tolerant traffic in a variable-rate link and can simultaneously schedule three different kinds of flow classes according to their class service policies. For handling uplink traffic, we propose a multiple access protocol called FQMA (Fair Queuing Multiple Access) which is a slot-based protocol to manage delay-intolerant and delay-tolerant traffics simultaneously using the proposed two scheduling disciplines, whilst performing a decoupling between both traffic to prevent delay-tolerant traffic from degrading the service performance for delay-intolerant traffic. We also propose the proxy packet-scheduling concept to serve distributed packets in mobile hosts as if they are queued in a single queue. It enables a scheduling scheme designed for downlink service to serve uplink traffic.

We also investigate and analyse the proposed system in the handover environment. We identified that unfairness occurs in some types of schedulers during handovers due to packet forwarding. We also identified that misordered packet delivery might occur during handovers if a handover protocol does not take account of the packet level service. We propose a compensation scheme and a packet forwarding-aware handover protocol to remedy both problems respectively.

Finally, we validate our proposed service model using simulations with the OPNET simulator. We conclude that the proposed service model is viable and can provide guaranteed integrated services in a cellular packet switched network.
Acknowledgements

I would like to express my sincerest gratitude and appreciation for my supervisor, Prof. Steve Wilbur, for his guidance, support and patience. The successful completion of this thesis is in a large part due to his encouragement, probing criticisms and judicious advice. I feel fortunate that I had the chance to work with such an excellent supervisor. I would also like to thank Dr. Soren-Aksel Sorensen for a fruitful collaboration. He supplied simulation ideas and sharp critical thinking at various stages in obtaining some of the result reported herein. Prof. Jon Crowcroft provided both encouragement and analytical expertise relating to my research. He also took the time to carefully read my several papers relating this thesis. Bhrat Patel was always available as a helpful and encouraging friend throughout my years of research. He also took time from his own research to read portions of my thesis. I am also very grateful to Paul White for reading my thesis and making useful suggestions. I have benefited greatly from the social and professional environment of my laboratory. I will forever remember as fond memories my daily contact with Richard Akester, Mazlisa Othman, Joc Chappell, Ian Brown, Ashok Gupta, Jungwon Kim, Hugh Mallinson and Ben Southall.

I wish to thank also to Prof. David Hutchison and Prof. Laurie Cuthbert for reading my thesis and recommend me as a PhD. Their critical comments helped me greatly to improve the final product.

I am eternally grateful to my deceased father during my studying period and my mother for their priceless gift of love throughout my whole life. My strength and determination is but a product of their unflagging efforts. There is no doubt that their love made it possible for me to fulfil my research. My debt to my loving wife for her selfless support and dedication is profound and incalculable. Through the years of my graduate study, she supported me patiently and unconditionally. I would like to thank or at least apologise to my son who has not been properly looked after by me due to my research. I am deeply grateful to all of them.

The research reported herein was supported by British Foreign and Commonwealth Office and LG Corporate Institute of Technology. I am very grateful for their support. I especially thank Dr. Kim Chang-Su, Dr. Lee Hee-Yeon, Dr. Lee Heun-Joo and Mr. Kang Shin-Wok who gave an opportunity to pursue my research.
Declaration

Except where otherwise stated in the text, this dissertation is the result of my own work and is not the outcome of work done in collaboration. This dissertation is not substantially the same as any I have submitted for a degree or diploma or any other qualification at any other University. No part of this dissertation has already been, or is being currently submitted for any such degree, diploma or other qualification.

Publications

Materials presented in several parts of this dissertation also appear in the following publications.


Contents

ABSTRACT .................................................................................................................................2

ACKNOWLEDGEMENTS ................................................................................................................3

DECLARATION ..............................................................................................................................4

LIST OF FIGURES .......................................................................................................................10

LIST OF TABLES ..........................................................................................................................13

GLOSSARY OF TERMS ...............................................................................................................14

CHAPTER 1 - INTRODUCTION ...................................................................................................17

1.1 INTEGRATED SERVICES IN A CELLULAR DATA NETWORK .............................................17
1.2 THE BARRIERS IN A CELLULAR PACKET SWITCHED NETWORK .....................................18
1.3 CONTRIBUTIONS ..................................................................................................................19
  1.3.1 Scheduling Algorithm ....................................................................................................19
  1.3.2 Multiple access protocol ...............................................................................................20
  1.3.3 Handover .........................................................................................................................21
1.4 DISSERTATION OUTLINE ...................................................................................................22

CHAPTER 2 - BACKGROUND ....................................................................................................23

2.1 CELLULAR DATA NETWORKS ............................................................................................23
  2.1.1 Macro-cellular data network ............................................................................................23
    2.1.1.1 HSCSD (High Speed Circuit Switched Data) - GSM .............................................24
    2.1.1.2 GPRS (Global Packet Radio Service) .....................................................................24
    2.1.1.3 CDPD (Cellular Digital Packet Data) .....................................................................25
  2.1.2 Micro/Pico-cellular data network ....................................................................................25
    2.1.2.1 Wireless LAN .........................................................................................................26
    2.1.2.2 PHS (Personal Handyphone System) ....................................................................27
  2.1.3 Hierarchical cellular network ..........................................................................................28
    2.1.3.1 UMTS/IMT2000 .....................................................................................................28
    2.1.3.2 BARWAN (Bay Area Research Wireless Access Network) ..................................29
  2.1.4 Summary .........................................................................................................................29
2.2 QoS Provision in an Integrated Services Packet Network ................................................. 30
  2.2.1 Two procedures for the first approach ................................................................................. 31
  2.2.2 Resource reservation protocol - RSVP ................................................................................ 31
  2.2.3 Resource distribution and Packet scheduling scheme ...................................................... 32
  2.2.4 Adaptive applications ................................................................................................................ 33
    2.2.4.1 Playback application................................................................................................................. 33
    2.2.4.2 Interactive application ............................................................................................................ 33
  2.3 QoS Provision in a Cellular Packet Switched Network.......................................................... 34
    2.3.1 Bottlenecks and scheduling discipline ................................................................................. 35
    2.3.2 Uplink traffic service and multiple access protocol .......................................................... 35
    2.3.3 Mobility related issues ............................................................................................................. 36
      2.3.3.1 Efficient handover protocol for seamless delivery ...................................................... 36
      2.3.3.2 Reservation protocol supporting mobility ........................................................................ 37
      2.3.3.3 Addressing ......................................................................................................................... 37
      2.3.3.4 Improving End-to-End performance .............................................................................. 38
      2.3.3.5 Call control issues .............................................................................................................. 38
  2.4 Research Motivation and Scope......................................................................................... 39

CHAPTER 3 - RELATED WORK .................................................................................................. 43
  3.1 Scheduling Disciplines ........................................................................................................ 43
    3.1.1 Fair Queuing ............................................................................................................................... 43
    3.1.2 GPS (Generalised Processor Sharing) .................................................................................. 44
    3.1.3 PGPS (Packetised GPS) ........................................................................................................... 45
    3.1.4 WF²Q (Worst-case Fair Weighted Fair Queuing) ............................................................ 46
    3.1.5 SCFQ (Self Clocked Fair Queuing) ...................................................................................... 46
    3.1.6 SFQ (Start-time Fair Queuing) .............................................................................................. 47
    3.1.7 Virtual Clock ............................................................................................................................... 48
    3.1.8 CBQ (Class Based Queuing) ................................................................................................. 49
    3.1.9 Comparisons of CBQ and Rate-based Scheduling ............................................................ 51
    3.1.10 Summary .................................................................................................................................... 51
  3.2 Multiple Access Protocols ................................................................................................ 51
    3.2.1 PRMA (Packet Reservation Multiple Access) .................................................................... 52
    3.2.2 Improved versions of PRMA ............................................................................................... 52
    3.2.3 I-ISMA (Idle Signal Multiple Access for Integrated services) ......................................... 53
List of Figures

2.1 Micro-cellular Network .................................................................................................................. 25
2.2 Pico-cellular network ...................................................................................................................... 26
3.1 Graphical comparison between CBQ and Rate-based scheduling .............................................. 50
4.1 Five groups of Delay tolerant-applications ................................................................................... 64
4.2 The hierarchy of applications ......................................................................................................... 64
4.3 An example of a wireless link share tree ........................................................................................ 67
4.4 The proposed traffic scheduling system for a base station .......................................................... 69
4.5 Common channel usage ............................................................................................................... 72
4.6 Slot allocation flow chart for uplink/downlink sharing scheme ................................................. 73
5.1 The guaranteed bandwidth effect of rate-based scheduling disciplines over a fixed and variable-rate link ......................................................................................................................... 77
5.2 The guaranteed bandwidth effect of the proposed scheduling discipline (CAWFQ) over a fixed and variable bandwidth link ........................................................................................................... 78
5.3 Left: The virtual time and virtual finish time of served packet in WFQ, Right: the difference of the two times ......................................................................................................................... 79
5.4 Left: The virtual time and virtual finish time of served packet in V-WFQ, Right: the difference of the two times ......................................................................................................................... 80
5.5 The packet service order Top: WFQ, Bottom: V-WFQ .................................................................. 80
5.6 An example of a backlogged and idle period ................................................................................. 82
5.7 CAWFQ service and service sequential orders ............................................................................ 85
5.8 Relationship between the packet departure times and the reference virtual times ...................... 90
5.9 The network configuration for the simulation .............................................................................. 98
5.10 Simulation results for a semi-deficit period .................................................................................. 100
5.11 Simulation results in a deficit period .......................................................................................... 100
5.12 Simulation results in a surplus period ....................................................................................... 101
5.13 Simulation results for a backlogged period .............................................................................. 101
5.14 Comparison of end-to-end delay between PCAWFQ and V-WFQ under a semi-deficit period ........................................................................................................................................... 102
6.1 Examples of uplink slot assignments .......................................................................................... 106
6.2 Uplink slot formats ...................................................................................................................... 107
6.3 Comparison between MET (Maximum Endurable Time) and PDT (Packet Drop Time) ....111
6.4 Slot type decision flowchart .............................................................................................116
6.5 Admitted flow number ..........................................................................................................119
6.6 Redundant Slot Time of the served packets .........................................................................120
6.7 The wireless network topology used in the simulation .........................................................121
6.8 The speech source model implemented in OPNET simulator .............................................121
6.9 Packet Drop Probability in the uplink/downlink partitioning scheme ..................................122
6.10 Bandwidth allocated to random traffic in the uplink/downlink partitioning scheme ..........124
6.11 Used bandwidth rate for RP slot issuing ............................................................................125
6.12 Comparison of packet drop probability between the uplink/downlink partitioning and sharing scheme ............................................................................................................................125
6.13 Comparison of the total bandwidth allocated to random traffic between uplink/downlink partitioning and sharing scheme ............................................................................................................................126
6.14 Simulation results for uplink random traffic service with various allocation rates in a surplus period - 4 speech uplink flows are connected ..............................................................................130
6.15 Simulation results for uplink random traffic service with various allocation rates in a semi-deficit period - 18 speech uplink flows are connected ..............................................................................131
6.16 Simulation results for uplink random traffic service with various allocation rates in a deficit period - 34 speech uplink flows are connected ..............................................................................132
6.17 Simulation results for uplink random traffic service with the same allocation rate in a surplus period - 3 speech uplink flows are connected ..............................................................................133
6.18 Simulation results for uplink random traffic service with the same allocation rate in a semi-deficit period - 19 speech uplink flows are connected ..............................................................................134
6.19 Simulation results for uplink random traffic service with the same allocation rate in a deficit period - 35 speech uplink flows are connected ..............................................................................135
6.20 Simulation results for uplink random traffic service with greedy AC flows in a semi-deficit period - 18 speech uplink flows are connected ..............................................................................136

7.1 Service distortion due to the virtual time lag .........................................................................143
7.2 Simulation configuration ........................................................................................................144
7.3 The F(t) and V(t) in WFQ of BS1 .........................................................................................146
7.4 The gap between F(t) and V(t) (F(t) - V(t)) in WFQ of BS1 .................................................147
7.5 Service order of flow 8 and 9 in WFQ of BS1 .......................................................................147
7.6 Estimation of correct virtual time at a handover ....................................................................148
7.7 The gap between F(t) and V(t) in corrected WFQs in BS1 ..................................................149
7.8 Service order of flow 8 and 9 in corrected WFQs in BS1 ................................................................. 150
7.9 Out-of-order packet delivery case example occurring in the Caceres’s handover protocol
upon a handover ........................................................................................................................... 153
7.10 The proposed handover protocol .................................................................................................... 155
7.11 The packet numbers of flow 1 and 2 arrived at corresponding MHS .............................................. 156
7.12 End to End delay of flow 1 and 2 .................................................................................................... 156
List of Tables

2.1 Comparison of PHS, wireless LAN, GSM, and CDPD ...............................................................30
2.2 Encoding schemes of DAAT .........................................................................................................34
6.1 System Parameters .........................................................................................................................122
6.2 Random traffic flows used in the rate-based service simulation ..............................................127
6.3 Random traffic flows used in the local fairness simulation .........................................................128
6.4 Random traffic flows used in the global fairness simulation ......................................................129
7.1 The parameters for flows used in the simulation ..........................................................................145
7.2 Assignment priority for PCAWFQ and PCAWFQ’ ...................................................................151
8.1 The scheduling disciplines used for each class ..........................................................................160
### Glossary of Terms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAF</td>
<td>Application Adaptation Function</td>
</tr>
<tr>
<td>AC</td>
<td>Absolute Class</td>
</tr>
<tr>
<td>AMPS</td>
<td>Advanced Mobile Phone Service</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BA</td>
<td>Bounded Adaptive</td>
</tr>
<tr>
<td>BARWAN</td>
<td>Bay Area Research Wireless Access Network</td>
</tr>
<tr>
<td>BC</td>
<td>Best-effort Class</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CAWFQ</td>
<td>Combined Absolute and Weighted Fair Queuing</td>
</tr>
<tr>
<td>CBQ</td>
<td>Classed Based Queuing</td>
</tr>
<tr>
<td>C-CRA</td>
<td>Centralised-Collision Resolution Access</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CDPD</td>
<td>Cellular Digital Packet Data</td>
</tr>
<tr>
<td>C-PRMA</td>
<td>Centralised Packet Reservation Multiple Access</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access with Collision Detection</td>
</tr>
<tr>
<td>DAAT</td>
<td>Dynamically Adaptable Audio Tool</td>
</tr>
<tr>
<td>DFWMAC</td>
<td>Distributed Foundation Wireless Media Access Control</td>
</tr>
<tr>
<td>DP</td>
<td>Data transmission slot for Periodic Traffic</td>
</tr>
<tr>
<td>DR</td>
<td>Data transmission slot for Random Traffic</td>
</tr>
<tr>
<td>DSMA/CD</td>
<td>Digital Sense Multiple Access with Collision Detect</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FCFS</td>
<td>First Come First Served</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FQMA</td>
<td>Fair Queuing Multiple Access</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GPRS</td>
<td>Global Packet Radio Service</td>
</tr>
<tr>
<td>GPS</td>
<td>Generalised Processor Sharing</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>HSCSD</td>
<td>High Speed Circuit Switched Data</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>I-ISMA</td>
<td>Idle Signal Multiple Access for Integrated services</td>
</tr>
<tr>
<td>IMT2000</td>
<td>International Mobile Telecommunication by the year 2000</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ISPN</td>
<td>Integrated Service Packet Network</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LAP</td>
<td>Low-power Access Protocol</td>
</tr>
<tr>
<td>LBA</td>
<td>Lower-Bounded Adaptive</td>
</tr>
<tr>
<td>MAC</td>
<td>Multiple Access Control</td>
</tr>
<tr>
<td>MET</td>
<td>Maximum Endurable Time</td>
</tr>
<tr>
<td>MH</td>
<td>Mobile Host</td>
</tr>
<tr>
<td>MRSVP</td>
<td>Mobile ReSerVation Protocol</td>
</tr>
<tr>
<td>MSC</td>
<td>Mobile Switching Centre</td>
</tr>
<tr>
<td>MTSO</td>
<td>Mobile Telephone Switching Office</td>
</tr>
<tr>
<td>NA</td>
<td>Non-Adaptive</td>
</tr>
<tr>
<td>NBA</td>
<td>Non-Bounded Adaptive</td>
</tr>
<tr>
<td>PC</td>
<td>Periodic Class</td>
</tr>
<tr>
<td>PCAWFQ</td>
<td>Packet-by-Packet Combined Absolute and Weighted Fair Queuing</td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Code Modulation</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PDT</td>
<td>Packet Drop Time</td>
</tr>
<tr>
<td>PEDQ</td>
<td>Packet Endurable Delay Queuing</td>
</tr>
<tr>
<td>PFD</td>
<td>Packet Forwarding Dependent</td>
</tr>
<tr>
<td>PFI</td>
<td>Packet Forwarding Independent</td>
</tr>
<tr>
<td>PGPS</td>
<td>Packetised Generalised Processor Sharing</td>
</tr>
<tr>
<td>PHS</td>
<td>Personal Handyphone System</td>
</tr>
<tr>
<td>PPG</td>
<td>Proxy Packet Generator</td>
</tr>
<tr>
<td>PR</td>
<td>Polling Register</td>
</tr>
<tr>
<td>PRMA</td>
<td>Packet Reservation Multiple Access</td>
</tr>
<tr>
<td>QoA</td>
<td>Quality of Application</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
</tbody>
</table>
RP Reservation request slot for Periodic traffic
RPS Rate-Proportional Server
RR Reservation request slot for Random traffic
RS Request Subslot
RST Redundant Slot Time
RSVP ReSerVation Protocol
RTP Real-time Transport Protocol
SAC Slot Allocation Controller
SCFQ Self Clocked Fair Queuing
SFQ Start Fair Queuing
STII+ Stream Protocol, Version 2+
STS Slot Type Selector
SVC State Confidence Value
TCP Transmission Control Protocol
TDM Time Division Multiplexed
TDMA Time Division Multiple Access
TELNET Telnet protocol and options
UBA Upper-Bounded Adaptive
UDP User Datagram Protocol
UMTS Universal Mobile Telecommunication System
USB Uplink-slot Start Beacon
VT Virtual Time
V-WFQ Variable-Weighted Fair Queuing
WC Weighted Class
WFQ Weighted Fair Queuing
WLL Wireless Local Loop
WRR Weighted Round-Robin
WWW World Wide Web
Chapter 1

Introduction

1.1 Integrated services in a cellular data network

With rapid development being made in the area of wireless communications, a mobile user is able to access data communication networks transparently from anywhere at any time through a mobile network. A mobile user typically uses a powerful PDA (Personal Digital Assistant) or a notebook computer that is equipped with wireless communication capability to connect mobile networks. The mobile network environment allows a mobile user to receive a wide variety of services such as video-telephony, FTP, TELNET and WWW (World Wide Web) along with voice telephony over a wireless medium while he or she is on the move. To enable a user to access such services freely, a mobile network should provide the user with not only a single service at a time but also with integrated services that can supply multiple services simultaneously.

The IETF (Internet Engineering Task Force) defined the term integrated services [15] as an Internet services model that includes best-effort service, real-time service and controlled link-sharing service. The traffic managed by the controlled link-sharing service can be divided into a few administrative classes, and each of which may represent different user groups or different protocol families and can receive a certain allocation of bandwidth which is guaranteed even during overload periods, while allowing "unused" bandwidth to be made available to other classes. We use the term integrated services as defined by the IETF.

A typical mobile network is a cellular network that can support a seamless connection while a user moves. In a cellular network, some characteristics of a wireless link such as multipath fading, co-channel interference, and noise disturbance degrade the transmission ability of a wireless link. In addition, a cellular network has another difficulty, entity mobility among cells which causes fluctuation of a cell load. Therefore, a cellular network is characterized by a dynamic resource environment, where available bandwidth varies owing to the physical characteristic and the mobility. In such a dynamic environment, the third-generation cellular network [41] should be capable of accommodating the integration of data sources with various QoS (Quality of Service) requirements. An efficient and fair usage of available bandwidth is necessary to provide those sources with their required QoS in such a network.
Two straightforward approaches are deployed as an infrastructure for a cellular data network: a cellular circuit switched network based on an assigned channel scheme and a cellular packet switched network based on a random access scheme. The two types of network provide quite different QoS for a user because of their nature. A cellular circuit switched network has rigid service boundaries, in which, however, bandwidth may be wasted when an allocated channel is not used. In contrast, a cellular packet switched network can maximize channel utilisation by sharing available bandwidth among active users. Considering the relatively lower bandwidth of a wireless link, a cellular packet switched network is more appropriate for accommodating integrated services in terms of service flexibility and economic channel usage than a cellular circuit switched network. However, it is not easy for the cellular packet switched network to guarantee QoS to a connection owing to the channel sharing with other connections.

1.2 The barriers in a cellular packet switched network

In a cellular packet switched network, bottlenecks arise due to quality discrepancies between a wired and a wireless link. For instance, the throughput of current wireless LANs [47] is one or two orders lower than the widespread 10 Mbits/s or 100 Mbits/s Ethernet. For a higher bandwidth wireless network, some research projects such as ACTS Magic WAND [62] and CITR ATM LAN [31] have been aiming at 20 Mbits/s in the 17 GHz frequency range and 160 Mbits/s in 20-60 GHz range each. However, shifting to microwaves brings about two problems besides technological difficulties. Firstly, the attenuation characteristic of $20\log f$ and its similarity to optical transmission restrict transmission to within a room [3] - where $f$ is the operating frequency. Secondly, silicon elements do not operate at such high frequencies, so expensive Gallium Arsenide (GaAs) must be used - which makes it difficult to produce low cost mobile terminals. Even if such research is successful, they will still be lower bandwidth than the wired Gigabit research networks. After all, because of the restriction of available radio spectrum for wireless data communication, we cannot avoid the quality and quantity discrepancy between a wireless and wired network.

Considering the bandwidth discrepancy, we can expect that a base station experiences bottlenecks induced by data packets originating from sources in wired networks being sent to mobile hosts if the wireless link capacity is smaller than the required bandwidth for the current connections, and a transport protocol does not have a congestion control scheme – for instance RTP (Real-time Transport Protocol) [82] or UDP (User Datagram Protocol) [25]. This results in many packets from higher speed wired networks being buffered at the base station and eventually being dropped. Therefore, we need to solve the bottleneck problem.
An admission control may solve the problem, limiting the number of connections to provide enough resources for admitted connections. However, it cannot guarantee a fair bandwidth distribution as connections require if the available bandwidth is fluctuating. The bandwidth fluctuation arises for various reasons: channel contention among mobile hosts, transmission errors due to physical environment, and an overlay network where the unused channels of an existing cellular phone system are used for data connection (e.g. CDPD [21], GPRS [20] and PRMA [42]). In such an available bandwidth fluctuating system, unfair services may occur in a downlink channel - which conveys data to mobile hosts from a base station - unless a packet scheduler provides fairness. In an uplink channel, a multiple access protocol may allocate channels to connections unfairly. Therefore, we should furnish fair bandwidth sharing schemes for both uplink and downlink to provide a connection with its required QoS.

In addition, a connection in a cellular data network may experience handovers during its lifetime. With respect to fair bandwidth distribution, two problems occur during a handover [51], [52] - unbounded fairness among connections in the old base station and out-of-order packet delivery. The problems should be tackled to provide seamless delivery of integrated traffic over a cellular packet switched network with integrated services.

1.3 Contributions

There has been no lack of research effort on integrated services for wired packet networks. However, these methods cannot be directly applied to a cellular packet switched network which has different characteristics compared to a wired network (e.g. limited bandwidth, longer delay and mobility). This thesis contributes to an integrated service architecture and its design issues for a cellular packet switched network in which QoS for uplink/downlink of diverse traffic connections can be reserved and QoS of a connection is guaranteed in the mobile environment, while maintaining fairness among connections.

1.3.1 Scheduling Algorithm

To schedule various delay-tolerant traffic in a downlink, we proposed a novel scheduling scheme, called CAWFQ (Combined Absolute and Weighted Fair Queuing). CAWFQ is a general packet scheduler designed for a variable-rate link, so it can be exploited in any switch with varying bandwidth on the output link. It can simultaneously schedule three kinds of flow classes1 - Absolute, Weighted and Best-effort class - according to the policy for each class in a variable-

1 We refer here to a connection with a QoS reservation as a flow.
rate link. The absolute class flows are guaranteed a minimum bit rate service even when the available bandwidth fluctuates, provided that available bandwidth is higher than the total bandwidth allocated to the absolute class flows. The weighted class flows can receive a rate-based scheduling service within the range of residual bandwidth after serving the absolute class flows. The best-effort class can exploit any remaining bandwidth following servicing of absolute and weighted class flows.

We adopt two kinds of virtual time - absolute and weighted virtual time - in order to track the bandwidth usage of the absolute and weighted classes respectively. The virtual times enable the bandwidth for the absolute class to be guaranteed even under conditions of fluctuating available bandwidth. Since CAWFQ is designed for a fluid-flow network [71], a packet by packet version of this scheme called PCAWFQ (Packet-by-Packet CAWFQ) is also presented. The performance is analyzed and evaluated empirically using the OPNET simulator. The simulation results show that absolute class served by PCAWFQ can receive a committed constant bit rate service even during link bandwidth fluctuations, while the received bandwidth of the weighted class are fluctuated as link bandwidth varies. Therefore, it is especially suitable for providing a guaranteed bandwidth service in a varying rate link such as a cellular packet switched network.

1.3.2 Multiple access protocol

We propose a reservation multiple access protocol called FQMA (Fair Queuing Multiple Access) to provide integrated services in an uplink. As a slot-based protocol, FQMA handles integrated traffic, decoupling delay-tolerant and delay-intolerant traffic. Frame-based reservation access protocols [4] can support easily the decoupling, since it can determine empty slots in each frame that are not used by delay-intolerant traffic. Thus, delay-tolerant traffic can use the empty slot without degrading the service quality for delay-intolerant traffic. In contrast, a slot-based (non-framed) protocol has no definitive frame, so it needs a more complicated scheme to protect delay-intolerant traffic against the interference of delay-tolerant traffic. FQMA is a slot-based protocol to perform this decoupling.

FQMA embodies two scheduling disciplines to schedule delay-intolerant and delay-tolerant traffic respectively. We propose PEDQ (Packet Endurable Delay Queuing) to schedule delay-intolerant traffic in FQMA. PEDQ serves packets in ascending order of remaining lifetime of packets, and yields parameters to estimate the empty slots and the generation time of the next served packet. With the parameters, FQMA serves delay-intolerant traffic as long as it has packets to send, assigning empty slots to delay-tolerant traffic. The servicing using PEDQ maximises the remaining lifetime of delay-intolerant packets, so that more of these packets can be
served even during transient overload periods. In addition, this technique is capable of servicing more active delay-intolerant flows than the channel capacity for a short period without packet drops since the maximised remaining lifetime functions as a bandwidth buffer in the face of traffic congestion.

For scheduling delay-tolerant traffic in FQMA, we propose PCAWFQ' which performs during the slots allocated to delay-intolerant traffic by FQMA. PCAWFQ' is a modified version for multiple queues of PCAWFQ. Because PCAWFQ is a scheduling discipline for a single queue, it cannot be directly exploited as a scheduler for a multiple access protocol where real packets are distributed in the mobile host queues. To deal with the distributed queue problem, we introduce proxy packets, which represent their corresponding real packets, to serve distributed packets. A proxy packet is generated when an uplink flow is admitted and queues into the PCAWFQ' in place of the real packet. PCAWFQ' serves the proxy packets with the same scheduling policy as PCAWFQ. After being served, a proxy packet re-queues to the PCAWFQ' representing the next real packet.

With regard to channel efficiency and load balancing, an uplink/downlink-partitioning scheme is inefficient for the integrated services network where uplink and downlink traffic are asymmetric. To enhance efficiency, an uplink/downlink-sharing scheme for a cellular packet switched network is adopted in designing FQMA. In the scheme, the uplink and downlink share a common channel and exploit channel slots in proportion to their demands. We simulate FQMA under both the uplink/downlink sharing and partitioning schemes. The simulation results with partitioning show that FQMA has lower packet drop probability than frame based protocols [4], while it keeps a similar channel allocation rate for delay-tolerant traffic compared with the frame-based protocols. FQMA with the sharing scheme shows better performance than FQMA with the partitioning scheme. If we consider the uplink/downlink pairs serving a mobile end system in the sharing scheme, unused slots of one link can be exploited by the other link. Thus, it reduces packet drop probability of delay-intolerant traffic dramatically at the cost of reducing delay-tolerant traffic bandwidth at high loads. The bandwidth reduction is at most 1%. These results show that FQMA has better performance than the frame based protocols and performs even better in the uplink/downlink-sharing scheme, as well as its comprehensive services based on 4 classes using dual scheduling disciplines which is not the case in the framed based protocols.

### 1.3.3 Handover

We also investigate and analyze the proposed system in the handover environment. We found that unfairness occurs due to the forwarding of packets during handovers in certain type of
schedulers such as [9],[30],[71] including PCAWFQ. A compensation scheme is suggested to fix the problem. We also identified that out-of-order packet delivery may occur during handovers when a handover protocol operates without considering the packet forwarding. We propose a new handover protocol which take into account the packet forwarding to remedy the out-of-order packet delivery phenomenon. The simulation results show that the unfairness occurring due to a handover is cumulative and the out-of-order packet delivery occurs during a handover. Finally, the simulation results also show that our proposed scheme can solve those observed problems.

1.4 Dissertation outline

The main objective of this thesis is to identify the obstacles in providing integrated services in a cellular packet switched network and proposes a new service architecture, which accommodates the integrated services with guaranteed QoS. The outline of this dissertation is as follows: Chapter 2 describes advantage and disadvantage of various types of cellular data networks with regards to integrated services, raises issues to be solved for guaranteed QoS in such networks and defines the scope of this research. Chapter 3 reviews related work on the research issues raised - multiple access protocols, scheduling disciplines and handover protocols - and analyses them as potential solutions. In chapter 4, we categorize traffic based on its characteristic to identify what kinds of services are required for integrated services. Based on the traffic analysis, we propose an integrated services architecture for a macro/pico-cellular packet switched network that consists of FQMA protocol and several scheduling disciplines. The uplink/downlink-sharing scheme to improve performance is also presented here. Chapter 5 presents the proposed scheduling scheme, PCAWFQ which is used in multiplexing downlink slots and is a basis to design its distributed version, PCAWFQ' to serve uplink traffic. In chapter 6, we present the FQMA protocol which is a multiple access protocol to enable mobile hosts to receive their allocated services. The two scheduling disciplines, PEDQ and PCAWFQ' which are the core service disciplines of this protocol are also described, along with the proxy packet concepts associated with the scheduling disciplines. In chapter 7, we analysis handover effects on our proposed system and propose compensation schemes to fix the negative effects and a packet-forwarding aware handover protocol to remove out-of-order packet delivery caused by handovers. In chapter 8, our conclusions are discussed and further work to be researched in the future is presented.
Chapter 2

Background

In this chapter, we first introduce various types of cellular data networks and describe the advantages and disadvantages of each network in terms of integrated services. Following that, we review a set of published approaches from link layers up to application layers to provide QoS for integrated traffic in a cellular data network comparing to those in a fixed network. Finally, we derive our research motivation of this thesis from the procedure and confine our research scope.

2.1 Cellular data networks

There are many different types of cellular data networks, and many different approaches have been tried over the last ten years. Cell size determines various features of a cellular network. For instance, a large cell network several km in diameter cannot exploit the frequencies in the microwaves band to support high bandwidth service because of limited propagation distance of high frequency waves. However, as a counterbalance, a large cell network encounters less frequent handovers than a smaller cell network, which reduce handover management burden. Here, we describe commercial or research cellular data networks classifying them into 2 groups according to a cell size – macro (> 1 km) and micro/pico (< 1 km) cellular data networks, considering their advantage and disadvantage for integrated services.

2.1.1 Macro-cellular data network

Two categories of commercial macro-cellular data networks exist today - a standalone system where the entire bandwidth is dedicated to the transmission of data packets (e.g. ARDIS [58], Mobitex [58]), and an overlay system, where the unused channels of an existing cellular phone network are used (e.g. CDPD [21], GPRS [20]). Macro-cellular data networks are widely deployed at the present. The supplied bandwidth of those systems is from 8 kbits/s up to 19.2 kbits/s at most, and long communication distance makes it have relatively high latency. Due to these characteristics, it is unlikely that macro-cellular networks are expected to attain dominance as the future cellular data networks for integrated services.
2.1.1.1 HSCSD (High Speed Circuit Switched Data) - GSM

GSM (Global System for Mobile Communications) [76] is a cellular circuit switched network, and is currently available commercially including many European countries. The GSM can support a variety of data transmission services including the connection of ISDN and packet switched public data networks, along with a voice connection. GSM users are currently limited to 9.6 kbits/s access to data communications services. However, two new service classes under development for GSM will expand the current data rate. In the extended GSM specification called GSM Phase 2+, which is currently being standardised, data over GSM can be transmitted in two ways: by circuit switching as in today's voice technology, using HSCSD [81], or by packet switching, using GPRS (Global Packet Radio Service) [20]. Whereas HSCSD is based on the same physical layer as today's GSM services, GPRS requires new nodes to handle packet switched data.

HSCSD uses the compression techniques of the V.42bis protocol allowing data rates to be increased to 19.2 kbits/s and in the optimum case 28.8 kbits/s. Even more radical changes are anticipated using multiple TDMA time slots for data connections. Multi-slot operation could even allow 64 kbits/s ISDN connections. Such high data rates would pave the way for integrated services. However, HSCSD is a service based on a connection-oriented network, exploiting FDMA and TDMA for its multiple access protocol. Though the scheme guarantees a continuous connection to an entity once admitted, its constant bit rate service does not efficiently handle bursty traffic streams, which are typical of most data sources.

2.1.1.2 GPRS (Global Packet Radio Service)

Bursty data traffic over a circuit switched system such as GSM results in an inefficient use of the channel as the full channel bandwidth is reserved for each user throughout the entire connection. GPRS is a developing standard that allows GSM to become a packet switched network to improve network utilisation, enabling multiple users to share the same radio channel. Thus, it can allocate channels more efficiently to data connections as well as voice calls. GPRS will handle rates from 14kbits/s, using just one TDMA slot, up to 115kbits/s, using eight TDMA slots. Since GPRS is a packet-switching technique, it is more suited to the highly bursty nature of most data applications. However, as with HSCSD, GPRS is being designed to work within the existing GSM infrastructure, so a voice call is still connection oriented that does not efficiently use channels. As it is not truly a packet switched network, it will be used as an interim toward the third generation cellular network such as UMTS/IMT2000 [10].
2.1.1.3 CDPD (Cellular Digital Packet Data)

CDPD [21] was developed by IBM and released by a consortium of the major United States cellular telephone carriers. It is implemented under the existing infrastructure of the analogue AMPS (Advanced Mobile Phone Service) cellular telephone network that is the analogue standard in the USA. To support a connectionless data packet service, it utilises idle channels in the AMPS system. CDPD assigns unused voice channels to data connections, charging for the number of packets to be sent [47]. The service operates at a raw data bit rate of 19.2 kbits/s, but the actual available bandwidth is further reduced by contention in gaining access to channels since it adopts a DSMA/CD (Digital Sense Multiple Access with Collision Detect) [79] as a multiple access protocol. Although CDPD has smaller cell size than GSM, the channel sharing with voice connections, in which voice service always has higher priority than CDPD, results in CDPD having higher latency than GSM. Since all users are served based on a best effort basis, it does not guarantee a constant bit rate all the time and cannot provide demand based QoS. Therefore, it is not a suitable model for the integrated services that we are pursuing.

2.1.2 Micro/Pico-cellular data network

A micro-cellular network consists of microcells that divide a large geographical area into small sub-regions. Each microcell has a diameter of the order of several hundred metres, and contains a base station that provides a connection point for mobile users in the microcell as depicted in Figure 2.1. This smaller cell size yields several advantages. Firstly, reducing the cell size increases system capacity proportional to $1/r^2$ ($r$ = cell radius) [80]. Secondly, a higher frequency band can be utilised to accommodate higher bandwidth service unlike a macro one. Finally, data can be transmitted with lower power, which promises a longer battery life.

![Figure 2.1 Micro-cellular Network](image)

**Figure 2.1 Micro-cellular Network**
A pico-cellular network is similar to a micro-cellular network except that its cell size is now of the order of 10 metres \[36\] and is typically a room in a building as depicted in Figure 2.2. The smaller cell size causes very rapid movement between cells, so it produces more problems of tracking and maintaining communication between mobile users and a pico-cellular network. For instance, assuming that an user moves with a speed of 2 to 3 metres/sec in a corridor and the cell diameter is 10 metres, the cell latency of a mobile user (the time an user stays in a cell) is of the order of 4 - 5 seconds. Another characteristic of a pico-cellular network is the very rapid changing population of mobile users per cell depending on the locations (e.g. conference room, classroom or private room).

![Figure 2.2 Pico-cellular network](image)

Since the available radio spectrum is limited, a micro/pico cellular architecture is likely to be the basis of future cellular data systems to support high capacity integrated services even though it has innate faster inter-cell node mobility of mobile hosts, which requires different approaches in routing, tracking and handovers compared to those in macro one.

2.1.2.1 Wireless LAN

Wireless LAN technology is derived from the Ethernet LAN technology. For that reason, most wireless LAN products adopt a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol which is similar to the Ethernet CSMA/CD (Carrier Sense Multiple Access with Collision Detection) LAN protocol. Both CSMA/CD & CSMA/CA work by a “listen before talk” scheme. This means that a mobile host wishing to transmit must first sense the radio channel to know whether another mobile host is transmitting. If the medium is not busy, the transmission may proceed. The difference between both is whether it can listen even during talk to stop transmitting as soon as collision occurs. While CSMA/CD can do it, CSMA/CA cannot stop signal transmission once it has started. Thus, CSMA/CA performs worse than CSMA/CD. The
reason that wireless MAC cannot use CSMA/CD is that it is difficult to listen while sending with a single antenna. Even if two antennas are installed, it is difficult and expensive to implement it since the transmitted signal is much stronger than the received one.

The IEEE project 802.11 is establishing a recommended international standard for wireless LANs. The final version of its MAC protocol has yet to be finalised, but it is expected that CSMA/CA will be chosen as the basis for the protocol. The 802.11 standard supports two primary network topologies: infrastructural and ad-hoc network. The former is efficient for mobile hosts accessing the backbone network via base stations. The latter is designed for groups of mobile hosts communicating directly each other independently of existed networks. Supporting the infrastructural network in the wireless LAN standard is a sub function of contention based communication; i.e. basically it is not designed for an infrastructural network. Thus, it is not suitable for providing a connection accessing a backbone network with guaranteed QoS, which can be efficiently achieved in an infrastructural network.

2.1.2.2 PHS (Personal Handyphone System)

PHS [99] is a micro-cellular network developed in Japan to support very-high density pedestrian traffic and WLL (Wireless Local Loop), targeting several million subscribers in an urban setting. This is one of first commercialised micro-cellular networks covering a whole urban area and serving a large number of subscribers. To achieve the system economically and quickly, the existing network is utilised rather than building a separate network afresh, as has been done for cellular services such as GSM. It has been quite successful in Japan since its commercial service was launched in 1995. The PHS market has over 7 million subscribers in Japan in 1997 and is expanding to other countries.

PHS provides users with voice connections and data connections, adopting a TDMA protocol as a multiple access protocol. It assigns a 32 kbits/s channel to a data connection. One of its distinctive features is its low power consumption. Since its output power ranges from 20mW to 500mW according to the number of users in the area to be served, typical talk times is up to seven hours and stand-by times up to 550 hours. A seamless handover is also provided, but due to the small cell size it cannot support fast moving users. Its range is up to 50 mile per hour. Though it requires the installation of many base stations, a base station weighs only 3–5kg and has a volume of just 3 – 4 litres (A4 size). As a result, a cell station can be installed economically in wide variety of locations. Indoor service areas, such as in department stores, are much smaller per cell station than their outdoor counterparts, because it is difficult for radio waves (1.9 GHz) of low power (20 mW) to propagate through enclosed spaces partitioned by floors or walls.
Although PHS has several beneficial features, it cannot assign bandwidth to a data connection based on demand as it is not a packet switched network. Therefore, it is not suitable for the third generation network that should support the demand-based services.

2.1.3 Hierarchical cellular network

Current cellular technology varies widely in bandwidth, latency, coverage, and media access methods. A macro-cellular data network provides a low-bandwidth service with high latency over a wide geographic area while a micro/pico-cellular network provides a high bandwidth service with low latency over a narrow geographic area. No single network simultaneously supports a high bandwidth service with a low-latency over a wide-area to a large number of users. Therefore, any future cellular network will be built upon heterogeneous overlay networks with hierarchical structure combining various wireless networks to accommodate a wide range of services, performing efficient resource allocation.

2.1.3.1 UMTS/IMT2000

Telecommunication-oriented services are still dominant in the second-generation cellular networks based on the GSM standard now that first-generation/analogue voice cellular networks are declining. The third generation network, referred to as UMTS (Universal Mobile Telecommunication System) by ETSI (European Telecommunications Standards Institute) or IMT2000 (International Mobile Telecommunication by the years 2000) by ITU (International Telecommunication Union), are being designed to carry multimedia traffic such as video, images, files of data, or combinations of these [77]. These third generation systems aim to support a wide range of services from voice and low-rate data up to high-rate data services with up to 144 kbits/s in vehicles, up to 384 kbits/s in outdoor and up to 2 Mbits/s in indoor and pico-cell environment [10]. The hierarchical cell structure is being considered to maximise network capacity in the current specification of UMTS. Two arrangements for the hierarchical cell structure will be selected by many operators: concentric cells and macro/micro layers. In the concentric cells, underlay and overlay are used. The underlay cell is equipped with traffic and signalling channels, while the overlay cell is assigned only traffic channels. All the signalling procedures are necessary for accessing a traffic channel. Upon completion of the preliminary signalling phase, the base station evaluates which layer the mobile host will be linked to during the active call. In the latter approach, microcells are embedded in a micro cellular system and can provide improved coverage and performance, increased system capacity, and delivery of innovative value-added services.
2.1.3.2 BARWAN (Bay Area Research Wireless Access Network)

BARWAN [56], which is a research network being developed in University of California, Berkeley, integrates and inter-operates across wide-area, metropolitan-area, campus-area, in-building, and in-room wireless networks. The overlay inter-network management allows mobile applications to operate across a wide range of network performance, and to choose a most suitable network among alternative overlays for best performance given the current network state and application requirements. The network is being researched in a wireless overlay network testbed being created in the San Francisco Bay Area by collaboration with various wireless network providers. The research includes diverse aspects that should be handled properly in a hierarchical network such as rough user tracking, vertical handovers [84], overlay cooperation, continuous connectivity, low-latency handovers within and between overlays, network load sharing between overlays, and dynamic reallocation of network resources to the areas of high user density. Since a handover across overlays changes an application's network bandwidth and latency, they are designing a new applications interface to the network management layers to allow them to initiate handovers, to determine changes in their current network capabilities, and to gracefully adapt to their communications demands.

2.1.4 Summary

As described above, there are two basic technical directions for cellular data networks aimed at improving either coverage or throughput. A macro-cellular network such as CDPD [21] covers a wide range but only provides low throughput and relatively high latency, and a micro/pico-cellular network has small coverage but provides relatively high throughput and low latency. Table 2.1 shows a comparison of a few networks from both architectures. With regards to network switching types, a cellular packet switched network is more efficient than a cellular circuit switched network [3] in dealing with bursty data sources. With the limited radio spectrum, future cellular data systems should provide integrated traffic with guaranteed QoS, which need higher bandwidth and flexible distribution. Therefore, micro/pico-cellular architecture with a packet switching scheme is suitable for the future cellular data systems to provide the QoS guaranteed integrated services. The third generation cellular networks by ITU and ETSI are being standardized adopting the packet switched scheme for data connections, along with the circuit switched scheme for voice connections. If a packet switched scheme is applied to voice connections as well, the channel efficiency increases dramatically at the cost of degradation of
voice quality [41]. However, they include the circuit switched service because of backward compatibility and service diversity.

Table 2.1 Comparison of PHS, wireless LAN, GSM, and CDPD

<table>
<thead>
<tr>
<th>Available Products</th>
<th>Micro</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Rate</strong></td>
<td>32kbits/s</td>
<td>2Mbits/s</td>
</tr>
<tr>
<td><strong>Max Cell Radius</strong></td>
<td>0.3 - 1 Km</td>
<td>240m</td>
</tr>
<tr>
<td><strong>Latency</strong></td>
<td>1-30msec</td>
<td>200msec</td>
</tr>
<tr>
<td><strong>Access Protocol</strong></td>
<td>TDMA</td>
<td>CSMA/CA</td>
</tr>
<tr>
<td><strong>Network Switching type</strong></td>
<td>Circuit Switched</td>
<td>Packet Switched</td>
</tr>
</tbody>
</table>

For conveying packets of burst streams, a packet switched system adopts a random multiple access protocol, such as CSMA/CA, DFWMAC (Distributed Foundation Wireless Media Access Control) [95], PRMA [42] and DSMA/CD (Digital Sense Multiple Access with Collision Detect) [47]. Those protocols exploit a contention algorithm to share a channel among mobile hosts. A macro-cellular data network has simple base stations that do little processing. Most functions, including a handover decision, are executed in the central service centre (MTSO in the case of AMPS and MSC in the case of GSM). In contrast, the base stations in a micro/pico-cellular network have the ability to control all kinds of functions by themselves in a distributed fashion without any help of a central service centre. It is natural for a base station to have such heavy control functions to manage the high volume of handovers efficiently.

2.2 QoS provision in an integrated services packet network

One of the important issues in an ISPN\(^2\)(Integrated Services Packet Network) is to provide a variety of traffic classes with the wide range QoS that they require. The traditional data service architecture underlying computer networks has no facilities for pre-scheduling resources or denying service on overload. Thus, it is unable to guarantee QoS. Recently, research into

\(^2\) Clark [F13] defined an integrated services packet network (ISPN) as a packet network that can support integrated services.
providing guaranteed QoS has actively been done. Three approaches are suggested. One is to
develop a new network architecture [37] which guarantees QoS, and applications simply request
their requirements. Another approach is to make applications [14], [12] adapt to current network
conditions to maximise the quality of the data delivered to the destinations. The other approach is
the predicted service [23] that combines above two approaches, in which the network can roughly
commit QoS through estimation of what kind of service it can deliver based on recent
measurement on the traffic load and an application can also adapt to violations of the QoS
commitment. Each approach has its advantages and disadvantages. In this section, we will present
several important elements in those approaches, especially laying stress on the first approach
since, as indicated in Chapter 1, this research focuses on the service architecture to provide
guaranteed QoS for integrated traffic.

2.2.1 Two procedures for the first approach

QoS provision by the first approach is achieved via two procedures. The first procedure is
performed by a signaling protocol such as RSVP (ReSerVation Protocol) [102] or STII+[28],
which enables an application to negotiate with each router along the possible data path so as to set
up a connection with its required QoS. Although the signaling protocol passes around the QoS
message to the routers along the path above a network layer, it does not practically distribute
resources to the flows at lower layers. Therefore, we need a second procedure, which performs
the resource distribution according to the QoS messages forwarded from the signaling protocol. It
usually consists of admission control, resource allocation and packet scheduling. The resource
distribution can be divided into two categories - statistical guarantee [100] and deterministic
guarantee [26], [27]. The statistical guarantee achieves higher network utilisation than the latter,
but it promises less reliable QoS due to probabilistic nature, whereas the latter can provide a
deterministic guaranteed QoS by considering worst-case behavior. Packet scheduling and
admission control play important roles in achieving the deterministic guaranteed QoS and
statistical guaranteed QoS respectively.

2.2.2 Resource reservation protocol - RSVP

A resource reservation protocol enables the senders, receivers, and routers of communication
sessions to communicate with each other in order to set up the necessary router state. Of all the
resource reservation protocols designed for IP networks, RSVP [16] has the most industry support
[96]. RSVP is a kind of Internet control protocol designed for integrated services Internet,
operating on top of IPv4 or IPv6. It can operate with current and future unicast and multicast
routing protocols. A host uses the RSVP protocol to request a specific QoS to the network for a particular application’s data stream. Routers use it to deliver QoS requests to all nodes along the path of the flow and to establish and maintain state to provide the requested service. For accommodating heterogeneous receiver requirements efficiently, receivers are responsible for requesting a specific QoS [102]: i.e. RSVP is receiver-oriented. The receiver of a data flow initiates and maintains the resource reservation for the incoming flow to the receiver. Therefore, both ends of a connection request the QoS for its incoming flow separately.

A receiver host application passes a QoS request to the local RSVP process. The RSVP protocol then carries the request to all the nodes along the reverse data path(s) to the data source, consulting the local routing database(s) to obtain routes. During the reservation setup, a RSVP QoS request is passed to two local decision modules: "admission control" and "policy control". Admission control determines whether the node has sufficient available resources to supply the requested QoS. Policy control determines whether the user has an administrative permission to make the reservation. If both the checks succeed, parameters are set in the packet classifier and in the link layer interface (e.g., in the packet scheduler) to produce the desired QoS. RSVP establishes "soft" state: i.e. RSVP sends periodic refresh messages to maintain the state along the reserved path(s). In the absence of refresh messages, the state automatically times out and is deleted. It provides graceful support for dynamic membership changes and automatic adaptation to routing changes.

2.2.3 Resource distribution and Packet scheduling scheme

While a reservation protocol is a signaling protocol to set up a resource reservation for a data connection in above IP layer, a packet scheduling discipline is a practical device to distribute bandwidth to the flows based on their requested QoS in below network layer. Current packet switched networks (such as the Internet) use FCFS (First Come First Served) queuing as a scheduling discipline. It is very simple and works well where the bandwidth is large enough to support all flows without congestion. However, a sophisticated scheduling scheme is needed to provide deterministic guaranteed services at the switch level where congestion frequently occurs. Thus a packet scheduling discipline at a switch node is one of the most important issues in supporting deterministic guaranteed QoS in an ISPN, and many packet-scheduling disciplines [71], [30], [43], [39], [9], [85], [101], [55] have been developed.

Packets from different connections passing through the same output port of a switch interact with each other in the output buffer queue. To support a guaranteed QoS to those connections even during congestion periods, a switch should serve the packets of each guaranteed QoS
connection independently of all other traffic. A packet scheduling discipline can control the service order of each connection to achieve the QoS committed to the connections. It also prevents mis-behaving or malicious flows from using more than their reservation so that well-behaving flows can receive the committed QoS.

### 2.2.4 Adaptive applications

For a non-adaptive application which cannot tolerate any delay bound and/or committed bandwidth violation, QoS provision by a network service commitment is necessary. However, an adaptive application which can tolerate or adapt to packet delay and bandwidth reduction can keep yielding a certain quality against such service degradation. The adaptive applications are needed not only in the network without QoS provision but also in a measurement based QoS provided network. Normally, non-real time applications can adapt to the network congestion, while real-time applications cannot. Contrary to the popular view regarding real time applications, some are more flexible and can adapt to network conditions to maintain a certain quality. There are two kinds of real time applications: playback application and inter-active application.

#### 2.2.4.1 Playback application

A playback application plays received data from a remote source. The application adapts to network jitter by buffering the incoming data, and can often tolerate the loss of a certain fraction of packets with only a minimal distortion of the signal. In such applications, only the packets arriving before their associated playback points can be used to reconstruct the source signal, so one of crucial factors is the individual packet delay to decide the playback point such that all packets can arrive before their associated playback points. As long as the network passes the delay information to the playback application, it can adapt to the network jitter. Clark [23] defines an adaptive application as an application that can measure the network delay experienced by arriving packets and then adaptively move the playback point to the minimal delay that still produces a sufficiently low loss rate. Thus, it will suffer less performance degradation caused by delay. He also suggests a predicted service architecture, which measures the network delay bound and informs it to the application to adapt to the current condition.

#### 2.2.4.2 Inter-active application

Inter-active real-time applications such as an Internet phone and conference application cannot move the playback point since the user cannot tolerate long response times. Those applications can be adaptive by using control mechanisms that select different resolution of
audio/video coding and decoding processes based on the characteristics of the channel to maximise the quality of the audio and video delivered to the destinations. Bolot [14] developed a set of mechanisms that attempts to eliminate or at least minimise the impact of packet loss and delay jitter. In the scheme, the coding rate at which packets are sent over a connection is controlled to match the current transmitting capacity of the connection, which minimises packet loss eventually. It also adds redundancy information for error control into the audio packets to minimise the impact of packet error.

Table 2.2 Encoding schemes of DAAT

<table>
<thead>
<tr>
<th>Encoding name</th>
<th>Data rate[kbits/s]</th>
<th>Relative CPU cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>ADM6</td>
<td>48</td>
<td>13</td>
</tr>
<tr>
<td>ADM4</td>
<td>32</td>
<td>11</td>
</tr>
<tr>
<td>ADM2</td>
<td>16</td>
<td>9</td>
</tr>
<tr>
<td>GSM</td>
<td>13</td>
<td>1200</td>
</tr>
<tr>
<td>LPC</td>
<td>4.8</td>
<td>110</td>
</tr>
</tbody>
</table>

Bhatti [12] models a DAAT (Dynamically Adaptable Audio Tool) for the audio tool, rat (Robust Audio-Tool) [45] developed at UCL. The DAAT is capable of handling six voice-encoding schemes that have various rates as shown in Table 2.2, considering power consumption cost. He also proposes a network model called QoSSpace to support the adaptive applications. The QoSSpace issues reports that contain a SCV (State Confidence Value) for each flow of an application. The application then combines the SCVs with other application specific information to make adaptation decision using the AAF (Application Adaptation Function). The AAF incorporates user preferences and application requirements into an automatically controlled adaptation policy for controlling the operation of the DAAT application. It allows an application to make adaptation decisions in response to fluctuations in QoS seen by a flow.

2.3 QoS provision in a cellular packet switched network

Compared to a fixed network, a cellular packet switched network has additional features such as handover and slow wireless link. Since those features affect the service performance, they should be considered in designing the elements such as packet scheduling discipline or
reservation protocol for supporting QoS guaranteed integrated services. In this section, we will discuss some important issues relating those features in terms of QoS provision.

### 2.3.1 Bottlenecks and scheduling discipline

Bandwidth discrepancy between wireless and wired link causes a bottleneck at a BS (Base Station), which degrades the performance of transport protocols. A reliable transport protocol, such as TCP may properly deal with the bottleneck situation by means of activating the slow start and congestion avoidance algorithm that diminish the data transport rate at the source to reduce data influx into a BS. Recently Manzoni [63] and Caceres [18] reported that packet loss caused by wireless transmission errors and communication pause during handovers may falsely trigger the congestion control procedure of TCP, which results in significant reductions in throughput and unacceptable interactive delays. While proposing a solution to settle the false trigger problem in an under-loaded network, they did not show the mixed case where real and false congestion exists together. It is still questionable how well TCP congestion control adapts to the mixed cases.

Meanwhile, connectionless transport protocols without congestion control like RTP or UDP are mainly exploited for transmitting delay intolerant streams such as voice and video packets. These packets must arrive at the destination within bounded time to be valid. Many packets of a UDP flow may be discarded at the BS buffer before passing through a wireless link due to insufficient wireless bandwidth, if the transmission rate of the flow is not calibrated for a low speed wireless link – it could occur in multicasting flows that include mixed destinations of a wired and wireless network. Even though the low speed link is considered, the variable bandwidth caused by dynamic topology of a cellular network – for instance, a handover to a new congested cell can produce unexpected congestion to the flow – prevents a flow from persistently transmitting packets with low loss rate.

Accordingly, in order to control congestion and allocate bandwidth properly to various protocol flows, a solution for the bandwidth discrepancy between the wireless and wired network must be made. Moreover, in the context of a cellular network, there is another big issue, namely mobility, that complicates the problem further. A packet scheduling discipline for a BS can support fair sharing of a wireless link to provide a guaranteed QoS for a flow.

### 2.3.2 Uplink traffic service and multiple access protocol

A base station usually communicates with mobile hosts via two different types of communication channel: broadcast channel for downlink and multiple access channels for uplink. While a cellular circuit switched network assigns a fixed channel to a connection once admitted, a
cellular packet switched network allocates channels dynamically according to connection's demands. There is much literature regarding resource assignment methods [89], [57], [67] for a cellular circuit switched network which basically exploits an assigned channel access scheme to commit a secure connection to a mobile entity. In the context of the assigned channel scheme, the main performance measures are new call blocking probability and a dropping probability upon handover as it provides a secure connection in a cell once admitted.

In a cellular packet switched network, the bandwidth distribution for a downlink is rather simple, as mobile hosts can listen to the broadcast channel simultaneously. The broadcast channel simply conveys base-to-mobile data via TDM (Time Division Multiplexed) mode using various scheduling schemes. However, packets for an uplink are placed in distributed mobile hosts not in a single queue, so the uplink bandwidth distribution is not as simple as that due to the channel contention among mobile hosts. Thus, in a cellular packet switched network, a multiple access protocol is the essential technology to provide uplink flows with their QoS requirements. Many multiple access protocols [42], [69], [64], [97], [13], [54], [4], [83], [68] have been studied to support integrated services in cellular packet switched networks. We will describe the protocols in detail in the Chapter 3.

2.3.3 Mobility related issues

The fixed data network solutions for addressing, routing and flow-control are not applicable in a cellular data network due to user mobility. In particular, handover events occur at a much higher rate in a micro/pico cellular network owing to the smaller coverage area compared to a macro-cellular network. Many research challenges have been raised regarding it. In this section, we introduce mobility issues to be solved related to QoS.

2.3.3.1 Efficient handover protocol for seamless delivery

One of the challenges is the seamless delivery of integrated traffic over wide-area integrated services networks to and from mobile hosts. Due to time-sensitive nature, real-time traffic often requires throughput and delay guarantees to the transport system. In the wired networking arena, there have been continuing technological advances toward integrated services. However, the situation becomes more complicated in the presence of MH (Mobile Host) movements. Since the interface of a wireless connection to its backbone changes frequently because of hand-over events in a micro/pico-cellular network, we need an efficient handover protocol to keep a seamless connection irrespective of handovers. We will describe this matter in chapter 3 in detail.
2.3.3.2 Reservation protocol supporting mobility

In a cellular data network, the path of a connection changes frequently due to the handovers, so a reservation protocol should manage the mobility. In the point of view, RSVP designed for a fixed network is inadequate to accommodate the mobile hosts, which frequently change their attachment points to the fixed network. MRSVP [87] is an extension of RSVP to accommodate host mobility in a mobile network, introducing several new features. A noticeable new feature is the concept of active and passive reservations. While an active reservation is the real reservation that is being used by a mobile host at the present, a passive reservation is a virtual reservation allocated in the surrounding cells. MRSVP provides two reservation services: mobility independent service and mobility dependent service. The mobility independent service guarantees a reservation path regardless of handovers, allocating active and passive reservation to a mobile host. When a mobile host moves to one of the surrounding cell, the passive reservation converts into active reservation. In contrast, the mobile dependent service allows a mobile host to have only an active reservation. In this case, the mobile host should obtain a new active reservation whenever it moves to another cell. The information of active and passive reservations is carried by the RSVP messages, but they are opaque to RSVP.

MRSVP uses proxy agents to make reservations along the paths. To obtain the IP address of the proxy agent, a mobile host uses a proxy discovery protocol [87]. There are two types of proxy agents: remote and local. The remote proxy agents make passive reservations on behalf of the mobile host, while the local proxy agent of a mobile host acts as a normal router for the mobile host and sets up an active reservation. MRSVP has four new messages to handle mobility. These messages are handled only by the proxy agents while other fixed nodes of a path need not to be aware of these messages. The messages are tunneled using RSVP encapsulation [87].

2.3.3.3 Addressing

In an Internet-based mobile system with traditional IP, mobile hosts cannot access the IP network outside of home region because each mobile node is always identified by its home address regardless of its current point of attachment to the Internet. Several proposals [48], [74], [90], [65] have been made for supporting host mobility within Internet environment. The fundamental concepts of each proposal are similar. They separate the dual nature of an IP address into the home address representing the original IP address of a mobile host and the foreign address representing the new address allocated in the foreign area. The major differences are the ways of propagating the location information to trace the mobile host and forwarding the packets to the new location of the mobile host after a handover. Although much work has been done in
this area, the routing optimisation problem still remains. Some research [22], [50] resolves the problem by tunneling between the source and the destination using encapsulation. However this method does not fit standard networking models and requires large storage for caching the redirection list. Meanwhile, Mobile IP [74] has been suggested as a standard for the mobile IP network by the IETF and many companies are backing it.

2.3.3.4 Improving End-to-End performance

The reliable transport protocols such as TCP have been tuned for networks composed of wired links and stationary hosts. They interpret unexpected increases in delay as packet losses caused by congestion. In the response to perceived packet loss, TCP aggressively slows transmissions to allow the network to recover. This congestion control policy has been proven beneficial in improving the overall performance of networks like the Internet. In a wireless network, however, frequent handovers or transmission errors in a wireless link disconnect the communication link for a moment. It may falsely trigger congestion control mechanism of TCP, which result in degraded end-to-end performance. Recently, several reliable transport-layer protocols for wireless networks have been proposed [98], [18], [7], [5]. The Indirect-TCP protocol [5] splits a TCP connection into two separate connections: a wireless and a wired part. The advantage is to achieve the separation of the flow control and congestion control of a wireless link from those of a fixed network. While it results in the sender keeping sending data at good bandwidth, there are some drawbacks such as semantics, application relinking and software overhead. The fast-retransmit approach [18] addresses the issue of TCP performance when communication resumes after a handover. This approach reduces unacceptably long pauses during a handover using end-to-end fast retransmission, but it addresses only handover problems of TCP connections without considering the error characteristics of the wireless link. As another approach, the caching method [7] is proposed. The bottom line of the idea is to cache packets toward a mobile host at the base station and to perform local retransmissions across the wireless link. This protocol needs the network-layered software at the base station to be modified. The performance is improved up to 20 times over normal TCP/IP for data transfer, but it requires large caching storage.

2.3.3.5 Call control issues

In contrast to fixed networks, where the user-network-interface remains unchanged throughout a connection lifetime, a cellular data network allocates resources to a wireless connection whenever a mobile host changes its cell. Congestion may be encountered after a handover, and the event results in either the termination of the connection, large delays, and/or
packet loss. Resource allocation to wireless connections is inherently different from that of wired connections. Thus, in the new paradigm of cellular networks, call control functions are required to keep seamless connections. One solution [70] was suggested to treat new calls and handover calls differently by giving higher service priority to the handover calls. This solution lowers the congestion probability of handover calls, but uses only local state information for accepting a new call. This single-cell approach has the advantage of simplicity and efficient use of network resources. It is suitable for situations where the cell size is large and handovers occur infrequently such as a macro-cellular network.

In a micro-cellular network, a substantial number of call processing and control functions must be invoked due to frequent handovers. If such control functions were executed in a centralised fashion, the call processing of handover events would cause congestion at the central server. Acampora [1], [2] and Levine [61] propose distributed (multiple-cell based) control methodologies to guarantee QoS for high-speed micro-cellular networks based on a hierarchical grouping of backbone and wireless network resources. To achieve the distributed control, Acampora proposes a virtual connection tree scheme for a wireless ATM. In his distributed approach, a mobile connection consists of a multicast tree which includes not only the base station associated with the mobile host but also other base stations in its vicinity. The admission control exploits the information of resource availability in the multicast tree. Levine suggests the shadow cluster concept to admit only those calls that can be supported adequately. It can be viewed as a message system where a mobile host informs the base stations in the neighbourhood about its requirements, position and movement parameters so that the base stations can project future demands.

The multiple-cell based approach is suitable for mobility support in the environment where a cell size is small, and handovers occur frequently such as micro/pico-cellular networks. The drawbacks of this scheme are the increase in call set-up latency and higher call blocking probability. Lee [59] suggests a hybrid scheme of the single-cell-based and multiple-cell-based approaches. The hybrid approach could achieve better balance between handover performance and network efficiency, but the drawback is that control traffic and protocol processing overhead are relatively high and handover control would fail if a mobile host moves to a cell before the branch connections are properly set up.

2.4 Research Motivation and Scope

As multimedia applications such as a multimedia conference, Internet phone service and WWW (World Wide Web) have become widespread, supporting integrated traffic composed of
those applications in a cellular data network has been raised as a research topic. Since a wireless link has usually lower bandwidth compared to a wired network, the wireless link is likely to be the most congested part over a connection. Consequently, the wireless part of a path can severely affect the end-to-end QoS of integrated traffic, and supporting the QoS in a wireless link is critical. As introduced earlier in this Chapter, various cellular data networks have been developed and are being researched to support the integrated service with guaranteed QoS. While a circuit switch network can guarantee a rigid service, it hardly supports diverse QoS required by various traffic as well as it may waste a wireless channel that is valuable in a scant resource environment when a connection is inactive. Meanwhile, though a packet switched network can utilise a wireless channel efficiently and support integrated services, it cannot guarantee QoS without QoS control techniques. Therefore, it is important to develop such QoS control techniques to provide multimedia applications with guaranteed QoS since most of those applications can operate properly only under a guaranteed QoS commitment.

There has been no lack of research on a set of techniques to support guaranteed QoS for integrated traffic in a cellular packet switched network. As introduced above, those are reservation protocols such as MRSVP [87], admission controls [59], [2], [70], [61], addressing with mobility [1], [90], [74], improving End-to-End performance [98], [18], [7], [5] and adaptive applications [14], [23], [12]. However, we discovered there is little research on resource distribution at packet level, while it has been actively researched for a wired network such as weight based fair scheduling schemes [71], [30]. The resource distribution scheme designed for a wired network cannot be directly applied to a cellular packet switched network due to the different characteristics such as scant bandwidth and mobility. In addition, a cellular packet switched network needs an additional technique for the resource distribution, a multiple access protocol. Therefore, the resource distribution scheme in a cellular packet switched network should take into account those factors. In this dissertation, we will tackle the resource distribution problem at packet level to provide integrated traffic with guaranteed QoS in a cellular packet switched network. The core components of the research will be a scheduling algorithm for up/down link traffic, multiple access protocol for uplink sharing, the handover protocol and its effect regarding the resource distribution.

**Scheduling Discipline:** Bandwidth discrepancy between wired and wireless links brings about a bottleneck in a base station. Accordingly, a cellular packet switched network should settle the disproportion of interface for supporting QoS guaranteed integrated services by controlling the congestion and properly allocating bandwidth
to flows as they require. For the purpose, a scheduling discipline should be adopted as an up/downlink scheduler of a base station, which schedule packets to provide a flow with the committed service regardless of congestion. Most published scheduling discipline is designed for a high-speed network. They did not focus on the cellular network characteristics such as varying bandwidth and flow handovers. Especially, a wireless link is relatively slow, so fairness is more important than low computation burden which is necessary in a high-speed network.

**Multiple Access Protocol:** In a cellular packet switched network, a multiple access protocol plays a significant role to provide integrated services. The protocol should guide each mobile entity’s behavior for channel contention and control the number of admitted flows so that a flow can access channels as committed regardless of other traffic load, performing high channel utilisation. It is different from that of a cellular circuit switched network, which adopts an assigned channel access scheme that guarantees an allocated channel once a flow is admitted. Many multiple access protocols have been researched, but most of them [88], [42], [69], [64] focused on the high channel utilisation and did not deal with dynamic bandwidth distribution according to each flow’s requirement. Some papers [97], [13], [54], [4], [83] tackled the dynamic bandwidth distribution, but have weak points on guaranteed QoS.

**Handover effects on QoS:** An efficient handover procedure is necessary to support seamless connections in a cellular packet switched network. During a handover of a flow, a base station should forward buffered packets of the flow to the new base station, but it may cause misordered packet delivery if the handover protocol does not consider the packet forwarding. In addition, a handover may affect the fair service among flows in the old base station unless the packet scheduler does not take into account the packet forwarding. We need to analyze the handover effects on a handover protocol and packet scheduler. Most handover protocols and packet scheduling disciplines did not consider it.

To sum up, a base station can be regarded as a bridge to connect between a wired and a wireless world. A cellular network has different characteristics compared to a wired network, which cause a base station to experience a bottleneck from the wired network and mobile hosts to suffer from sharing a low quality wireless link. In addition, handovers introduce another trouble. To solve those problems, we, therefore, need some schemes for fair sharing of a wireless link and
a resource distribution scheme to support a guaranteed QoS to integrated traffic in a cellular packet switched network.
Chapter 3

Related Work

In this chapter, we present the related work in our research arena. We first review the scheduling disciplines for packet switching networks, showing the similarities and differences among them, and then discuss the requirement of service disciplines for a base station in a cellular packet switched network that supports guaranteed integrated services. Following that, we describe and discuss the multiple access protocols for packet switched wireless networks in relation to integrated services, emphasizing necessary functionality to support integrated services and approaches to improve performance. Finally, we introduce several handover protocols and discuss some problems that could occur when they are operating with a rate-based scheduling discipline.

3.1 Scheduling disciplines

Many packet scheduling disciplines [71], [30], [43], [39], [9], [85], [101], [55] for packet switches have been studied to provide a deterministic or statistic guaranteed QoS. In this section, we describe several scheduling algorithms and identify their advantages and disadvantages.

3.1.1 Fair Queuing

One of the first attempts was the Fair Queuing algorithm [29] introduced by Demers. The main purpose of the algorithm is to provide all flows with an equal allocation of bandwidth in the traditional data service architecture. It exploits a bit-based round-robin service discipline to distribute the bandwidth equally to all flows since a packet based round-robin discipline induces unequal bandwidth distribution because of the variation in packet sizes. To solve the unequal distribution problem, he introduces a time stamp $F_{i,a}$ calculated by bit-by-bit round-robin, where $F_{i,a} = P_{i,a} + \text{MAX}(F_{i-1,a}, R(t_{i,a}))$. In the equation, $\alpha$ denotes flow number, $i$ refers to the packet order of a flow, $P$ represents the packet size. $R(t)$ is defined as the number of rounds made in the round robin service discipline up to the time $t$. If the round-robin service is conducted in a bit-by-bit fashion, all flows can receive equal bandwidth. A practical packet-by-packet scheduling can be
easily derived from the bit-by-bit one, choosing the packet that has the smallest value of \( F_i^a \) as the next served packet.

The scheme can also embrace a priority policy, replacing \( F_i^a \) with \( B_i^a = P_i^a + \text{MAX}(F_{i-1}^a, R(t_{i-1}) - \delta) \), where \( \delta \) is a nonnegative parameter. In the new time stamp, the parameter \( \delta \) allows a flow that has consumed lower bandwidth than allocated to receive more prompt service. Demers also mentioned the possibility of generalizing the Fair Queuing algorithm in order to allocate bandwidth in proportion to the share rate of a flow. Clark [23] named it WFQ (Weighted Fair Queuing) and Parekh [71] formulated it in the context of a fluid-flow modeled network. The model provides a traffic stream with an effective throughput rate no worse than \( ur^a \Sigma p e^a \), where \( r^a \) represents the relative share of the link bandwidth for the traffic stream \( a \), the denominator is the overall share rates of currently active flows and \( u \) denotes the full link speed. The details of the model follow in the next two sections.

3.1.2 GPS (Generalised Processor Sharing)

Parekh suggested an ideal scheduling discipline [71] called GPS that provides individual flows with guaranteed bandwidth in a fluid-flow modeled packet network, where it is assumed that a packet is infinitely divisible – this is an imaginary network. The assumption enables more than one packet to be transmitted simultaneously over a communication link. The discipline can provide two desirable properties to realise guaranteed service: bounded end-to-end delay and guaranteed throughput. For the delay, it is demonstrated that GPS can guarantee an end-to-end delay bound for a flow, provided that its source traffic is constrained by the leaky bucket model [92]. With reference to the throughput, it can allocate fair link bandwidth to flows at the rate of weights associated with flows without any restriction of traffic source pattern.

Consider a work-conserving GPS server that operates at a fixed rate \( r \). A work-conserving server is never idle as long as there is at least a queued packet to send. Let \( J \) represent a set of flows served by the GPS and \( \phi_j, j \in J \) represents a positive service share that is allocated to the flow \( j \). Then the GPS server is formally defined for any flow \( i \) that is continuously backlogged in the interval \( (\tau, t] \) as

\[
\frac{W_i(\tau, t)}{W_j(\tau, t)} \leq \frac{\phi_j}{\phi_i}, \quad j \in J
\]

where \( W_i(\tau, t) \) indicates the amount of traffic for flow \( i \) served in the interval \( (\tau, t] \). Let \( B(\tau, t) \) be the set of flows that are continuously backlogged in the interval \( (\tau, t] \). Summing over all flows \( j \), the service \( W_i(\tau, t) \) is given by
Thus, in any interval of time, a flow \( i \) can receive at least at a rate of

\[
    r_i = \frac{\phi_i}{\sum_{j \in B(\tau,t)} \phi_j} \cdot r(t - \tau)
\]

The commitments of the bounded end-to-end delay and guaranteed throughput are very appropriate for designing a guaranteed integrated services network. However, the assumption of GPS – infinitely divisible packet - is not realistic even though GPS features ideal characteristics for guaranteed services. In practical packet network systems, only a packet can be served at a time, and a whole packet must have been served before another packet is served because a packet cannot in general be divided and a link transmits only serialised bits. Since GPS cannot be implemented in practice due to the unrealistic assumption, Parekh, accordingly, proposed GPS as an ideal reference service discipline to measure the performance of any practical scheduling discipline served based on packet unit.

### 3.1.3 PGPS (Packetised GPS)

PGPS [71] is the packetised version of GPS that is designed to emulate the hypothetical GPS policy. Technically, it is identical to WFQ. Since PGPS is a packetised version, it cannot perform exactly the same way of GPS that serves packets in parallel. Instead, PGPS tries to maintain the same departure order of packets in the corresponding GPS. To compute the departure order of packets in the corresponding GPS server, PGPS tags a packet with its virtual finish time that reflects the past bandwidth usage of the flow. The virtual finish time is computed based on the virtual time \( V \) that evolves at the following rate:

\[
    \frac{dV(t)}{dt} = C \cdot \left( \sum_{j \in B(t)} \phi_j \right)^{-1}
\]

where \( C \) indicates output link capacity and other parameters follow the definitions in §3.1.2. Since the virtual time ticks at the summed share rates of backlogged flows, forcing synchronisation of the virtual start time of a packet with current virtual time prevents a flow that has stopped for a while from monopolising the output link for a short period after the pause. This does not allow a flow to save bandwidth or to obtain more prompt service after the flow has paused for a period. Without the virtual time, other flows may not receive any service due to the link monopoly by a flow with bandwidth saving, which is undesirable for guaranteed service.
When a PGPS server is ready to transmit a packet at the time $t$, it singles out the packet that would complete service first among all the packets queued in the corresponding GPS server at the time $t$, on the assumption that no additional packets arrive after the time $t$. Since PGPS transmits only a packet at a time, unfairness must occur between flows in the micro point of view – i.e. a flow must wait for its service while another flow’s packet is being transmitted even though both flows have the same share rate. Apart from the microscopic unfairness resulting from the packetised discipline, the assumption adopted in PGPS could allow another unfairness: i.e. a packet tagged with an earlier finish time may be scheduled later than one tagged with a later finish time if the former does not arrive before the latter starts to be sent. To avoid this problem, PGPS should be a non-work-conserving server\(^3\) which can stop serving to control traffic pattern distortions even if there are packets in the queue. However this is against the definition of GSP.

3.1.4 WF\(^2\)Q (Worst-case Fair Weighted Fair Queuing)

To alleviate the unfairness effect of the packet-by-packet fashion in WFQ, Bennett [9] proposed WF\(^2\)Q, which can emulate GPS with higher accuracy. Since WFQ considers only the virtual finish time of a packet when it chooses a packet for next service, the packet with smaller normalised service time - computed by $L_k/r_k$, where $L_k$ and $r_k$ are the packet size and the share rate of the flow $k$ - has higher possibility to be served earlier than others irrespective of the service order in GPS: i.e. the larger the normalised service time of a packet, the higher the delayed service possibility. In the WF\(^2\)Q system, the server selects a packet with the earliest virtual finish time from among the set of packets that would have started (and possibly finished) their service in the corresponding GPS system. The enforced strategy of packet selection minimises the discrepancy between GPS and its packetised version, so WF\(^2\)Q is more similar to GPS differing by no more than one maximum size packet. The only disadvantage is higher burden of computation.

3.1.5 SCFQ (Self Clocked Fair Queuing)

Although WFQ and its derivatives have the best performance among packet-by-packet scheduling disciplines both in terms of latency and fairness properties (demonstrated by Stiliadis [85]), they have higher computational complexity of $O(V)$, where $V$ represents the number of

\(^3\) This may waste bandwidth and is used where end-to-end delay bound is more important than average delay.
backlogged flows, compared with others [101], [55]. Every time-stamped discipline such as WFQ [30] or VirtualClock [101] has an inevitable computation burden for sorting the time-stamps. Apart from this, WFQ has further complexity associated with the evolving of virtual time, which depends on the rate of change of the number of the backlogged flows. WF²Q has even more complexity owing to more stringent rule to select served packets. These algorithms with high computational burden are difficult to be implemented in a high-speed network such as a Giga bit network. Thus, Golestani [39] proposed a simplified version (SCFQ) of WFQ to reduce the computational complexity. Instead of computing the virtual time, SCFQ simply estimates the virtual time with the virtual finish time of the current served packet. Although the estimation of the virtual time lessens the computational burden, the delay bound for a flow is larger than that of WFQ. The delay bound also increases in proportion to the number of flows [40]. Thus SCFQ is not suitable for networks that have to manage a large number of connections.

3.1.6 SFQ (Start-time Fair Queuing)

Recent work by Goyal [43] has proposed SFQ that is another packetised version of the GPS server. This is similar to the SCFQ except that it schedules packets based on their virtual start time instead of the virtual finish time. He emphasises that the virtual start time method enables a low-throughput flow - such as a user interface flow for an interactive application where fast response is required - to experience less delay than a high-throughput flow. Provided that all applications send the same size packets, this is true since in the virtual finish time based scheme a lower share rate flow has larger normalised service time than a higher share rate flow. However, a user interface flow usually exploits a small packet to send a signal, which results in the flow having a small-normalised service time. So it can naturally obtain prompt service even in the virtual finish time based queuing. Conversely, the start time based queuing has a disadvantage. Assuming that all flows have the same share rates, the flow with larger packets obtains more prompt service than other flows. This is not correct when considering the general fairness concept.

Goyal [43] analysed its performance bounds in a variable-rate link, showing that the performance bounds vary as the available bandwidth fluctuates, i.e. SFQ can guarantee fixed bandwidth only in a fixed bandwidth link. Therefore, the discipline is not applicable to the network that has a variable-rate link and should support a constant bit rate or a guaranteed minimum bit rate service.

Goyal also reports that unfairness occurs when WFQ is adopted in a variable-rate link and proved that adoption of the self-clocked virtual time - using the virtual finish time of current
served packets as the current virtual time as in SCFQ - can remedy the unfairness problem. In WFQ, the virtual time ticks faster than its correct pace whenever the outgoing link rate is reduced. The time lead results in unfair services. Since SCFQ and SFQ adopt the self-clocked virtual time scheme, they do not have the unfairness in a variable-rate link. However, as introduced in the previous section, the simplified virtual time method has the disadvantage that it increases delay bounds in proportion to the number of flows.

3.1.7 Virtual Clock

The VirtualClock algorithm [101] is similar to the PGPS in the sense that a flow can obtain an arbitrary fraction of total bandwidth, but the VirtualClock algorithm does not exploit the "virtual time" unlike PGPS. Instead of the virtual time, it uses "real time" to compute a time stamp \( auxVC \) (equivalent of virtual finish time in PGPS). The algorithm is follows:

1) Upon receiving the first data packet from \( flow_i \),
\[
VirtualClock_i = auxVC_i = \text{realtime};
\]

2) Upon receiving each subsequent packet from \( flow_i \),
\[
VirtualClock_i = (VirtualClock_i + Vtick_i);
\]
\[
auxVC_i = (\max(\text{realtime}, auxVC_i) + Vtick_i);
\]
\[
Vtick_i = \frac{1}{AR_i} \text{ (packet/sec); where } AR_i \text{ indicates the average rate of } flow_i
\]

Stamp the packet with the \( auxVC_i \) value;

3) Packets are queued up and served in increasing order of the stamps.

The use of real time causes severe unfairness. For instance, assume that a bursty flow starts to send packets after a pause, and then the flow's time stamp starts to increase from the current "real time" of the starting moment. Meanwhile, since the bursty flow has apparently been inactive before the moment even though it has a link share, the time stamp of the current served packet has higher value than real time. As the VirtualClock serves in increasing order of time stamps, the new backlogged flow receives service exclusively until the real time catches up the time stamp of the last served packet. To sum up, a pause period allows a flow to save bandwidth, and it causes unfair bandwidth distribution. According to the fairness definition of Golestani [39] (in his fairness definition, the smaller fairness value, the fairer service), the fairness value of VirtualClock is infinite, while PGPS and SCFQ has a bounded fairness.

To deal with the unfairness problem, the VirtualClock Algorithm monitors individual flows against its reservation and uses it for feedback control. A warning message is sent to the flow source if the flow’s sending rate is higher than the allocated one by a threshold. But all that the fairness control by the traffic monitoring/warning can do is to warn the greedy source to reduce
the source rate. Even if a further control action – for instance stop the service or reduce the allocated rate – is adopted, it cannot adapt swiftly to bursty flows nor solve the unbounded fairness.

### 3.1.8 CBQ (Class Based Queuing)

The rate-based disciplines, including the methods explained above, distribute the bandwidth to flows in proportion to the relative fraction of total bandwidth at all times. Assume that the flows to be served form a tree structure according to multiple agencies, protocol families and/or traffic types and they have their hierarchical share rates where the share rate of a child node is limited by the share rate of its parent node. In that case, the simple rate-based scheduler cannot support the hierarchical share rate service. To endow a scheduler with the hierarchical services, hierarchical queuing algorithms have been studied. Floyd [33] developed a class based hierarchical link sharing mechanism named CBQ, which enables a flow to hold a hierarchical share of the output bandwidth.

To achieve the hierarchical share rate service, she developed a rule-based strategy rather than a mathematical method. The strategy has a simple sharing principle: a class can exploit more bandwidth than its hierarchical share if it has a not-overlimit ancestor at level \( i \) and there are no unsatisfied classes at levels lower than \( i \). Under this guideline, a sub-tree can exist independently of other siblings so long as the top of the sub-tree is not over its allocated share rate. To schedule packets, two schedulers operate in concert: a general scheduler and a link-sharing scheduler. The role of link-sharing scheduler is to control the packet scheduling of the regulated classes that are over their service limit unsatisfying more than a class, while the general scheduler handles the other classes. Floyd does not confine the schedulers to specific ones. As a result, any kind of scheduler can be adopted as long as it can conform to the sharing policy. The CBQ can be regarded as a hierarchical extension of the VirtualClock or WFQ, given that the VirtualClock or WFQ is designated as the general scheduler.

For implementing the algorithm, Floyd [34] proposes WRR (Weighted Round-Robin) scheduling method as the general scheduler. The WRR allows a class to exploit the link bandwidth in proportion to its share rate. She also implemented a link-sharing scheduler that sets a time-to-send field when a class changes into a regulated one. The general scheduler excludes the regulated class from its round-robin service until the indicated time in time-to-send field of the class. Thus, the service pause by the time-to-send field reduces the service rate for the regulated class down to its allocated rate.
Floyd suggested an estimator that determines the usage status using an exponential weighted moving average. The estimator measures the service status of a class using the average discrepancy, \( \text{avg} \) between the actual inter-departure time and the allocated inter-departure time, which is derived from the following equations.

\[
\begin{align*}
    f(s,b) &= s / b \\
    \text{diff} &= t - f(s,b) \\
    \text{avg} &\leftarrow (1 - w) \ast \text{avg} + w \ast \text{diff}
\end{align*}
\]

where \( t \) indicates a measured inter-departure time, \( s \) denotes the size of the recently-transmitted packets in bytes, \( b \) represents the link-sharing bandwidth allocated to a class, \( w \) refers the time constant of the estimator that determines the past \( \text{avg} \) effects. If the estimator changes a class status to a regulated one, the link-sharing scheduler sets the time-to-send field of the class to \( s / b \) seconds ahead of the current time to reduce the service rate of the class.

Though CBQ features the hierarchical bandwidth distribution, it has also several disadvantages. Firstly, it cannot rapidly adapt to varying traffic patterns, allowing a flow to exceed its allocation for a short period. Secondly, it is not clear that it can guarantee the delay bound as well as the fairness.

![Graphical comparison between CBQ and Rate-based scheduling](image-url)

**Figure 3.1** Graphical comparison between CBQ and Rate-based scheduling
3.1.9 Comparisons of CBQ and Rate-based Scheduling

The different functions of the two algorithms can be easily seen in Figure 3.1. As can be seen, CBQ can have a tree structure and allocate hierarchical share rates. Suppose that the 4 flows at the bottom possess a 25% share each among the total bandwidth and in the case of CBQ there are two agencies, A and B that have a 50% share each. Figure 3.1 (a) and (b) show the bandwidth consumption status in CBQ and rate-based scheduler when \( f_2 \) is utilizing only 10% bandwidth, while the others are sending data at higher rate than their allocated shares. In the rate-based scheduling, the unused bandwidth of \( f_2 \) is equally shared by the other 3 flows, but CBQ allows only \( f_1 \) to utilise it because B class can utilise only 50% bandwidth upon over-loaded period. This example describes well the difference between CBQ and rate-based scheduling.

3.1.10 Summary

A scheduling discipline is one of the important components in supporting guaranteed QoS. All disciplines explained so far assumed that an outgoing link rate is fixed. The assumption confines their applicable areas to static communication links. As explained in the introduction, however in reality, various reasons induce an outgoing link rate to be variable especially in a cellular packet switched network. However, most scheduling schemes are not designed for variable-rate links, so they do not have the desired properties to support a guaranteed QoS for a varying capacity network. Although SFQ can schedule packets in a variable-rate link, it cannot guarantee a fixed QoS. To provide a flow that requires a guaranteed QoS regardless of the link property, we need a scheduler that can guarantee constant bit rates to intolerant flows in a variable-rate link.

3.2 Multiple access protocols

Multiple access protocols enable mobile users to share the same communication channels. Many multiple access protocols have been proposed and analyzed to deal with the shared use of a transmission link by multiple users. The protocols are grouped into four general classes: fixed-assignment, demand assignment, random access and hybrid access. In this section, we focus on the multiple access protocols for packet switched networks rather than those for circuit switched networks. Usually, TDMA and FDMA have been used for a circuit switched network, and recently CDMA has become widespread because of its greater user capacity than TDMA/FDMA. Meanwhile, in packet switched networks many different kinds of protocols have been developed. They are usually based on the Aloha protocol. In the ALOHA-type protocol, when a conflict occurs during a contention phase, all packets involved in the conflict must be retransmitted. So it
is important to reduce conflicts to improve channel efficiency. In addition, a multiple access protocol for an integrated services packet network should have the capability to distribute channels to flows according to the required QoS of each flow, as well as maintaining the high channel efficiency. We present here some multiple access protocols that can handle flexible resource assignment in a cellular packet switched network and discuss them.

3.2.1 PRMA (Packet Reservation Multiple Access)

Fixed-assignment techniques such as traditional TDMA and FDMA, incorporating permanent sub-channel assignment for individual users, perform well with constant bit rate traffic. In the presence of a discontinuous or bursty data stream such as voice or data connection, however, they do not fully utilise wireless channels. PRMA [42] has been proposed to allow integrated traffic flows to share the same wireless channel. PRMA, which is a close relative of reservation ALOHA [88], can be viewed as a combination of TDMA and slotted ALOHA [88]. Unlike reservation ALOHA, PRMA distinguishes two types of information packets: periodic and random. A periodic connection can obtain a guaranteed slot in every frame after it has achieved a successful contention. The guaranteed reserved slots enable a periodic connection to have an exclusive assignment like TDMA, but the connection loses the guaranteed slots when it does not send a packet during its assigned slot. A random connection should contend for a slot whenever it has a packet to send. Due to the dynamic slot allocation, silence periods of a discontinuous connection can be utilised by the other connections. Thus it improves the channel utilisation. The operation of the scheme is similar to TDMA when it runs with only steady streams and slotted ALOHA with only bursty streams.

3.2.2 Improved versions of PRMA

The important performance measure of PRMA is the packet drop probability since a delay intolerant periodic source such as speech should contend to obtain a slot reservation at the beginning of every talkspurt. PRMA may not provide an acceptable packet drop rate due to the unstable characteristic of ALOHA contention. Integrated FRMA [69] has been proposed to provide a speech connection with a better packet drop probability, in which a base station classifies every slot as either reserved, voice contention or data contention at the beginning of every frame. It prevents voice and data connections from contending in the same slot and also controls the partition rate between voice and data, maximizing bandwidth allocated to data connections subject to keeping the voice packet drop rate to at most 1%.
Another problem of PRMA is that it allows the available slots for reservation to shrink down to zero under high load conditions. Thus, at high load, the success rate of contention decreases, which results in access delay increasing. Mitrou [64] suggested a reservation multiple access protocol, which reserves a minimum number of slots for reservation to keep the access delay within set limits. It also adopts an adaptive retransmission probability for stable operation. In his system, the retransmission probability for contention decreases as the system load increases.

3.2.3 I-ISMA (Idle Signal Multiple Access for Integrated services)

The reservation multiple access protocols using random access techniques [42], [69], [64] are not efficient to provide diverse sources with their required QoS. Demand assignment schemes with centralised control, like polling, can more efficiently perform statistical multiplexing for appropriate resource distribution. In the demand assignment scheme, a base station can control a slot allocation sequence with the help of a central scheduler. Wu [97] proposed I-ISMA that is the modified version of ISMA to provide the integration of voice and data traffic by borrowing the PRMA concept. Instead of the frame and slot structure, I-ISMA introduces very short idle signals and polling signals. A base station periodically sends a very short idle signal to declare that the shared channel is idle. The first packets of talkspurts and all data packets are conveyed by ALOHA contention scheme whenever an idle signal is broadcast. A base station polls the following packets of a talkspurt after the successful transmission of the first packet of the talkspurt. This scheme shows higher capacity than that of PRMA in low load status. It is because idle signals in low load enable mobile hosts to access the channel swiftly. It can also support the network that allows various packet sizes. However, the benefit gradually disappears as the signalling distance and the number of talkspurts increases. Especially, longer signalling distance makes I-ISMA perform worse than PRMA even in low load. Therefore, it is a suitable model for micro- or pico-cellular networks. It has also a decreasing capacity problem like PRMA as load increases, which are caused by the instability of slotted ALOHA contention.

3.2.4 C-PRMA (Centralised Packet Reservation Multiple Access)

Recently, another polling scheme called C-PRMA, has been presented by Bianchi [13]. In the scheme, all uplink packets are conveyed by guidance of the polling signal. A centralised scheduler installed in a base station decides the polling signal sequence. Since it is a slot-based protocol, the polling signals are broadcast at the beginning of each downlink slot, which synchronises with uplink slots. An out-Slot scheme is adopted, where a reservation request is transmitted during a separate reservation slot subdivided into minislots. The efficient use of the
reservation channel enables C-PRMA to have higher performance gain than PRMA. C-PRMA also introduces a dynamic slot allocation scheme by the centralised scheduling function PR (Polling Register) that handles various sources with different transmission rates, priorities, delays and packet loss requirements. However, the scheduling by PR has several limitations. Since PR slot saturation causes packets to be dropped, PR should have a certain amount of empty slots to satisfy the minimum required packet drop rate. However, he didn’t suggest any analytical tool to compute the proper empty slot number that varies depending on the QoS parameters of connections. Therefore, uncontrollable packet drop is inevitable to all connections at high load or when bandwidth is reduced by physical impediments. This restriction makes C-PRMA particularly unsuitable for data sources that cannot endure the packet drop.

3.2.5 C-CRA (Centralised-Collision Resolution Access)

The protocols [42], [69], [64], [97], [13] that use slotted ALOHA for contending channels are efficient for multiplexing varying numbers of bursty sources, but stable operation cannot be guaranteed unless the contention rate - the number of contending connections / free slots – is low enough. C-PRMA is still unstable if the number of minislots is not large enough compared with the number of mobile hosts. To alleviate the instability problem of slotted ALOHA, C-CRA [4] was proposed to improve the stability using the two-cell algorithm [73], where the contending probability is adaptable. This scheme is especially suitable where the number of mobile hosts is not limited. However, when the maximum number of mobile hosts can be estimated, C-CRA has similar performance to C-PRMA provided that minislot number in C-PRMA is selected large enough based on the maximum mobile host numbers.

3.2.6 LAP (Low-power Access Protocol)

Sivalingam [83] suggested a contention-free reservation protocol, LAP, which is also a centralised multiple access protocol based on scheduling strategies. The common consensus of the protocols [13] and [83] is the introduction of scheduling discipline at a base station to deal with QoS issues. However, LAP is different from C-PRMA in terms of reservation request scheme. Slotted Aloha is still used when a base station admits a new mobile host, but after a successful call set-up, the base station assigns one mini-slot during a reservation phase to each registered mobile host so that the mobile host can communicate with the base station through the assigned minislot. Therefore, a mobile host can obtain reservation slots at the beginning of talkspurt without contention. Since the scheme does not use contention algorithm to reserve slots, it can save power and channel that are wasted on conflicting in contention modes. However, as
the number of mobile hosts increases, the fixed allocation scheme for request channels is no longer practical. It is only suitable for a small number of mobile hosts due to the mini-slot size for reservation requests. Sivalingam also did not propose a specific scheduling algorithm for the protocol, so the performance of the protocol is not verified under a specific scheduling algorithm.

3.2.7 Packet CDMA (Code Division Multiple Access)

The use of CDMA technique for multiple access communications was first considered for satellite applications, and it has become widespread since the Qualcomm CDMA cellular system has been adopted as IS-95 standard by the US TIA (Telecommunication Industry Association). Properly augmented and power-controlled CDMA promises a quantum increase compared to the current cellular capacity – such as TDMA or FDMA - due to its inherent immunity to co-channel interference and multipath propagation [38]. It also offers greater flexibility with regard to cell site selection and frequency coordination because it reuses the entire spectrum in all cells. Unlike TDMA and FDMA whose capacity is mainly bandwidth-limited, CDMA is only interference-limited. However, the CDMA base station should have the same number of receivers as MHs to demodulate received signals which have different codes. So, if the total number of potential data terminals in the network is much larger than the number of active terminals at any given time, considerable complexity into the design of a CDMA multiple access system could be introduced.

In a cellular packet network, CDMA has another disadvantage. A major weakness of CMDA for integrated services is that, for a given system bandwidth, spectrum spreading limits the peak user data rate to a relatively low value [78]. Secondly, arbitrary interference patterns resulting from random channel access of packets in a packet CDMA increase the variance of interference, while the variance in a voice network keeps low due to the continuous streams of all connections. The high variance leads to lower capacity because of higher outage probability of the system [17]. Thirdly, in order to provide a different spreading sequence to each user in a dynamically changing user set, the CDMA system must provide a separate ALOHA request channel. So the request channel introduces a delay and an overhead in the call setup process. For a network serving only voice traffic it does not appear that this overhead and delay is a serious problem. But in the network that includes a significant amount of data traffic with less regularity than voice traffic, both the overhead and the delay could limit the network flexibility and the ability of the network to adapt to a more general traffic mix.

Zorzi [103] compared the system performance between the slotted Aloha packet system and CDMA system for packet voice transmission and he shows that the capacity of Slotted Aloha is
comparable with or better than the capacity of CDMA if an efficient packet acknowledgment transmission is performed.

3.2.8 Summary

A multiple access protocol for QoS guaranteed integrated services should support flexible bandwidth allocation to random traffic connections to provide them with their required QoS. C-PRMA suggests the flexible bandwidth allocation scheme using PR that allocates bandwidth to periodic and random traffic in the same way. However, periodic and random traffic have different characteristics: for instance, periodic can accept packet drop but is delay intolerant, whereas random traffic can tolerate delay, but packets should not be lost. Due to these different characteristics, random traffic should be handled in different and de-coupled way from periodic traffic. In conclusion, a multiple access protocol should have a flexible bandwidth allocation scheme for random traffic and lossless packet delivery function.

Meanwhile, all multiple access protocols [97], [13], [54], [4], [83] explained above are designed for uplink channel only. If we consider the uplink/downlink pair servicing a mobile end system, an uplink/downlink-partitioning scheme, where an uplink and downlink are separated, does not allow empty slots of one link to be exploited by the other link. Thus, it has disadvantages in term of channel utilisation and performance improvement, especially in an integrated services network. The centralised multiple access protocol can accommodate the uplink/downlink sharing via a scheduler located in a base station. However, most centralised schemes [97], [13], [54], [4] did not suggest the sharing scheme. Although LAP [83] allows downlink and uplink to share a common channel, it does not suggest any scheduling scheme to perform dynamic slot allocation between uplink and downlink. Therefore, we need an uplink/downlink sharing scheme that performs better channel utilisation than partitioning scheme in an integrated service network that has asymmetric traffic load between uplink and downlink in a macro or micro level period.

3.3 Handover protocols

A handover is regarded as one of the important operations in a cellular network since it allows an application to maintain its flow without interruption. A handover protocol enables flows to migrate seamlessly from one cell to another cell during a handover. To support a seamless handover in a cellular data network, there has been much research effort in different ways. Bakre [6] suggested a handover method operating at a transport layer based on Indirect TCP (I-TCP), but the proposal was a dedicated handover process to TCP connections. On the other hand, many handover protocols [49], [19], [24], [74], which are implemented at the network
layer level to support transparent IP address irrespective of the mobile host’s location, have been suggested.

3.3.1 IP based handover protocol

Ioannidis [49] suggested one of the first IP based handover protocols which can hide mobility from applications and the transport layer, providing continuous Internet access to mobile hosts in a campus area network. According to his model, a mobile host can always retain its home IP address even though it migrates to foreign subnets. In the context of Internet, an IP packet is routed toward its destination subnet address, so a mobile host, which retains its home IP subnet address, cannot receive any packet in the foreign subnet which address is different from the home address of the mobile host. To tackle this problem in a campus size network, he lets base stations manage an additional routing table for mobile hosts and perform special functions to route packets properly.

When a stationary host sends packets to a mobile host, the base station attached in the same subnet of the sender queries the other base stations to locate the foreign base station servicing the mobile host. After obtaining the address of the foreign base station, it sends encapsulated packets, which contain the foreign address as a destination address, to the foreign base station. Upon arriving at the foreign base station, the encapsulated packets are unpacked, and the original datagram is delivered to the mobile host by the foreign base station. The packet forwarding by encapsulated technique is known as IP tunneling.

Let us look at what is happening in mobile hosts. A base station periodically broadcasts beacon messages containing the IP address of the base station. If a mobile host receives a stronger beacon signal than the current one, it responds with a greeting message containing the home address of the mobile host and the previous base station. When the base station receives the greeting message, it acknowledges the greeting message. After that, it adds the mobile host address into its routing table entry and notifies the previous base station of the handover. Upon receipt of the acknowledgement message, the mobile host changes its access point to the new base station. Meanwhile the previous base station, which receives the notifying message, should send a redirect message to the counterpart base station, which has sent packets to the mobile host, such that the counterpart base station changes the forwarding address to the new base station.

3.3.2 Caceres's handover protocol

Recently, Caceres [19] proposed an IP layer based protocol with a hierarchical mobility management. It extends Mobile IP [74] to improve the inefficiency of home agent concept in
which a home agent should be notified of every subnet change of handovered mobile hosts. The handovers in his scheme have a three level hierarchy: between administrative domains, within the same administrative domain and within the same subnet. The global handovers across administrative domains exploits the Mobile IP without modification. For the second level handover that occurs within the same administrative domain, domain and subnet foreign agents are adopted to remove redundancy of Mobile IP. The domain foreign agent maintains per-mobile host routing entries so that a mobile host could remain within a domain with the same care-of-address.

For the lowest level handover that occurs in the same subnet, he proposed a local handover protocol similar to Ioannidi’s one, but a retransmission packet scheme and a redirect message notification method are newly introduced. After a handover, an old base station forwards few last packets in the retransmission buffer, which are already transmitted to the mobile host by the old base station, to the new base station so that the new base station transmits them to the mobile host again. It reduces packet loss during handover especially between non-overlapped cells, but it may cause duplicated packet delivery. This idea is similar to Balakrishman’s [7] method in which he uses it to improve TCP performance.

Another difference is the way of informing redirect addresses. A new base station broadcasts a redirect message on the wired link to notify a handover such that any related nodes including the router on the same subnet can change their destination to the new base station. He experimented single flow handover case with FCFS queue, but he did not consider the case of multiple flows handovers - a mobile host handover with multiple flows or simultaneous handovers of multiple mobile hosts. In those case, the static retransmission buffer scheme may not be effective because several flows share one retransmission buffer.

### 3.3.3 Summary

A handover protocol can be implemented in any layer, from the link layer to the transport layer. Most handover protocols have dealt with the network layer problems, especially transparent IP addressing problem. However, there is another important issue in handover. It is to design lossless handover as Caceres’s handover protocol. During a handover, packets buffered in the old base station are lost if they are not forwarded to new base station. If the packet loss occurs, it seriously affects data connections that are intolerant to packet loss. It is certain that the packet loss can be recovered by an upper layer protocol such as TCP. As explained in section §2.3.3.4, however the packet loss during handovers may be misunderstood as congestion by TCP, then TCP may falsely trigger the congestion control mechanism, which result in degraded end-to-end
performance. Since handovers occur frequently in a micro/pico cellular network, the overall effect of the packet loss caused by handover is significant. Therefore, designing lossless handover is important, and handover protocols should embrace the packet forwarding function to realise the lossless handover.

There are some difficulties in designing a handover protocol with packet forwarding function. In the system that schedules packets via a rate-based scheduler to guarantee QoS, the handover protocol with packet forwarding function influences performance of the rate-based discipline due the forwarding packets - for instance, unfairness and out-of-order packet delivery. The phenomenon does not occur at the FCFS queuing system which most published protocols use as a packet scheduling discipline. However, if a base station adopts a rate-based scheduling discipline to ensure a committed QoS to flows, unfairness and out-of-order packet delivery may occur in those protocols. As a consequence, for a QoS supported cellular network, a new handover protocol that conforms consistently to rate-based scheduling discipline is needed.
Chapter 4

Services architecture

As indicated through Chapters 2 & 3, the wireless part among a whole network path severely affects the QoS of a flow due to the performance discrepancy between wired and wireless links. In such environments, it is important to allocate and distribute wireless bandwidth to flows according to their requirements so that an admitted flow can continuously receive acceptable QoS maximizing the utilization efficiency of a wireless link. Since uplink and downlink have different characteristics, many studies on uplink and downlink QoS have been achieved separately. It is certain that those uplink and downlink QoS cannot be compatible with each other. However, service consistency between both is necessary. For example, assume that there is a connection between MH (Mobile Host) 1 and MH 2. To provide the flow from MH 1 to MH 2 with a consistent service throughout the whole path, the uplink part of MH 1 and the downlink part of MH 2 should support the same QoS. Therefore, we need a new service architecture, which provides uplink and downlink traffic with consistent service.

In this Chapter, we propose a service architecture for a micro/pico cellular packet switched network, which can enable both uplink and downlink traffic to receive the same QoS. In the service architecture, a flow can choose its suitable class among 4 service classes and service parameters according to its traffic characteristic to receive service satisfying its requirements. In the following sections, we will analysis the characteristics of applications that our proposed service architecture should cover, and then based on the analysis we will define the service classes to accommodate the various applications’ requirements. Finally, we will propose our service architecture over which the service classes can be supported.

4.1 Application Analysis

In this section, we investigate what network elements affect the performance of the applications that decides the quality of the application’s output – we call this QoA henceforth, and how the way that an application responds to the various QoS provided by networks influences its QoA. Then, we categorize the applications into several groups according to its responding behaviour. The application groups will be referred to in designing the service classes for our proposed architecture.
4.1.1 Delays

The delays that occur in a base station consist of 3 parts: packet-processing delay, waiting delay and bit transmission delay. The packet processing delay represents the time gap between received time of the last bit of a packet and queuing time at an output queue. It depends on the complexity of the applied packet scheduling algorithm and hardware power of the base station. The waiting delay denotes the period during which a packet stays at the output queue. This delay is divided into two periods: a queuing delay and access delay. While the queuing delay is the period of time until reaching the head of the queue from the time the packet was queued, the access delay is the time from arriving at the head of the queue to leaving the queue. In a shared medium network, even though a packet reaches the head of the queue, it cannot be sent immediately. It must wait until a channel slot is given to it. A packet scheduling discipline, which has its peculiar policy regarding fairness and service allocation, determines the queuing delay of a packet. The access delay mainly varies according to the network load and channel-sharing scheme, so it can be reduced by an efficient channel-sharing scheme or congestion control. The bit transmission delay is the time to transmit the packet and depends on the link characteristic and propagation distance.

4.1.2 Delay-tolerant vs. Delay-intolerant

The delay occurring in a base station affects the QoA. We divide applications into two categories according to their delay characteristics: a delay-intolerant and a delay-tolerant application. We define a delay-intolerant application as the application that has time-dependent data and a fixed playout point that is the time when the arrived data is picked up from the buffer and actually used. By the definition, all packets of the delay-intolerant applications have a fixed lifetime. Due to the fixed lifetime only the packets that have arrived at the destination within their lifetimes can be utilised, Thus, the QoA of the delay-intolerant application can be measured by the packet drop rate that indicates how many packets have not arrived within their lifetimes. In the reverse way, a delay-tolerant application is defined as an application in which a packet does not have a fixed lifetime.

Note that even though an application has time-dependent data, if it has an adaptable or adjustable playout point, then it is a delay-tolerant application. For instance, an Internet phone application could be delay-tolerant or delay-intolerant according to its playout policy. If an Internet phone application has a fixed playout point to support the same response time as the telephone network supports, it is a delay-intolerant application. However, if an Internet phone application can adjust the playout point according to the network delay – in this case, the user
may experience delayed response or shorter silence periods between talkspurts, it is a delay-tolerant application. These kinds of delay-tolerant applications can set a playout point manually or automatically to play as many packets as possible, depending on the packet delay. They need sufficient buffers to even out these delay jitters.

A delay-tolerant application can adapt to the network delay since its packets have adaptable or unbounded lifetimes. However, it does not mean that all packets can be used at the destination because some may be dropped in a congested switch during transmission. If the application uses a reliable transport protocol such as TCP, the dropped packet can be recovered by means of the retransmission procedure, but otherwise the dropped packet cannot be used. Also the adaptable ability against network delay does not mean that the QoA maintains steady all the time irrespective of network delays. Apparently, network delay affects the QoA of a delay-tolerant application, and especially the waiting delay dominantly influences the QoA more than packet-processing delay and bit transmission delay since it is much larger and more changeable than the others. The reason which the waiting delay has higher varying length and relatively larger value is that it is highly dependent upon available bandwidth which varies according to the network loads. In contrast, packet-processing delay and bit transmission delay are not dependent on network loads.

In addition to the waiting delay, delay variation known as jitter should be considered for the playout applications to decide a playout point to ensure that all arrived packets can be used in reconstructing the original signals. Therefore jitter, which occurs due to the available bandwidth fluctuation, also influences the QoA of some delay-tolerant applications.

In conclusion, the QoA of a delay-intolerant application is restricted by dropped packets, and the QoA of a delay-tolerant application is dependent on the waiting delay and jitter, which are affected by available bandwidth for the delay-tolerant applications.

### 4.1.3 Delay-tolerant application classification

In the analysis of the previous section, we identified that a delay-tolerant application is mainly affected by available bandwidth that produces varying delays. We here classify the delay-tolerant applications according to the adaptability against available bandwidth. This classification concept is derived from generalising the adaptive application model of Lee [59].

Lee suggested an adaptive application model for mobile data networks with varying bandwidth, illustrating the adaptability of an application by plotting its QoA as a function of bandwidth. The QoA function has an adaptable region of bandwidth referred to \((B_{min}, B_{max})\). The QoA function is on the increase during the adaptable region since a bandwidth increase does not
yield negative effects on the QoA. It keeps a constant value beyond $B_{max}$ while an application cannot obtain adequate QoA below $B_{min}$. The QoA functions have different shapes depending upon the application’s adaptability. Thus, he classified adaptive applications into four groups according to the shape of the QoA function in the adaptable region: none, slightly, moderately and highly adaptive applications. The derivative of the QoA function of a slightly adaptable application rises slowly during the early stage of the adaptable region, but it soars when approaching to the end point of the adaptable region: i.e. it responds slowly to the bandwidth increase. In contrast, that of a highly adaptive application rises rapidly at the beginning of the adaptable region: i.e. it responds quickly to the bandwidth increase.

However, the important factor in QoA is not in the adaptable property of an application, but in keeping an application to work all the time within acceptable QoA region irrespective of the supplied bandwidth fluctuation. Therefore, $B_{min}$ is more important index than adaptable ability to keep providing an application with QoS. If a network system could keep supplying bandwidth to an application above $B_{min}$, then the application can operate all the time without reducing its QoA down to zero. We, therefore, group the applications based on the $B_{min}$ characteristic of an application. In addition, we also consider $B_{max}$ in grouping the applications since $B_{max}$ is also an important index that shows a bandwidth response characteristic of an application. In conclusion, we classify delay-tolerant applications into five groups according to the bound of $B_{min}$ and $B_{max}$ as the following:

- **Adaptive application**: $Q(B_{min}) < Q(B_{max})$
  - BA (Bounded Adaptive) application: $B_{min} > 0, B_{max} < \infty$
  - UBA (Upper-Bounded Adaptive) application: $B_{min} = 0, B_{max} < \infty$
  - LBA (Lower-Bounded Adaptive) application: $B_{min} > 0, B_{max} = \infty$
  - NBA (Non-Bounded Adaptive) application: $B_{min} = 0, B_{max} = \infty$

- **NA (Non-Adaptive) application** : $Q(B) = \begin{cases} 0, & B < B_{min} \\ Const., & B \geq B_{min} \end{cases}$

where $B_{min} < B_{max}$ and $Q(B)$ is a monotonic increasing function that indicates the QoA of the application. Fig 4.1 shows the QoA functions of the five application groups. The function shape in the range $(B_{min}, B_{max})$ depends on the adaptability of the application as can be shown, which is defined by Lee [59]. To summarize, we illustrate the hierarchy of applications based on our definition in Figure 4.2.
Figure 4.1 Five groups of delay-tolerant applications

Figure 4.2 The hierarchy of applications
We can use this application model to classify delay-tolerant applications. For instance, we can take a massive file transfer program such as FTP as an example of an NBA application. FTP (File Transfer Protocol) provides better QoA as more bandwidth is made available and works to near zero bandwidth. Thus, its QoA rises over the full range of bandwidth not just a bounded range. Meanwhile, the Internet voice application with adaptive playout points shows the same QoA above a certain amount of bandwidth ($B_{\text{max}}$) and conversely an user cannot recognize any single word below a certain amount of bandwidth ($B_{\text{min}}$) due to massive packet drops occurring in the switches on the path. Thus, it belongs to the BA application group.

4.2 Service Classes and Allocation

As we explained above, applications have different response characteristic against delay and bandwidth: for instance some applications are delay-tolerant and others are not. To serve those applications, it is efficient for a network system to serve the flows of the diverse applications using several service classes, each of which provides different kinds of QoS according to its member characteristics. In this section, we define several service classes to handle various kinds of applications, and explain how an application flow can request its service class and obtain an allocation in our proposed architecture. We use the term 'periodic traffic' for the traffic of delay-intolerant applications, and 'random traffic' for the traffic of delay-tolerant applications in this dissertation.

4.2.1 Service class types

We define 4 types of service classes: periodic, absolute, weighted and best-effort, which are ordered from the highest priority to the lowest. The difference between simple priority system and the defined classes is that a higher priority class has the priority right within its share limit, and the flows in the same class have their shares. The shares could be absolute or weighted depending on the class. Also each class serves different kinds of QoS to its member flows since the classes are designed for specific types of applications. Thus, an application can choose a class that is suitable for its flow and receive service according to the class policy once obtaining an allocation. The class policy details are as followings:

**Periodic Class (PC):** This service class is mainly designed for delay-intolerant applications. A packet based telephone is one of the typical applications. Since a delay-intolerant application generates data that have fixed lifetimes, the flows of this class have higher service priority than any flows of the other 3 classes which are designed for delay-tolerant applications. The base station can limit the bandwidth allocated to this class to prevent the other 3 classes from
starvation of bandwidth as the periodic class monopolises the whole wireless bandwidth. A base station commits a statistical guarantee of packet drop probability to this class flows, and limits admitted flow number to maintain the probability figure. The statistically guaranteed packet drop probability and the admitted flow number are the system design parameters.

**Absolute Class (AC):** This class allocates bandwidth to a flow within the range of the remaining bandwidth after serving the periodic class flows. It provides a flow with a minimum constant bandwidth service. This class has higher service priority than weighed and best-effort classes. However, the higher service priority stands within its allocated bandwidth. When a flow does not use its allocated bandwidth, the other flows in this class can exploit the unused bandwidth based on their shares. Thus this class flows could receive more bandwidth than their allocation if there is at least an admitted flow of periodic or absolute class which does not fully use its allocated bandwidth. The unused bandwidth is not refundable, so a flow cannot save bandwidth by means of not using its allocated bandwidth. This class is suitable for BA, LBA and NA delay-tolerant applications that require minimum bandwidth to be executed.

**Weighted Class (WC):** This class provides a rate-based service for a member flow. The flows in this class are served according to their share rates within the remaining bandwidth after periodic and absolute class flows are served. Thus, the service amount of a flow in this class fluctuates dependent on the traffic load of periodic and absolute classes. Although this class cannot guarantee any bandwidth to its flows since it uses the remaining bandwidth which may be zero in a congested period, it guarantees fair bandwidth distribution among flows according to their share rates. This class is suitable for NBA and UBA applications that do not require minimum bandwidth guarantee to be executed and can adapt to varying bandwidth.

**Best-effort Class (BC):** This class flow cannot obtain any types of bandwidth share. Only when there is a vacant room after serving other class traffic, it can exploit the wireless link channels. Thus this class flow may not transmit any data upon a congestion period. This type is suitable for certain NBA and UBA applications that do not need immediate response such as E-mail programs.

### 4.2.2 Service Allocation

As introduced in Chapter 2, if a network adopts a receiver oriented reservation protocol such as RSVP, an application is responsible for setting up an incoming flow, sending its requests to the protocol. We assume that there is a such reservation protocol to link our proposed service and application.

In our proposed service architecture, when an entity (mobile host or the other end terminal)
requests a downlink or uplink flow service, it should send the class type and class parameters for
the flow to the base station. Then the base station allocates resources for the flow if it has enough
resources to admit the flow. In contrast, a best-effort class application does not need to request an
allocation to the base station since the best-effort class cannot have any allocation. Best-effort
class flows are served based on FCFS (First Come First Served) whenever a slot is available to
the best-effort class.

Figure 4.3 An example of a wireless link share tree

A base station manages the wireless link using the allocation tree that is composed of classes
and allocated flows. There are four levels in the tree as indicated in Figure 4.3. The top node
represents the total wireless link bandwidth which can be used in the cell where the base station is
located, and below that there are uplink and downlink classes layer that have 3 service classes as
child nodes. The bottom layer represents allocated flows. Since the best-effort class cannot have
any allocation, it is not included in the tree.
4.3 System Description

Let us consider a micro cellular packet switching system with few hundred-metre diameter cells, in which the round-trip propagation delay is of the order of 1μs. We assume that a channel reuse mechanism is deployed to suppress the interference from adjacent cells and the wireless link is error free.

An uplink and downlink of a cell share a common channel allocated to the cell, and the common channel is divided by slots of constant size. The SAC (Slot Allocation Controller) determines the sequence of uplink and downlink slots, and uplink/downlink traffic is conveyed via the slots allocated to each link. The mobile host may not receive its uplink slots in time upon high load. If it occurs, due to the lifetime of periodic traffic packets the mobile host drops the those packets exceeding their valid periods instead of transmitting them. The packet drop strategy does not apply to the random traffic that is delay endurable. As the slot size is constant, the transport/network layer datagrams may be fragmented or assembled into the slot size before loaded in slots.

The base station broadcasts an uplink-slot start beacon (USB) at the beginning of an uplink slot to inform mobile hosts of an uplink slot starting point as can be seen in Figure 4.5. Thus, the mobile hosts can access uplink slots whenever they listen to USB, whereas the base station just exploits downlink slots to transmit downlink data without signals. This uplink-polling scheme may not be efficient in the large cell system because the response time to the signal is long due to the long round-trip propagation delay. However, in a small cell system like our proposed system, the round-trip propagation delay is very small. Wu [97] shows that such uplink-polling strategy in a small cell system is more beneficial than a frame based strategy, where uplink slot locations are informed at the beginning of a frame that consists of multiple slots, due to the flexible uplink usage of the uplink-polling system.

4.3.1 Traffic Scheduling

Our proposed service policy is achieved by uplink/downlink traffic schedulers located at a base station which schedule uplink/downlink traffic in order to fulfil the services committed to flows. Even though uplink and downlink support the same service classes, the ways that both the traffic schedulers work should be different because the downlink is a broadcast channel while the uplink is a shared channel. We explain below how both the traffic schedulers operate to achieve the committed service. The detail queuing disciplines and multiple access protocol will be covered in the following chapters.
4.3.1.1 Downlink traffic scheduling

To achieve the committed class services in a downlink, we adopt a downlink scheduler which multiplexes downlink traffic coming from fixed networks. The downlink scheduler consists of PCAWFQ and FCFS (First Come First Served) scheduling disciplines. The PCAWFQ, which we develop and explain details in Chapter 5, serves the three classes of traffic - absolute, weighted and best-effort – as we defined in the above section, and the FCFS serves periodic traffic.
The data packets may be fragmented or assembled to the slot size before being queued in the downlink scheduler, and the downlink scheduler places the incoming packets in either FCFS or PCAWFQ queue according to the packet class type. FCFS always has higher priority than PCAWFQ in using downlink slots. Thus, PCAWFQ uses the downlink slots only when FCFS queue is empty: i.e. the PCAWFQ uses the residual slots after serving the periodic traffic packets. Whenever SAC assigns a slot to the downlink, the downlink scheduler chooses one queue among both queues according to the explained rule and transfers the packet at the head of the chosen queue to SAC. Then SAC broadcasts the downlink slot data to the mobile host. Figure 4.4 illustrates how the downlink scheduler works.

4.3.1.2 Uplink traffic scheduling

A multiple access protocol enables mobile hosts to share a common channel. To provide uplink flows with QoS, a multiple access protocol should associate with a scheduling discipline which can schedule uplink channel slots to convey packets scattered over the mobile hosts according to the committed QoS. Even though many multiple access protocols [42], [69], [64], [97], [13], [4], [83] have been proposed, those cannot be used in our proposed service architecture since they support only simple QoS. Therefore, we propose a slot based FQMA (Fair Queuing Multiple Access) protocol, which can serve uplink traffic according to our proposed class service policy.

Although the FQMA performs uplink traffic scheduling to achieve the same class based services as a downlink does, the used methods are slightly different from those of the downlink since the uplink is a shared medium. Unlike the downlink traffic scheduling, we do not know whether or not the queues of uplink sources in the mobile hosts are empty or how long their packets have waited for service. This information is essential to serve the periodic traffic with higher priority than random traffic, maximizing channel utilization. For instance, an uplink slot may be wasted if it is assigned to a periodic source which queue is empty, or it is difficult to decide when is the time to give an uplink slot to random traffic. Thus without those information we cannot serve random traffic without affecting the periodic traffic service, maximizing the channel utilization. FQMA can estimate the periodic traffic queues status of mobile hosts using parameters passed from mobile hosts. Based on this parameters, Slot Type Selector (STS) shown in Figure 4.4 allocates uplink slots to both traffic achieving those goals. The details of FQMA and STS will be presented in Chapter 6.

We propose two scheduling disciplines to allocate uplink slots in FQMA: PEDQ (Packet Endurable Delay Queuing) and PCAWFQ'. The PEDQ schedules the uplink periodic traffic in
ascending order of the remaining lifetimes of packets to maximise channel usage - we will explain PEDQ in detail in Chapter 6. It is more efficient than FCFS, which is adopted for the downlink periodic traffic scheduling. However, we cannot apply PEDQ to the downlink scheduling because we cannot estimate the remaining lifetime of a downlink packet. The packet header of a usual transport protocol does not embody any clue to calculate it. Although uplink periodic traffic scheduling performs better than the FCFS since it can use more information from the mobile hosts through MAC level packet header, the uplink periodic traffic scheduling should spend additional channel costs for contention among mobile hosts and communicating with mobile hosts. After all, the uplink and downlink scheduler for periodic traffic have performance reduction factor respectively: i.e. the uplink scheduler wastes channels due to channel sharing, while the downlink scheduler does since it cannot access the remaining lifetime of packets. We will show how the negative factors affect both scheduler performances in Chapter 6.

PCAWFQ' is a modified PCAWFQ and is designed for scheduling uplink random traffic. Since PCAWFQ is designed for single queue, it cannot be applied to the uplink scheduling directly. The PCAWFQ' schedules uplink random packets as if real packets are queued in a single queue using proxy packets which are representing the real packets in the mobile hosts. After all, PCAWFQ' is the same discipline as PCAWFQ except that it uses proxy packets. We will explain the PCAWFQ' in detail in Chapter 6. The uplink schedulers in Figure 4.4 illustrate what we have presented in this section.

Meanwhile, when a mobile host receives its uplink slot, it transmits a slot of data. The data arrived at the base station is queued in the FCFS queue as can be seen in Figure 4.4 and transmitted to the wired network.

4.3.2 Uplink/downlink Sharing and Partitioning

Consider an integrated traffic service such as a third generation cellular network. In such a network, diverse sources, along with voice sources, should be served by various transport protocols. One of typical transport protocols, TCP usually sends data packets to one direction, whereas the other direction conveys only acknowledgement packets, which are much smaller than the data packets. Therefore, TCP connections are asymmetrically loaded between uplink and downlink in usual cases. The subscription of video or conference session also induces severe unbalanced usage between uplink and downlink channel. The asymmetrical load varies according to the types of involved sources and their transmitted direction. We also expect that typical users in an integrated services cellular network mainly download data from servers in a wired network. In such an asymmetric loaded environment, the uplink/downlink sharing is beneficial for high
channel utilisation. Even in a typical voice network, it has better performance than the partitioning scheme because the peak-load periods of uplink and downlink connections do not coincide with each other in micro level even if uplink and downlink traffic are statistically balanced in macro level

In our proposed system, channel slots can be dynamically assigned to uplink or downlink using SAC since uplink and downlink share a common channel. We propose two models of SAC: uplink/downlink partitioning scheme and sharing scheme. In the partitioning scheme, SAC allocates slots to uplink and downlink in alternate way so that one link uses half channels each, as indicated in Figure 4.5. Thus, the slot positions of uplink and downlink in a channel are fixed, and uplink and downlink each use exactly half slots among whole slots.

![Common Channel](Image)

**Figure 4.5 Common channel usage**

In contrast, the uplink/downlink-sharing scheme dynamically allocates channel slots to downlink and uplink, considering the traffic load of uplink/downlink. The dynamic slot allocation improves channel efficiency by reducing the unused slot rate. In the sharing scheme, the basic slot allocation policy is the same as the partitioning scheme: i.e. half slots are potentially assigned to both links alternately. However, a potentially assigned slot to one link can be used by the other link when certain conditions are satisfied. The flow chart in Figure 4.6 shows the conditions. Apparently, the sharing scheme performs better than the partitioning scheme. We will use the partitioning scheme in comparing performance with other systems that have separate channels for uplink/downlink.

SAC determines the final user (uplink or downlink) of a slot using the flow chart in Figure 4.6. Just before the beginning of a slot, odd numbered slots are potentially allocated to downlink, and even numbered slots to uplink. Then uplink and downlink schedulers select the candidate for next service among their flows respectively, and input them to SAC. If a link is supposed to serve random traffic in the next service and the other link periodic traffic, then SAC assigns the current slot to the other link. This simple algorithm gives higher priority to periodic traffic than random traffic among not one link but both links. If one link’s queue is empty, the other link uses the slot
unconditionally, and if both the links are empty, then the slot is allocated to uplink or downlink in alternate way.

This sharing model has three benefits. Firstly, it reduces the packet drop probability of periodic traffic, as a periodic traffic has higher priority than random traffic in not half channel but whole channel. The packet drops of periodic traffic usually occur during an overloaded period. Since the traffic patterns of uplink and downlink are different, the overloaded periods of uplink/downlink do not coincide each other. The sharing scheme allows one link to use slots of the other link when the other link is low-loaded, so the packet drops occur only when both links are in an overloaded period. Therefore, it lessens the chance of packet drops.

![Slot allocation flow chart for uplink/downlink sharing scheme](image)

Figure 4.6 Slot allocation flow chart for uplink/downlink sharing scheme

Secondly, it enables mobile hosts to use more channel slots than the partitioning scheme since a slot is wasted only when both links' queues are empty while in partitioning scheme a slot is wasted even when only one of both queues is empty. However, in the sharing scheme random traffic may have less allocated slots at high load compared with the partitioning channel scheme because more slots are allocated to the periodic traffic. However, the effect is not significant because the bandwidth reduction is within the range of packet drop probability.
Thirdly, the sharing scheme allows uplink and downlink flows to share the whole available random traffic slots because SAC can allocate the whole available random slots to each link according to its demand. On the contrary, in the partitioning scheme, the random traffic slots of uplink/downlink are de-coupled from each other, so the random channel slots cannot be utilised efficiently especially when the uplink and downlink traffic is asymmetric.
Chapter 5

Packet-by-Packet Combined Absolute & Weighted Fair Queuing (PCAWFQ)

In this Chapter, we propose a new packet scheduling discipline, PCAWFQ (Packet-by-Packet Combined Absolute and Weighted Fair Queuing) which is designed for scheduling random traffic in the downlink in our proposed service architecture, but it can be used for any network with variable rate links. It can simultaneously schedule three classes of flows - absolute class, weighted class and best-effort class - under a variable-rate link. Minimum bit rate services are guaranteed to the absolute class flows even when the available bandwidth fluctuates, provided that the available bandwidth is higher than the total bandwidth allocated to the absolute class flows. The weighted class flows can receive a rate-based scheduling service within the range of residual bandwidth after serving the absolute class flows. The best-effort class can exploit any remaining bandwidth based on FCFS (First Come First Served) following servicing of absolute and weighted class flows.

In contrast, rate-based schedulers such as WFQ provide only single class rate-based service. Apart from the simple scheduling function, they distribute bandwidth unfairly in a varying-rate link, whereas our proposed scheme does not have the unfairness problem since it is designed for a variable-rate link. In the next section, we will investigate the unfairness problem of WFQ under a variable-rate link and propose a modified WFQ for a variable-rate link, called V-WFQ (Variable-bandwidth WFQ) which remedies the unfairness problem. Following that, we will define an ideal CAFQ scheduler for a fluid-flow modeled packet network. Next we will derive PCAWFQ (Packet-by-packet CAFQ) for a practical packet network and compute the delay bound of the PCAWFQ. Then, we demonstrate how the proposed PCAWFQ works in a packet network using simulation, and then compare it with V-WFQ in terms of delay characteristic. Finally we discuss the results.

5.1 Influences of variable-rate links

An available bandwidth of a wireless link may fluctuate due to various reasons. Firstly it occurs due to channel contention among mobile hosts in a shared medium link. A contention based access protocol such as CSMA/CA forces available bandwidth to fluctuate due to
bandwidth waste by channel contention or backoff procedure. Secondly, transmission errors due to physical environment affect the available bandwidth of a wireless link. Wireless link error rates are very dynamic owing to physical environment such as multipath fading and propagation degradation. Finally, overlay networks, in which data connections use the unused bandwidth of voice connections such as (e.g. CDPD [21], GPRS [20] and PRMA [42]), have a variable-rate link to data connections. In the overlay networks, the available bandwidth is highly dynamic according to the number of active voice connections.

When GPS [71] serves flows over such a variable-rate link, the flow \( i \) with share rate \( r_i \) receives a guaranteed data rate, \( g_i(t) \) which is given by

\[
g_i(t) = \frac{r_i}{\sum_{j \in S} r_j} \cdot C(t)
\]

where \( C(t) \) is the outgoing link bandwidth and \( S \) is the set of flows with allocated share rates. In this case, the guaranteed bandwidth of each flow fluctuates as the available outgoing link bandwidth varies, i.e. if \( C(t) \neq \text{const.} \) then \( g_i(t) \neq \text{const.} \)

Assume that there is a minimum limit of \( C(t) \) for all \( t \), then the flow \( i \) receives at least the guaranteed bandwidth \( g_i^\text{min} \) for all \( t \) as follows:

\[
g_i^\text{min} = \frac{r_i \cdot \min_{t \geq 0} \{C(t)\}}{\sum_{j \in S} r_j}
\]

Let us define \( \phi \) to be the required bandwidth rate of the flow \( i \). Then, a GPS server should allocate a share rate to the flow \( i \) based on its minimum link capacity \( \min_{t \geq 0} \{C(t)\} \) to guarantee \( \phi \). Although this allocation scheme using the minimum limit of the outgoing link can guarantee a minimum constant bandwidth to a flow regardless of bandwidth fluctuation, it results in low utilisation of the link. The inefficiency increases as the minimum limit of an outgoing link decreases.

The packetised version of the GPS, WFQ (or PGPS) has another problem as well as the under-utilisation problem. It is the unfair bandwidth distribution among flows due to distortion of the virtual time caused by variable-rate links. As explained in Chapter 4, WFQ uses the virtual time as an usage indicator which advances by

\[
v(t_m) = v(t_{m-1}) + \frac{C(t)}{\sum_{j \in B(t_{m-1}, t_m)} r_j}
\]

where \( t_m \) are sequential event times of packet arrival and departure and \( B(t_{m-1}, t_m) \) is the backlogged set during the period \( [t_{m-1}, t_m] \). The service based on virtual time prevents a flow from
saving its unused bandwidth. For a variable-rate link, the use of $C(t)$ in computing the virtual time is not practical and the use of minimum limit of $C(t)$ allows the virtual time to tick faster, which causes two problems; unfair bandwidth distribution and no throughput guarantee.

For instance, if the total allocated bandwidth and actual traffic amount are higher than $\min t>0 \{C(t)\}$ respectively, the virtual time computed using the static rate $\min t>0 \{C(t)\}$ is ahead of the virtual finish time of the currently served packet since the virtual finish time reflects the fluctuating outgoing link rate while the virtual time does not. Since the virtual time indicates the bandwidth usage, the inaccurate virtual time results in unfair service. For this reason, GPS and PGPS (or WFQ) do not have the desired properties to support a guaranteed QoS for a varying capacity network.

SFQ and SCFQ do not produce the unfairness problem in a variable-rate link unlike WFQ, since they do not compute the virtual time. Instead, they simply use the virtual finish time of the served packet as the virtual time. However, as indicated in Chapter 4, SFQ and SCFQ basically have another unfairness problem which increases in proportion to the number of flows.

The packet-by-packet rate-based disciplines including the above three methods enable flows to share outgoing link bandwidth based on their allocated share rates. Accordingly, assuming that the link bandwidth is static, the bandwidth allocated to a flow is guaranteed over any backlogged period. However, in a variable-rate link, the guaranteed bandwidth fluctuates as the link capacity varies. When the total allocated bandwidth is over the minimum link capacity $\min t>0 \{C(t)\}$, none of the existing rate-based schemes guarantees the allocated bandwidth. Figure 5.1 shows the comparison of a rate-based discipline service in a fixed and variable bandwidth link.

![Figure 5.1 The guaranteed bandwidth effect of rate-based scheduling disciplines over a fixed and variable-rate link](image)

77
Figure 5.2 The guaranteed bandwidth effect of the proposed scheduling discipline (CAWFQ) over a fixed and variable bandwidth link

Although Goyal [43] provides an analysis tool for computing guaranteed bandwidth in a varying-rate link, his guaranteed bandwidth also fluctuates according to the burstiness of outgoing link bandwidth. Contrary to the rate-based scheduler property over a variable rate link, the integrated traffic requires various QoS including constant bit rate service, e.g. 64 kbits/s PCM voice. In the ATM traffic classification [75], class 1 is also provided for the constant bit rate service. However, all existing rate-based schemes are incapable of supporting such a service in a network where the individual outgoing link bandwidth is subject to fluctuation.

The PCAWFQ that we propose in this Chapter can provide absolute and weighted fair queuing services for multi-QoS flows to support integrated services even over a variable-rate link. Figure 5.2 shows the functional difference between our proposed scheme and existing rate-based schedulers.

5.2 V-WFQ (Variable-bandwidth WFQ)

As we mentioned above, since WFQ is designed for a fixed-rate link, it is difficult to compare our proposed PCAWFQ and WFQ in a variable-rate link. We, therefore, propose a modified WFQ, called V-WFQ which corrects the unfairness problem in a variable-rate link. V-WFQ enables the virtual time to tick correctly reflecting the outgoing bandwidth fluctuation, and is designed for a shared medium network in which data is transmitted at full constant speed once accessed even though the available outgoing bandwidth varies due to channel contentions or link sharing.

In a WFQ implementation, a WFQ server uses the virtual time to compute the virtual finish time of an arrived packet: i.e. the virtual time is only required at packet arrival events. While the virtual time is accessed only at packet arrival events, it must be updated at all packet arrival and
departure events because the ticking rate of the virtual time changes whenever the set of the backlogged flows is altered.

V-WFQ updates the virtual time at the packet arrival and departure events, replacing $C(t)$ with served bandwidth between the previous event and the current event: i.e. the proposed V-WFQ updates the virtual time at both events using the following equation.

$$v(t_m) = v(t_{m-1}) + \frac{C^* \left( \min(t_s, t_d) - \max(t_s, t_p) \right)}{\sum_{j \in B(t_m, t_{m-1})} r_j}$$

where $t_s$ and $t_e$ represent the service start and end time of the current (or previous - if a packet is not being served at $t_m$) served packet. $t_a$ and $t_p$ represent the arrival time of the current (or last – if $t_m$ is a departure event) packet and the previous (or second last) packet. $C$ is the full speed of the outgoing link. $t_m$ and $B(t_m, t_{m-1})$ follow the definitions in the previous section. In the equation, the upper term of the fraction indicates the served bandwidth over the interval $(t_{m-1}, t_m)$. With the proposed virtual time computation, the virtual time ticks based on the practical serving speed. Since it ensures that the virtual time and the virtual finish time of the served packet tick at the same rate, it eliminates the unfairness problem.

![Graph](image)

Figure 5.3 Left: The virtual time and virtual finish time of served packet in WFQ, Right: the difference of the two times

We identify the performance of the V-WFQ using simulations and also simulate WFQ to see how the variable-rate link affects WFQ fairness. Figures 5.3 and 5.4 show the results of the WFQ and V-WFQ simulations respectively. In both simulations, flow 1 and flow 2, which have the same share rate, start to transmit packets over a variable-rate link at 500ms and at 700ms each. WFQ uses a fixed minimum limit of the variable-rate link in computing the virtual time, and V-WFQ uses the equation as we proposed above to compute the virtual time. As can be seen in
Figure 5.3, in the case of the WFQ the difference between virtual time and virtual finish time of the served packet increases due to the accumulated bandwidth gap between practical service rate and the fixed minimum limit of the variable-rate link.

![Graph showing virtual time and virtual finish time comparison in WFQ](image)

Figure 5.4 Left: The virtual time and virtual finish time of served packet in V-WFQ, Right: the difference of the two times

![Service order diagrams for WFQ and V-WFQ](image)

Figure 5.5 The packet service order  Top: WFQ,  Bottom: V-WFQ

Let us look at how the time gap affects the service in WFQ. The virtual finish time of the first packet of flow 2 is computed based on the virtual time at 700 ms when flow 2 starts to transmit, but there is a time gap between the virtual time and virtual finish time of the served packet at 700 ms. Thus flow 2 cannot receive service until the virtual finish time of the served packet is equal to the virtual finish time of the first packet of flow 2: i.e. flow 1 keeps receiving service for a while even after flow 2 joins. The top picture of Figure 5.5 shows the unfairness problem. As can be seen, flow 2 starts to receive the first service at 910 ms even though it joins at 700 ms. In contrast to WFQ, the proposed V-WFQ can enable the virtual time and the virtual
finish time of the served packet advance at the same rate due to the reflection of real service rate in computing virtual time as depicted in Figure 5.4. Consequently flow 2 receives the first service as soon as it joins in V-WFQ as can be seen in the bottom of Figure 5.5. Furthermore, Figure 5.4 shows that the difference between virtual finish time of the served packet and virtual time is bounded in V-WFQ. If the gap is bounded, then the fairness is also bounded [52]. Therefore, the proposed V-WFQ can support fair bandwidth distribution even in a variable-rate link, maintaining the WFQ property.

5.3 Combined Absolute & Weighted Fair Queuing

V-WFQ solves the unfairness problem of WFQ over a variable-rate link, but it cannot provide a flow with a fixed guaranteed bandwidth independently of fluctuating outgoing bandwidth. PCAWFQ is designed to overcome this problem, handling three kinds of flow classes as we defined in Chapter 4. They are AC (Absolute Class), WC (Weighted Class), and BC (Best-effort Class). As the BC is served based on FCFS (First Come First Served) only when link bandwidth remains after serving the AC and WC, the BC traffic is queued in a separate queue and scheduled whenever the queue for the other two classes is empty. Therefore, we focus here on the scheduling discipline of the queue for AC and WC. In this section, we define the CAWFQ for a fictitious fluid-flow modeled network, and describe a practical packet-by-packet CAWFQ in the next section.

5.3.1 Definition of terms

We refer to a backlogged period as the period during which a server queue is never empty and an idle period as the period during which a server queue is empty. A backlogged period consists of membership-step periods. We refer to a membership-step period as the maximal period during which the membership of a backlogged set does not change. A leading membership-step sub-period is defined as a membership-step sub-period that starts at the beginning of a membership-step period. There are three kinds of membership-step periods according to the classes involved in the period; a-membership-step period (only AC flows exist in this period), w-membership-step period (only WC flows exist in this period) and mixed-membership-step period (Both AC and WC flows exist in this period). In the same way, we refer to a-backlogged period and w-backlogged period as the maximal consecutive membership-step periods that include AC flows and WC flows, respectively. Figure 5.6 shows an example of a backlogged and idle period. The figure also shows examples of leading membership-step sub-periods which are indicated by solid arrows.
In a variable-rate link, an arbitrary period can be classified as one of the following according to the available bandwidth:

**Definition 1:** A **deficit period** is any period during which the available bandwidth is lower than allocated bandwidth for the backlogged AC flows.

**Definition 2:** A **semi-deficit period** is any period during which the available bandwidth is higher than allocated bandwidth for the backlogged AC flows but lower than the allocated bandwidth for the backlogged AC and WC flows.

**Definition 3:** A **surplus period** is any period during which the available bandwidth is higher than the allocated bandwidth for the backlogged AC flows.

### 5.3.2 Definition of CAWFQ

The CAWFQ discipline is defined as follows: During a deficit membership-step period, the backlogged AC flows are always served with higher priority than the backlogged WC flows. During a semi-deficit and surplus membership-step period, the backlogged AC flows are served with higher priority than the backlogged WC flows within the range of their allocated bandwidth. Thus, the backlogged WC flows can exploit only the remaining bandwidth after servicing AC flows according to the priority policies. As a result, WC flows may receive less than their allocated bandwidth during a semi-deficit membership-step period and no bandwidth during a deficit membership-step period depending on the load of AC flows. However, during a surplus membership-step period, WC flows can receive their allocated bandwidth fully independently of the load of AC flows. In a surplus period, the backlogged AC and/or WC flows can share the remaining bandwidth according to their share rates.
When the membership of backlogged flows changes, the service relationship among flows before the change breaks up due to the emergence or disappearance of flow(s). Thus, in CAWFQ, a service relationship in a membership-step period is not carried over to the next membership-step period: i.e. a membership-step period is served independently of other membership-step periods. For instance, any bandwidth shortage experienced by flows in the previous membership-step period cannot be re-coupled during the present membership-step period.

The WC and AC flows maintain fairness among flows within its category. We call it local fairness. In a surplus membership-step period, the surplus bandwidth should be fairly shared among all flows regardless of service classes according to their share rates. We call it global fairness.

5.3.3 Formal expression

In this section, we define CAWFQ operating at a variable-rate fluid-flow server with a formal expression.

Consider a Combined Absolute & Weighted Fair Queuing (CAWFQ) that is a work-conserving server operating at a varying outgoing link $C(t)$. We refer to the fluid-flow as the property that the outgoing link of the server can serve multiple flows simultaneously. Let $A$ define the set of AC flows that are set up on the link. Similarly, define $W$ as the set of WC flows. Define $r_i$ as the service share (bit rate) allocated to the flow $i$, $i \in A \cup W$. Let $u_i(t)$ be the served rate of the flow $i$, $i \in A \cup W$. Define the served traffic amount of the flow $i$, $i \in A \cup W$ in the interval $(t_1, t_2]$ by $W_i(t_1, t_2)$: i.e.

$$W_i(t_1, t_2) = \int_{t_1}^{t_2} u_i(t)dt$$

Let $B(t_1, t_2)$ and $B(t)$ define the set of backlogged flows in the period $(t_1, t_2]$ and at the time $t$, respectively. Then the CAWFQ server is defined as the discipline which satisfies the following equations (1)-(6):

During a deficit or semi-deficit leading membership-step sub-period $(t_0, t_e]$ such that

$$\int_{t_0}^{t_e} C(t)dt < (t_e - t_0) \cdot \sum_{j \in B(t_0, t_e)} r_j$$

$$\sum_{j \in B(t_1, t_e) \setminus A} W_j(t_1, t_e) = \min\left(\int_{t_1}^{t_e} C(t)dt, \sum_{j \in B(t_1, t_e) \setminus A} r_j \cdot (t_e - t_1)\right)$$

$$\sum_{j \in B(t_1, t_e) \setminus W} W_j(t_1, t_e) = \max\left(0, \int_{t_1}^{t_e} C(t)dt - \sum_{j \in B(t_1, t_e) \setminus A} r_j \cdot (t_e - t_1)\right)$$

83
\[
\int \frac{u_i(t)}{r_i} \geq \frac{\int u_j(t)}{r_j} \quad \forall \ i,j \in B(t) \cap A
\]
(3)

\[
\int \frac{u_i(t)}{r_i} = \int \frac{u_j(t)}{r_j} \quad \forall \ i,j \in B(t) \cap W
\]
(4)

where \( t \) such that \( t_s < t \leq t_e \)

During a surplus leading membership-step sub-period \((t_s, t_e]\) such that

\[
\int_{t_s}^{t} C(t) dt \geq (t_e - t_s) \cdot \sum_{j \in B(t_s, t_e)} r_j
\]

\[
\sum_{j \in B(t_s, t_e)} W_j(t_s, t_e) = \int_{t_s}^{t} C(t) dt
\]
(5)

\[
\int \frac{u_i(t)}{r_i} = \int \frac{u_j(t)}{r_j} \quad \forall \ i,j \in B(t)
\]
(6)

where \( t \) such that \( t_s < t \leq t_e \)

According to Eq (3), (4) and (6), the fairness among the same class flows is upheld over a whole backlogged period. Eq (1), (2) and (5) imply that the bandwidth distribution of the discipline during a membership-step period is independently of that of other membership-step periods. For instance, even though an AC flow cannot receive its allocated service during a sub membership-step period owing to the reduction of available bandwidth, the service loss may be compensated sometime later within the same membership-step period if the available bandwidth rises enough to support the AC flows. However, the compensation quota disappears after the end of the membership-step period. Thus, a flow that has not received compensating bandwidth during a membership-step period cannot be compensated during the following membership-step periods. If the compensation scheme carries over in the next membership-step period, unfairness may occur. For instance, assume that a newly arrived flow enters in a membership-step period, and past compensation quota carries over, then the new flow may receive more or less service due to fair service of the discipline amongst same class flows. Therefore, it is not suitable to prolong the compensation quota.

To sum up, the CAWFQ schedules the packets of AC flows with higher priority until AC flows consume their shares, keeping fairness. WC flows are also served up to their shares if there is available bandwidth. The surplus bandwidth is distributed to all flows in a fair manner.
5.4 Packet-by-Packet CAWFQ (PCAWFQ)

The CAWFQ cannot be implemented in a practical packet network owing to the assumption of the parallel packet service. Thus, we consider a packetised version - PCAWFQ, which can schedule packets in a practical packet network as close as possible to CAWFQ. As an approximation for PCAWFQ, we could consider a packet-scheduling scheme that serves packets with the same service order in the corresponding CAWFQ. There are two ways to order the packets: based on service finish time or on start time. Figure 5.7 depicts one example of the service ordering based on the either finish time or start time of packets in CAWFQ. We adopt the former as the reference service order because the latter has disadvantages as explained in §3.1.6. Consequently, PCAWFQ is designed to serve packets with the same sequential order made by using the service finish time of packets in the corresponding CAWFQ.

However, as explained in Chapter 3, the packetised version of a fluid work-conserving server cannot serve packets with the same service order based on either finish or start time in the fluid server all the time over a whole backlogged period. For instance, in the example of Figure 5.7, the packetised scheduler cannot help choosing $P_3^2$ at the time $\odot$ for the next service even though $P_2^2$ has earlier finish time than $P_3^2$ because $P_2^2$ does not arrive by the time $\odot$. Only a non-work-conserving server – the server can stop service even when there is a packet to serve usually for controlling jitter – can achieve the same order service, but it may waste link bandwidth. This is undesirable characteristic for wireless links that have poor bandwidth. Therefore, we should design PCAWFQ as a work-conserving server.

![Figure 5.7 CAWFQ service and service sequential orders](image)

In order for PCAWFQ to schedule packets with the same service order in the corresponding CAWFQ, we should know the departure time of each packet in the corresponding CAWFQ. It is not easy to compute the departure time especially over varying bandwidth links. Thus, we
introduce the reference virtual time to discover the service order in the corresponding CAWFQ in a simple way.

5.4.1 Reference virtual time

To introduce the reference virtual time, we first define the virtual times for AC and WC. For a membership-step period \((t_1, t_2]\), let

\[ w_i^a(t) = \frac{\int_{t_1}^{t} u_i(t) dt}{r_i}, \quad \forall i \in B(t) \cap A \]  

\[ w_i^w(t) = \frac{\int_{t_1}^{t} u_i(t) dt}{r_i}, \quad \forall i \in B(t) \cap W \]

where \(t_1 < t \leq t_2\)

From the (7) and (8), we get that (3), (4) and (6) follow that

\[ w_i^a(t) = w_j^a(t), \quad \forall i, j \in B(t) \cap A \]

\[ w_i^w(t) = w_j^w(t), \quad \forall i, j \in B(t) \cap W \]

Since all \(w_i(t), i \in B(t) \cap A\) have the same value at the time \(t\), we introduce a representative function, \(v_a\) as the following:

\[ v_a(t) = w_i^a(t), \quad \forall i \in B(t) \cap A \]

(9)

Similarly, for \(w_i^w\), \(v_w\) can be defined as following:

\[ v_w(t) = w_i^w(t), \quad \forall i \in B(t) \cap W \]

(10)

We call \(v_a\) a-virtual time and \(v_w\) w-virtual time. The virtual times \(v_a\) and \(v_w\) advance in proportion to the normalised service amount of each class. The rates of \(v_a\) and \(v_w\) change according to the fluctuation of outgoing link bandwidth.

The virtual time increment for a semi-deficit or deficit leading membership-step sub-period \((t_s, t_e]\), from (1), (7) and (9), we get

\[ v_a(t_e) - v_a(t_s) = \frac{\min\left(\int_{t_s}^{t_e} C(t) dt, (t_e - t_s) \cdot \sum_{i \in B(t_s, t_e) \cap A} r_i\right)}{\sum_{i \in B(t_s, t_e) \cap A} r_i} \]

(11)

Similarly, according to (2), (8) and (10), we have

\[ v_w(t_e) - v_w(t_s) = \frac{\max\left(0, \int_{t_s}^{t_e} C(t) dt - (t_e - t_s) \cdot \sum_{i \in B(t_s, t_e) \cap A} r_i\right)}{\sum_{i \in B(t_s, t_e) \cap W} r_i} \]

(12)
For a surplus leading membership-step sub-period, from Eq (5) and (7)-(10) we get

\[ v_a(t_e) - v_a(t_s) = v_w(t_e) - v_w(t_s) = \frac{\int_{t_s}^{t_e} C(t) dt}{\sum_{i \in B(t_s,t_e)} r_i} \]  

(13)

Eqs (11)-(13) show the increment of the two virtual times during any leading membership-step sub-period, \([t_s, t_e]\).

Now let us look at how the virtual times advance between any interval within a membership-step period. Choose any \(t\) and \(\tau\) such that \(t < \tau\) within a semi-deficit or deficit membership-step period beginning at time \(t_s\). Subtracting the equation made by substituting \(t\) into the variable \(t_e\) of (11) from the equation made by substituting \(\tau\) into the variable \(t_e\) of (11), for a semi-deficit or deficit membership-step period we have:

\[ v_a(\tau) - v_a(t) = \frac{\min(\int_{t_s}^{\tau} C(t) dt, (\tau-t_s) \cdot \sum_{i \in B(t_s,\tau) \cap A} r_i)}{\sum_{i \in B(t_s,\tau) \cap A} r_i} \]

\[ = \frac{\min(\int_{t_s}^{\tau} C(t) dt, t \cdot (\tau-t_s)) - \min(\int_{t_s}^{t} C(t) dt, t \cdot (t-t_s))}{\sum_{i \in B(t_s,t) \cap A} r_i} \]

(14)

Similarly, we have

\[ v_w(\tau) - v_w(t) = \frac{\max(0, \int_{t_s}^{\tau} C(t) dt - \sum_{i \in B(t_s,\tau) \cap A} r_i \cdot (\tau-t_s)) - \max(0, \int_{t_s}^{t} C(t) dt - \sum_{i \in B(t_s,t) \cap A} r_i \cdot (t-t_s))}{\sum_{i \in B(t_s,t) \cap W} r_i} \]

(15)

For a surplus membership-step period, we have

\[ v_a(\tau) - v_a(t) = v_w(\tau) - v_w(t) = \frac{\int_{t_s}^{\tau} C(t) dt - \int_{t_s}^{t} C(t) dt}{\sum_{i \in B(t_s,\tau)} r_i} = \frac{\int_{t_s}^{\tau} C(t) dt}{\sum_{i \in B(t_s,\tau)} r_i} \]

(16)

**Lemma 1**: In CAWFQ, let \(t_0^+\) be the beginning point of a new membership-step period and \(t_0^-\) be the end point of the previous membership-step period, then \(v_a(t_0^+) = v_w(t_0^+)\) irrespective of relationship of \(v_a(t_0^-)\) and \(v_w(t_0^-)\).

**Proof**: By the virtual time definition, \(v_a(t_0^+)\) and \(v_w(t_0^+)\) are measurers of the normalised services by \(t_0^+\) for AC and WC respectively. Assume that \(v_a(t_0^+) \neq v_w(t_0^+)\). The assumption means that AC flows have received more normalised service than WC flows by \(t_0^+\) because AC flows are served with higher priority than WC flows. Then, the inequality of both virtual times comes
from the service result of the previous membership-step period. However, it is against the
definition of CAWFQ that the service of each membership-step period should be independent of
the other membership-step periods. Therefore, \( v_a(t_0^+) \) should be equal to \( v_w(t_0^+) \) irrespective of
the relationship of \( v_a(t_0) \) and \( v_w(t_0) \)
\[ \square \]

Since \( v_a \) and \( v_w \) might not be equal at the end point of a membership-step period, they should
be made equal at the beginning of a membership-step period according to Lemma 1. Now, let us
consider how to balance them. There are three kinds of membership-step periods: a-membership-
step, w-membership-step, and mixed-membership-step period. In addition, a membership-step
period may change into an idle period or vice-versa. Thus there are 16 possible combinations of
transitions which need virtual time balance. Among the 16 transition combinations, only 12
period transitions need the initial virtual times because transitions to an idle period do not need
the initial virtual times. The initial virtual times for the 9 transitions out of 12 are self-evident as
followings:

\[
\begin{align*}
\text{w-membership-step OR idle period} & \Rightarrow \text{a-membership-step period} \quad : v_a(t_0^+) = 0 \\
\text{a-membership-step OR idle period} & \Rightarrow \text{w-membership-step period} \quad : v_w(t_0^+) = 0 \\
\text{idle period} & \Rightarrow \text{mixed-membership-step period} \quad : v_a(t_0^+) = 0, v_w(t_0^+) = 0 \\
\text{mixed- OR a-membership-step period} & \Rightarrow \text{a-membership-step period} \quad : v_a(t_0^+) = v_a(t_0) \\
\text{mixed- OR w-membership-step period} & \Rightarrow \text{w-membership-step period} \quad : v_w(t_0^+) = v_w(t_0)
\end{align*}
\]

where \( t_0^+ \) and \( t_0^0 \) follow the definition of Lemma 1.

The other three transition cases are follows: For the transition to a mixed-membership-step
period from a w-membership-step period, the initial value of w-virtual time should be adjusted by
\( v_w(t_0^+) = v_w(t_0) \). How should the initial value of a-virtual time be set? By the definition of
CAWFQ, newly backlogged AC flow(s) should not be affected by existing flows when entering a
membership-step period. So the initial a-virtual time should be adjusted by \( v_a(t_0^+) = v_w(t_0) \). This
means that the normalised service amount of the new AC flow(s) equals that of WC flows at \( t_0^+ \).

Similarly, for the transition to a mixed-membership-step from an a-membership-step period, the
initial virtual times are set by \( v_a(t_0^+) = v_a(t_0), v_w(t_0^+) = v_a(t_0) \). To decide initial virtual times for
the transition to a mixed-membership-step period from another mixed-membership-step period,
let us examine the relationship between \( v_a(t) \) and \( v_w(t) \)

**Lemma 2:** In a mixed-membership-step period, \( (t_1, t_2], v_a(t) \geq v_w(t) \) for all \( t \) such that
\[ t_1 < t < t_2 \]

**Proof:** The proof is presented in Appendix A.1.
The transition to a mixed-membership-step from another mixed-membership-step period occurs due to membership change without removing all AC or WC flows. In the transition, if the two virtual times are not equal at $t_0^-$, we should choose either the a-virtual time or w-virtual time at $t_0^-$ as a reference time and set the two initial virtual times at $t_0^+$ to the reference time. Since we cannot deprive a flow of service received in the past, the two virtual times should not decrease during all their backlogged periods. Therefore, we should take a-virtual time as the reference since AC always has higher virtual time than WC flows according to Lemma 2, i.e. $v_a(t_0^-) = v_a(t_0^-)$, $v_w(t_0^+) = v_a(t_0^-)$. This ensures that the w-virtual time function does not decrease between $t_0^-$ and $t_0^+$.

**Lemma 3:** The virtual times $v_a(t)$ and $v_w(t)$ are non-decreasing functions of time during a-backlogged and w-backlogged period, respectively.

**Proof:** The proof is presented in Appendix A.2.

**Lemma 4:** $v_a(t)$ and $v_w(t)$ are identical functions over a mixed-membership-step period under the ideal link such that

$$C_f(t) = \sum_{k \in B(t)} r_k$$

(17)

**Proof:** The proof is presented in Appendix A.3.

According to Lemma 4, $v_a$ equals $v_w$ under the ideal link, $C_f(t)$. Let us introduce an ideal virtual time, $\nu_i$ which is the virtual time $v_a$ or $v_w$ computed under the ideal link. Let us define $\xi(t)$ as a-delay time that is the function of the time gap between $\nu_i$ and $v_a$, and $\delta(t)$ as w-delay time that is the function of the time gap between $\nu_i$ and $v_w$ under any practical link $C(t)$, that is,

$$\nu_i(t) = v_a(t) + \xi(t), \quad \nu_i(t) = v_w(t) + \delta(t)$$

Then, we have:

$$v_a(t) = v_w(t) + \delta(t) - \xi(t)$$

(18)

The a-delay time and w-delay time vary according to the current available bandwidth, representing the shortage of the normalised service of AC and WC flows, respectively. Then, the delay time gap, $\delta(t) - \xi(t)$ represents the normalised service time difference between AC and WC flows. Therefore, by the definition of CAWFQ,

$$\delta(t) - \xi(t) = 0 \quad \text{during a surplus period}$$

$$\delta(t) - \xi(t) > 0 \quad \text{during a semi-deficit and deficit period}$$

The $\delta(t)$ and $\xi(t)$ should be zero whenever a membership-step period ends so that the service status in the pervious membership-step period does not affect the new membership-step period.

Now, let us define $R_k^f$ the reference virtual time of $p_k^f$ such that
\[
R^j_k = \begin{cases} 
\nu_a(d^j_k), & k \in A \\
\nu_w(d^j_k) + \delta(d^j_k) - \xi(d^j_k), & k \in W 
\end{cases}, \quad k \in A 
\]

where \( p^j_k \) represents \( j^{th} \) packet of the flow \( k \) and \( d^j_k \) represents the departure (service finish) time of \( p^j_k \). Figure 5.8 illustrates the relationship between the packet departure time and the corresponding reference virtual time. As can be seen, the order of packet departure times equals the order of corresponding reference virtual times since \( \nu_a \) is a non-decreasing function according to Lemma 3. Therefore, we have the following corollary:

**Corollary 1:** In the CAWFQ, the packet departure time order \( d^j_k \), is the same as the reference virtual time order of the packets, \( R^j_k \), where \( k \in A \cup W \) and \( j = 1, 2, 3 \ldots \)

Reference Time

\[ \begin{array}{c}
R_1^3 \\
R_2^3 \\
R_1^2 \\
R_2^2 \\
R_1^1 \\
R_2^1 \\
d_1^1 \\
d_2^1 \\
d_1^2 \\
d_2^2 \\
d_1^3 \\
d_2^3 \\
\end{array} \]

Departure Time

\[ \nu_a(t) \]

**Figure 5.8 Relationship between the packet departure times and the reference virtual times**

### 5.4.2 Implementation of PCAWFQ

On account of Corollary 1, we can define PCAWFQ using the reference virtual time as followings:

**Definition 4:** The work-conserving PCAWFQ server schedules packets in increasing order of the reference virtual times of queued packets.

We now discuss how to compute the reference virtual time of a packet under PCAWFQ. Since the w-delay time and a-delay time advance in proportion to the shortage of bandwidth, the delay time gap \( \delta(t) - \xi(t) \) also varies as available bandwidth fluctuates. This means the reference virtual time of a WC packet cannot be fixed while the packet is in the queue. Thus, we should compute reference virtual times of all queued packets each time we schedule and choose a packet with earliest reference virtual time, even though it is very inefficient. Therefore, we introduce the virtual finish time of a packet, \( F^j_k \) such that
\[ F_k^j = \begin{cases} R_k^j, & k \in A \\ R_k^j - \delta(t) + \xi(t), & k \in W \end{cases} \]

\( F_k^j \) is a fixed value from which we can easily compute the reference virtual time of a packet whenever PCAWFQ needs it to choose a packet for service. The sequence order of WC packets sorted by virtual finish times is equal to that sorted by reference virtual times because all packets have the same delay time gap when they are compared. Thus, we can sort the WC packets using the virtual finish times. The problem arises when WC and AC packets are sorted together because the reference virtual times of WC packets have the delay time gap which varies against time. To solve this sorting problem, we use two separate queues for both classes. All packets of each class are queued in increasing order of virtual finish times within its queue. When choosing a packet for service, PCAWFQ picks up the packet placed at the head of WC queue and computes the reference virtual time of the packet by adding the current delay time gap to the virtual finish time of the packet. The reference virtual times of all AC packets are the same as the virtual finish time of the packets. After computing both the reference virtual times of the AC packet and WC packet placed at each queue’s head, PCAWFQ compares their reference virtual times choosing the packet that has the earlier reference virtual time. The virtual finish times of packets can be obtained using the following Theorem:

**Theorem 1:** For a membership-step period beginning at \( t_0 \) in a PCAWFQ server, let \( a_k^j \) be the arrival time of \( p_i^j \), \( j^{th} \) packet of \( k^{th} \) flow and let \( d_k^j \) be the departure time of \( p_i^j \). Let \( L_k^j \) be the length of \( p_i^j \). Let us define the virtual start time, \( S_k^j \) such that

\[
S_k^j = \begin{cases} \max(F_k^{j-1}, v_a(a_k^j)), & k \in A \\ \max(F_k^{j-1}, v_w(a_k^j)), & k \in W \end{cases} \quad j = 1,2,.....
\]

Then the virtual finish time \( F_k^j \) is represented as the following relationship:

\[
F_k^j = S_k^j + L_k^j / r_k, \quad k \in A \cup W, \quad j = 1,2,..
\]

\[
F_k^0 = \begin{cases} v_a(t_0^+), & k \in A \\ v_w(t_0^+), & k \in W \end{cases}
\]

**Proof:** The proof is presented in Appendix A.4.

Theorem 1 provides a tool to compute the virtual finish time of a packet when the packet arrives at PCAWFQ.

Now, if we can obtain \( v_a(t) \) and \( v_w(t) \) for computing the virtual finish time and the delay time gap, \( \delta(d_k^j) - \xi(d_k^j) \) for computing the reference virtual time from the virtual finish time, then we can discover the packet departure order in the corresponding CAWFQ. Let us look at how to calculate
the virtual times and delay time gap. Let \( t_m \) define the event sequence time of a packet departure and arrival over a membership-step period beginning at \( t_0 \), where \( m \) becomes 0 whenever a new membership-step period begins. Let \( \rho_a(t_{m-1}, t_m) \) and \( \rho_w(t_{m-1}, t_m) \) be the allocated service amount for AC and WC in the interval \((t_{m-1}, t_m]\), and \( S(t_{m-1}, t_m) \) be the served amount to the AC and WC in the interval \((t_{m-1}, t_m]\), i.e.

\[
\begin{align*}
\rho_a(t_{m-1}, t_m) &= \sum_{j \in B(t_{m-1}, t_m)} r_j \cdot (t_m - t_{m-1}) \\
\rho_w(t_{m-1}, t_m) &= \sum_{j \in B(t_{m-1}, t_m)} r_j \cdot (t_m - t_{m-1}) \\
S(t_{m-1}, t_m) &= \sum_{j \in B(t_{m-1}, t_m)} W_j \cdot (t_m - t_{m-1})
\end{align*}
\]

where \( m = 1, 2, \ldots \)

Then, a-delay time \( \xi(t_m) \) evolves as follows:

\[
\xi(t_m) = \max(0, \xi(t_{m-1}) + \rho_a(t_{m-1}, t_m) - S(t_{m-1}, t_m)) \quad \sum_{j \in B(t_{m-1}, t_m)} r_j
\]

where \( \xi(t_0) = 0, \ m = 1, 2, \ldots \)

\( v_a(t) \) and \( v_w(t) \) evolve as follows:

\[
\begin{align*}
v_a(t_m) &= \begin{cases} 
  v_a(t_{m-1}) + \min(S(t_{m-1}, t_m), S(t_{m-1}, t_m) - \Re_a(t_m)) \\
  v_a(t_{m-1}), \rho_a(t_{m-1}, t_m) \neq 0 \\
  v_a(t_{m-1}), \rho_a(t_{m-1}, t_m) = 0
\end{cases} \\
v_w(t_m) &= \begin{cases} 
  v_w(t_{m-1}) + \max(0, \min(\Re_a(t_m), \Re_a(t_m) - \Re_w(t_m))) \\
  v_w(t_{m-1}), \rho_w(t_{m-1}, t_m) \neq 0 \\
  v_w(t_{m-1}), \rho_w(t_{m-1}, t_m) = 0
\end{cases}
\end{align*}
\]

where \( v_a(t_0) \) and \( v_w(t_0) \) follow the initial condition as explained in the previous section, \( m = 1, 2, \ldots \), and \( \Re_a(t_m) = S(t_{m-1}, t_m) - \rho_a(t_{m-1}, t_m) - \xi(t_{m-1}) \cdot \sum_{j \in B(t_{m-1}, t_m)} r_j \)

\[
\Re_w(t_m) = \Re_a(t_m) - \rho_w(t_{m-1}, t_m) - \delta(t_{m-1}) \cdot \sum_{j \in B(t_{m-1}, t_m)} r_j
\]

\[
\psi(t_{m-1}, t_m) = \max(0, \Re_w(t_m)) \cdot \sum_{j \in B(t_{m-1}, t_m)} r_j
\]

The w-delay time \( \delta(t_m) \) evolves as follows:

\[
\delta(t_m) = \begin{cases} 
  \max(0, \delta(t_{m-1}) + \frac{\rho_w(t_{m-1}, t_m) - \Re_a(t_m)}{\sum_{j \in B(t_{m-1}, t_m)} r_j}), \rho_w(t_{m-1}, t_m) \neq 0 \\
  \delta(t_{m-1}), \rho_w(t_{m-1}, t_m) = 0
\end{cases}
\]

92
where $\delta(t_0) = 0$, $m=1,2,\ldots$

Using the above equations, we can obtain a-virtual, w-virtual time and delay time gap $\delta(t_m) - \xi(t_m)$.

### 5.5 Delay bound of PCAWFQ

In this section, we will investigate the delay bounds of the AC flows in PCAWFQ when the flows are characterized by the leaky bucket model [92].

#### 5.5.1 Leaky bucket traffic

Bounding traffic characterization can either be deterministic or stochastic. Many traffic models have been proposed to compute deterministic traffic characteristics. Among them, the leaky-bucket model [92] has been widely used. In the model, an incoming packet of a traffic stream can enter the network only after removing $n$ bits from the token bucket, where $n$ is the size of the packet. The token bucket is filled at a fixed rate $\rho$ and contains at most $\sigma$ bits worth of tokens, and initially contains $\sigma$ bits worth of tokens at the beginning of a backlogged period. If the available tokens in the bucket are smaller than the incoming packet size, the packet is buffered and waits until the token queue length is bigger than the packet size. Therefore, an incoming traffic stream is constrained by two parameters $\rho$ (average rate) and $\sigma$ (burstiness). Let $A_k(\tau, t)$ be the arrival traffic amount of flow $k$ during $(\tau, t)$, then under the leaky bucket constraint, $A_k(\tau, t)$ is bounded as the following equation:

$$A_k(\tau, t) \leq \sigma + \rho (t - \tau), \sigma > 0, \rho > 0$$

In the above leaky bucket model, $\rho$ and $\sigma$ can be viewed as the long term bounding rate and the maximum burst size of a source traffic, respectively. If we can restrict a traffic stream with the leaky bucket model, deterministic delay bound of a traffic stream can be computed.

#### 5.5.2 Delay guarantee

Most scheduling schemes [71], [9], [101] can guarantee a deterministic delay bound to a flow constrained by leaky bucket under a fixed rate link, but no scheduling scheme can guarantee it under a varying rate link even if a leaky bucket filter is adopted. Goyal [43] shows that the guaranteed delay bound fluctuates depending on the burstiness of available bandwidth of the link. Now we will investigate the delay characteristic of CWFQ.

**Definition 5:** CWFQ or PCAWFQ is labeled as *semi-stable* if $\sum_{k \in A} r_k \leq \min_{t>0} \{C(t)\}$ for $\forall t$ and labeled as *fully-stable* if $\sum_{k \in A \cup W} r_k \leq \min_{t>0} \{C(t)\}$ for $\forall t$.  

93
Lemma 5: A semi-stable CAWFQ provides AC flows with a bounded delay guarantee if all AC flows are constrained by the leaky bucket model and \( \sum_{k \in A} \rho_k \leq \min_{t > 0} \{ C(t) \} \) for \( \forall t \)

Proof: Suppose that \( (\tau, t] \) is an a-backlogged period. Let \( A_a(\tau, t) \) and \( W_a(\tau, t) \) be the summed arrival amount of AC flows and served amount to all AC flows during the period, respectively. Let \( W_w(\tau, t) \) be the total served amount to all WC flows during the period \( (\tau, t] \).

Then in a CAWFQ \( W_a(\tau, t) \) must equal \( A_a(\tau, t) \) since \( (\tau, t] \) is an a-backlogged period and semi-stable.

According to (28)
\[
W_a(\tau, t) = A_a(\tau, t) \leq \sum_{k \in A} (\sigma_k + \rho_k (t - \tau))
\]

Since \( (\tau, t] \) is a backlogged period and CAWFQ is a work-conserving server,
\[
W_a(\tau, t) + W_w(\tau, t) \geq \min_{t \in } \{ C(t) \} \cdot (t - \tau)
\]

Thus, we have
\[
\min_{t \in } \{ C(t) \} \cdot (t - \tau) - W_w(\tau, t) \leq \sum_{k \in A} (\sigma_k + \rho_k (t - \tau)),
\]

which yields
\[
t - \tau \leq \frac{\sum_{k \in A} \sigma_k + W_w(\tau, t)}{\min_{t \in } \{ C(t) \} - \sum_{k \in A} \rho_k}
\]

Since \( W_w(\tau, t) \) is bounded by the link capacity and \( \sum_{k \in A} \rho_k \leq \min_{t > 0} \{ C(t) \} \), the above inequality shows that the length of an a-backlogged period is bounded. Therefore, the delays of all AC flows are bounded.

The global minimum of the link bandwidth could be very tiny in practice. In that case the CAWFQ server can allocate only a small part of the whole bandwidth to AC flows so that CAWFQ server can remain at least semi-stable. Therefore, we also define the stability condition based on a sub-period not a whole period in order to increase allocation efficiency.

Definition 6: CAWFQ or PCAWFQ is labeled as semi-stable during the period \( (t_1, t_2] \) if
\[
\sum_{k \in A} r_k (t_2 - t_1) \leq \min_{t_1 < t_2} \{ C(t) \} \cdot (t_2 - t_1) \text{ for } t_1 < t_2
\]

and labeled as fully-stable during the period \( (t_1, t_2] \) if \( \sum_{k \in A \cup W} r_k (t_2 - t_1) \leq \min_{t_1 < t_2} \{ C(t) \} \cdot (t_2 - t_1) \) for \( t_1 < t_2 \)

Considering Lemma 5, we can easily conclude the following corollary

Corollary 2: A semi-stable CAWFQ in the period \( (t_1, t_2] \) provides AC flows with bounded delay guarantee if all AC flows are constrained by the leaky bucket model and
\[
\sum_{k \in A} \rho_k (t_2 - t_1) \leq \min_{t_1 < t_2} \{ C(t) \} \cdot (t_2 - t_1), \text{ for } t_1 < t_2
\]
Lemma 5 and corollary 2 show that a semi-stable CAWFQ guarantees a bounded delay to all AC flows even though the bound is not tight. Now, let us compute the delay bound of an AC flow under a semi-stable PCAWFQ.

**Theorem 2:** If there is no allocated WC flows in a semi-stable PCAWFQ server, then the PCAWFQ server is identical with the equivalent V-WFQ server.

**Proof:** During a backlogged period \((t_1, t_2)\) since PCAWFQ is semi-stable and work-conserving, and serves only AC flows, we have

\[
\min_{t_1 \leq t \leq t_2} \{C(t)\} \cdot (t_2 - t_1) \leq \sum_{k \in B(t_1, t_2)} W_k(t_1, t_2) 
\]  

(29)

According to the definition 6 and (29), in the semi-stable PCAWFQ server we have

\[
\sum_{k \in B(t_1, t_2)} r_k \cdot (t_2 - t_1) \leq \sum_{k \in B(t_1, t_2)} W_k(t_1, t_2) 
\]  

(30)

Substituting (30) into (21), we conclude

\[
\xi(t_m) = 0 \quad m = 0, 1, 2, 3, \ldots 
\]  

(31)

Then, substituting (31) and (24) into (25), we get

\[
R_w(t_m) = S(t_{m-1}, t_m) - \rho_a(t_{m-1}, t_m) - \rho_a(t_m, t_{m+1}) - \delta(t_m) \cdot \sum_{j \in B(t_m, t_{m+1})} r_j 
\]

Since there is no allocated WC flows, \(\rho_w(\tau, t) = 0\). From (27) \(\delta(t_m) = 0\) for all \(m\).

Thus, we have

\[
R_w(t_m) = S(t_{m-1}, t_m) - \rho_a(t_{m-1}, t_m) = R_a(t_m) \quad \text{for all } m 
\]  

(32)

Substituting (32) into (26) and since \(R_a(t_m) \geq 0\) from (30), we have

\[
\psi(t_{m-1}, t_m) = \frac{R_a(t_m)}{\sum_{k \in B(t_{m-1}, t_m)} r_k} \quad m = 1, 2, 3, \ldots 
\]  

(33)

Then, the a-virtual time is increased by the following equation from (22) and (33)

\[
v_a(t_m) = v_a(t_{m-1}) + \frac{\sum_{k \in B(t_{m-1}, t_m)} W_k(t_{m-1}, t_m)}{\sum_{k \in B(t_{m-1})} r_k} = v_a(t_{m-1}) + \frac{\int_{t_{m-1}}^{t_m} C(t) \, dt}{\sum_{k \in B(t_{m-1})} r_k} 
\]

Meanwhile, the virtual time in V-WFQ evolves as follows under link speed \(C(t)\)

\[
v(t_m) = v(t_{m-1}) + \frac{\int_{t_{m-1}}^{t_m} C(t) \, dt}{\sum_{k \in B(t_{m-1})} r_k} 
\]

Since the virtual finish function in V-WFQ and a-virtual time function in PCAWFQ are identical and (20) and (19) show that the reference virtual time of AC flow packets equals the virtual finish time in V-WFQ, Theorem 2 follows \(\square\)
Parekh [72] computed the delay bound in a WFQ server. According to it, if flow $k$ is constrained by leaky-bucket filter and $r_k \geq \rho_k$, the delay bound of the flow $k$, $D_k$ is

$$D_k \leq \frac{Q_k}{r_k} + \frac{L_{\text{max}}}{C}$$

where $L_{\text{max}}$ represents the maximal packet size among all flows and $C$ is link speed.

The result is very straightforward. The first term $\sigma_k/r_k$ of the delay bound represents the maximal period to clear the backlogged bits of flow $k$ at the guaranteed rate $r_k$. The following inequality shows that the maximal backlogged bits of flow $k$, $Q_k^*$ is not bigger than the burstiness of flow $k$, $\sigma_k$.

$$Q_k^* = \text{arrival amount} - \text{served amount} \leq (\sigma_k + \rho_k(t - \tau)) - r_k(t - \tau) \leq (\sigma_k + \rho_k(t - \tau) - \rho_k(t - \tau) = \sigma_k$$

The second term $L_{\text{max}}/C$ represents the service period of the maximal packet. Since packet-by-packet discipline cannot interrupt the current service, a newly arrived packet should wait until the service finish point of the current served packet even if the arriving packet has an earlier finish time.

The following Theorem shows the delay bound of an AC flow in a semi-stable PCAWFQ.

**Theorem 3:** In a semi-stable PCAWFQ server with a varying serving rate $C(t)$, if AC flow $k$ is constrained by the leaky-bucket model and is locally stable i.e. $r_k \geq \rho_o$ then regardless of the traffic characteristics of other flows, the delay of the flow $k$ satisfies the following bounds

$$D_k \leq \frac{O_k}{r_k} + \frac{L_{\text{max}}}{\min_{t>0} \{C(t)\}}$$

where $r_k$ is share rate of flow $k$ and $L_{\text{max}}$ is the largest packet size among AC and WC flows.

**Proof:** Let $p_k^i$ and $L_k^i$ be the $i^{th}$ packet of $k^{th}$ flow and its packet size. Let $F_k^i$ and $\hat{F}_k^i$ be the service finish time of $p_k^i$ under CAWFQ and PCAWFQ, respectively. Let $T(L)$ be the time duration to finish the service of $L$ size packet under PCAWFQ. Let $s_k^i = F_k^i - T(L_k^i)$.

Since CAWFQ is a work-conserving discipline, PCAWFQ and CAWFQ have identical backlogged periods. However, the finish time of each packet may be different owing to the packet-by-packet scheme of PCAWFQ. $p_k^i$ should start to be served at $s_k^i$ in PCAWFQ in order that it has the same finish time with the equivalent CAWFQ. However, if another packet may be being served at $s_k^i$, then it should wait until the finish time of the current service. Therefore, the service for $p_k^i$ may be delayed up to $F_k^i + T(L_{\text{max}})$, so we have

$$\hat{F}_k^i \leq F_k^i + T(L_{\text{max}})$$

Since $T(L_{\text{max}}) \leq L_{\text{max}}/\min_{t>0} \{C(t)\}$,
According to the leaky-bucket definition, the arrival function of flow $k$, 
$A_k(t) \leq \sigma_k + \rho_k(t - \tau)$

In a semi-stable CAWFQ server, if flow $k$ starts to be backlogged at $\tau$, then during any period $(\tau, t)$,
$S_k(\tau, t) \geq r_k \cdot (t - \tau)$ where $S_k$ is the served amount of flow $k$

Therefore, the backlogged bits of flow $k$ in the server at $t$, $Q_k(t)$ is
$Q_k(t) = A_k(t) - S(\tau, t) \leq \sigma_k + \rho_k(t - \tau) - r_k(t - \tau) \leq \sigma_k + \rho_k(t - \tau) - \rho_k(t - \tau) = \sigma_k$

Meanwhile, the packet $p_k^t$, arriving at $t$ in the CAWFQ server, should wait until clearing $Q_k(t)$ before it is served. In the semi-stable CAWFQ server, $Q_k$ is cleared at least at the rate of $r_k$. So the delay bound in the semi-stable CAWFQ server is
$D_k^F \leq \frac{Q_k(t)}{r_k} \leq \frac{\sigma_k}{r_k}$

According to (34), the service duration time of any packet of flow $k$ in PCAWFQ server is at most $L_{\max} / \min_{t \in 0 \{C(t)\}}$ later than that in the CAWFQ server. Therefore we conclude that the delay bound in PCAWFQ server is
$D_k^F \leq \frac{\sigma_k}{r_k} + \frac{L_{\max}}{\min_{t \in 0 \{C(t)\}}}$

5.6 Demonstration

We simulated PCAWFQ to verify its performance using the OPNET simulator. The simulation experiments are based around the topology shown in Figure 5.9. A BS (Base Station) and 5 workstations (tagged src in the Figure 5.9) are attached to the same 10Mbits/s Ethernet segment and the BS communicates with 5 MHs (Mobile Host) via 2Mbits/s wireless link to connect the workstations and MHs. Since we are interested in the rate-based bandwidth service in a BS, the network delay in the Ethernet segment is negligible even though it may affect the delays of each packets of flows. Thus, we ignore the delay in the Ethernet segment in the following simulations. The PCAWFQ scheduler is installed over the MAC layer in the BS. We set up 5 downlink flows between workstations and MHs for the workstation $i$ to transmits UDP packets to the MH $i$, where $i = 1, 2, ..., 5$. The fifth pair (src5 - MH5) is used to make the wireless link-rate fluctuate in such a way that the BS serves the packets generated from the 5th source with the highest priority. This is similar to CDPD [21] where data connections can use only the remaining
bandwidth after the analogue voice connections are served. As a result, the available bandwidth to the other 4 flows varies according to the traffic pattern of the 5th source. To compare the services received by the 4 flows easily, all 4 flows have the same packet size (8656 bits), reserved bandwidth (400 kbits/s) and fixed packet sending rate (865.6 kbits/s) in the simulations, but two flows out of the 4 flows are assigned to the WC and the other 2 flows to AC.

The 5th source transmits UDP packets with exponentially distributed burst, so the available bandwidth is also exponentially distributed. We simulated four cases: a semi-deficit, deficit, surplus membership-step period and a backlogged period that includes multiple membership-step periods. The four simulation cases are made by adjusting the average data-sending rate of the 5th source as follow:

<table>
<thead>
<tr>
<th>Simulation case</th>
<th>Data rate of the 5th sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>deficit period</td>
<td>557 Kbits/s</td>
</tr>
<tr>
<td>semi-deficit period</td>
<td>1.134 Mbits/s</td>
</tr>
<tr>
<td>surplus period</td>
<td>1.784 Mbits/s</td>
</tr>
<tr>
<td>backlogged period</td>
<td>1.134 Mbits/s</td>
</tr>
</tbody>
</table>

![Diagram](image)

**Figure 5.9 The network configuration for the simulation**

Figures 5.10 - 5.13 show the four simulation results. In the figures, the service time represents the virtual finish time of the current served packet. In Figure 5.10 (a), a-virtual time is the upper bound curve and w-virtual time is the lower bound curve. The virtual finish time of the served packet plots an oscillating curve which is bounded by a-virtual time and w-virtual time. As can be seen in Figure 5.10 (a), the a-virtual time advances with average gradient of 1 and w-virtual time less than 1, during the semi-deficit period as we expected. The gradient of each virtual time measures how much bandwidth each class has received compared with its allocated bandwidth. The gradient (with average value 1) of a-virtual time shows that the AC flows
received just its allocated bandwidth while the WC flows received less than its allocated bandwidth. Figure 5.10(a) also indicates that the AC and WC flows can barely be served between 0.8 and 0.9 sec due to the bandwidth shortage. However, the AC flows received the compensated service (higher than usual rate) just after the temporary shortage period, whereas the WC flows experienced a longer no-service period owing to the compensated service to the AC flows. Note that this is the case of a membership-step period where the membership does not change and the compensation scheme works. The amplitude of the oscillating curve represents an instantaneous delay gap $\delta(t) - \zeta(t)$. Whenever the class of the served packet changes between AC and WC, the service time oscillates due to the difference of $a$-virtual time and $w$-virtual time. Figure 5.10 (b) shows the actual received services of the four flows by plotting packet numbers received by each MH.

Figure 5.11 shows that the WC flows cannot receive any service in a deficit membership-step period since the available bandwidth is insufficient even to serve AC flows. All available bandwidth is used to serve the AC flows in this period, so the $w$-virtual time does not advance at all. The 0.5 gradient of the $a$-virtual time in Figure 5.11 (a) shows that the AC flows receive only half of allocated bandwidth. Figure 5.12 shows the case of a surplus period. All flows receive more than their allocated bandwidth, sharing the available bandwidth in a fair manner irrespective of service class.

Figure 5.13 shows the simulation results over a backlogged period that includes multiple membership-step periods, which usually occurs when AC flows are under-loaded. To make the scenario, the 2 AC flows and 2 WC flows transmit UDP packets at the same rate of their allocated bandwidth (400Kbits/s respectively) under the variable-rate link of which average rate is 1.134Mbits/s. In this case, the available bandwidth is sufficient to support the AC flows, whereas it is not sufficient to serve the WC flows. Thus, AC flows are in and out of backlogged status, while packets of WC flows are always backlogged. It results in $w$-membership-step periods and mixed-membership-step periods appearing alternately. In Figure 5.13 (a), we indicate the start and end point of a $w$-membership-step period. Although the graph shows a falling curve of $a$-virtual time during the $w$-membership-step period, it is made because the plotter makes lines between every points. Actually, the $a$-virtual time does not exist during the period. As we explained earlier and can be seen, $a$-virtual time equals the $w$-virtual time when a mixed-membership-step period begins immediately following the $w$-membership-step period. Note that the $a$- or $w$-virtual times are on increase only during $a$- or $w$-backlogged period. In Figure 5.13 (a), the decreasing point of the $a$-virtual time is when an $a$-backlogged period ends. We can also see that the service time plots an oscillating curve bounded by the $a$-virtual time and $w$-virtual
time. Figure 5.13 (b) shows that the AC flows received their allocated bandwidth while WC flows do not.

![Graph showing virtual times and service time](image1)

(a) virtual times and service time

![Graph showing received packet numbers](image2)

(b) received packet numbers

Figure 5.10 Simulation results for a semi-deficit period

![Graph showing virtual times and service time](image3)

(a) virtual times and service time

![Graph showing received packet numbers](image4)

(b) received packet numbers

Figure 5.11 Simulation results in a deficit period
Figure 5.12 Simulation results in a surplus period

Figure 5.13 Simulation results for a backlogged period
5.7 Discussion

5.7.1 Functionality

Lee [60] investigated the performance bounds for an isolated variable-rate communication link in both the deterministic and statistical approaches. The analysis attempts to determine packet delay under a simplest scheduling discipline, FCFS (First-Come-First-Serve) queuing. As introduced in Chapter 3, Goyal [43] has proposed SFQ (Start-time Fair Queuing) that is another packetised version of a GPS server and analysed its performance bounds under a variable rate.
link, showing that the performance bounds vary as the available bandwidth fluctuates. Therefore, the discipline cannot be applicable to the flows that require a constant bit rate or a minimum guaranteed bit rate, whereas the CAWFQ can guarantee a constant bit rate for AC flows if we can expect the minimum bandwidth of a variable-rate link.

In the rate-based schedulers [71], [30], [43], [39], [9], [85], [101], [55], the servers distribute bandwidth to flows according to their respective share-rates, which results in each flow receiving a guaranteed fixed bandwidth in the case of a fixed-rate link. However those rate-based schedulers are unable to make similar guarantees in the case of a variable-rate link. A priority based scheduling could be a solution for a variable-rate link because it can serve the delay sensitive flow first. However, the more a flow is greedy, the more the flow receives service since it cannot limit the service rate received by a flow: i.e. it is not a fair discipline. By contrast, CAWFQ can support prioritised class service, maintaining rate-based services. In addition, CAWFQ can limit the bandwidth used by the higher class (absolute class) so that the higher class cannot monopolies the link, whereas the priority-based method cannot limit the bandwidth for high priority flows: i.e. CAWFQ maintains fairness between classes. The hierarchical schemes [33], [8] can support class based queuing, but they have the same limitation as rate-based schedulers have since they adopt a rate-based queuing for their node schedulers. CAWFQ is a kind of mixed discipline of priority-based and rate-based queuing. Between classes, the server uses a priority-based scheduling while it uses rate-based queuing within a class.

5.7.2 Complexity

PCAWFQ has the same complexity of \(O(V)\) such as WFQ, where \(V\) represents the number of backlogged flows. For a high speed network or backbone network that requires fast scheduling, the high complexity of \(O(V)\) is a critical problem since those network should manages many flows. Therefore, fast scheduling algorithms with low complexity of \(O(\log V)\) or \(O(1)\) have emerged at the cost of performance reduction. Virtual Clock and SCFQ are in this category. However, a low bandwidth network such as a cellular data network can endure the computational burden increasing in proportion to the number of flows since it establishes only limited flows due to low bandwidth as well as relatively ample time period for computation.

5.8 Summary

We have investigated the unfairness problem of WFQ under a variable-rate link. Based on the investigation, we proposed V-WFQ that removes the unfairness so that we can compare our proposed scheme with the V-WFQ. We proposed a scheduling algorithm, PCAWFQ which
distributes bandwidth to a flow according to the flow's associated class and share rate. The allocated bandwidth of AC flows can be guaranteed even in a variable link provided that available bandwidth is higher than the total bandwidth allocated to the AC flows in a membership-step period. The residual bandwidth is exploited by WC flows. The best-effort class can exploit any remaining bandwidth following servicing of AC and WC flows. This approach ensures optimal efficiency of bandwidth usage. The simulation results demonstrate the performance of the proposed scheduler and contrast the functions of CAWFQ with those of WFQ or V-WFQ. These results highlight the usefulness of the proposed scheme in supporting flows requiring diverse QoS in a variable-rate link.

We conclude that CAWFQ can remedy the deficiencies of the rate-based queuing in a variable-rate link and has desirable characteristics with regard to supporting integrated traffic in a packet network with varying-rate bandwidth.
Chapter 6

Fair Queuing Multiple Access (FQMA)

In this Chapter, we propose a slot based multiple access protocol called FQMA, which is used for the uplink traffic service in our proposed service architecture explained in Chapter 4. It handles periodic and random traffic keeping the de-coupling between both. The de-coupling allows random traffic to exploit only remaining slots following servicing of the periodic traffic. In a multiple access protocol that can support integrated traffic services, the de-coupling functionality plays an important role because it prevents delay tolerant packets (random traffic) from impeding prompt service of delay intolerable packets (periodic traffic). Since FQMA guides an uplink slot by a polling signal such as C-PRMA and I-ISMA, it is suitable for micro- and pico-cellular networks. In this Chapter, we will use several terms defined in Chapter 4 and 5.

6.1 FQMA protocol description

FQMA is a kind of the demand assignment protocol with central control such as [13], [54], [4]. In this type of protocol, a mobile host should contend to obtain a reservation through a separated reservation request slot, whereas it can transmit data without contention by guidance of base station once admitted. An assigned reservation becomes invalid at the end of a backlogged period, so a mobile host should request a reservation at the start points of every backlogged period.

6.1.1 Issues to be considered

For conveying uplink traffic data, FQMA operates during the uplink slot periods determined by the SAC (Slot Allocation Controller – refer to Chapter 4). The uplink slots are assigned to either periodic traffic or random traffic by STS (Slot Type Selector) ensuring that random traffic does not affect the performance of the periodic traffic service, and the slots assigned to each traffic are used for two purposes: data transmission and reservation requests. Consequently, FQMA classifies uplink slots into four types: DP (Data transmission slot for Periodic traffic), RP (Reservation request slot for Periodic traffic), DR (Data transmission slot for Random traffic) and RR (Reservation request slot for Random traffic). Figure 6.1 shows examples of uplink slot assignment according to the slot types under the two SAC models. Thus, there are four issues we should resolve in designing the FQMA.
1. How should we distribute uplink slots between periodic and random traffic, de-coupling both streams such that random traffic does not degrade the quality of periodic traffic service?

2. Once uplink slots are allocated to each traffic class, how should we control or set the issuing rates of reservation request slots for each traffic class?

3. How should we schedule periodic or random traffic during its assigned data transmission slots?

4. How does a mobile host request a reservation to a base station through the reservation request slots?

The fourth issue has been extensively researched in the literatures [13], [4], [83]. Therefore, we just choose and use a reservation request scheme that is suitable for our proposed protocol, whereas we develop an uplink scheduler to resolve the other three issues in this dissertation.

![Uplink slot formats](image)

**Figure 6.1 Examples of uplink slot assignments**

**6.1.2 Uplink slot formats**

Figure 6.2 shows the formats of the four types of uplink slots. As explained in Chapter 4, an uplink slot follows USB signal. An ID subslot, which indicates the owner of the uplink slot, positions at the beginning of an uplink slot. A base station informs mobile hosts of their allocated uplink slots using this ID subslot. ID = 0 indicates it is a reservation request slot for periodic
traffic and ID = 1 for random traffic. As can be seen in Figure 6.2, both traffic classes have the same uplink slot format except the format of RS (Request Subslot) that is a subslot of a reservation request slot. It is because both traffic has different requirements for slot reservation. A reservation slot consists of several RSs and ACKs (acknowledgement subslots), and the RSs are composed of several fields as shown in Figure 6.2 (b) and (c). The number of RS subslots in a reservation slot is a design factor chosen depending on the slot length and the maximum expected population of mobile hosts in a cell.

USB : Uplink-slot Start Beacon
ID : Mobile host Identifier
ACK : Acknowledgement
RS : Request Subslot
W : Waiting time measured in slots
T : Packet generation interval in slots
U : Lifetime in slots

Figure 6.2 Uplink slot formats
6.2 Reservation Phase

For slot reservation requests, we use the out-Slot scheme [4] as C-CRA and C-CPRMA do, in which a separate slot is subdivided into several subslots for reservation requests. To indicate whether a request has arrived successfully without conflict, an out-Slot based protocol acknowledges the reservation request at the beginning of the downlink slot following the request slot. In a wireless link where uplink and downlink use different frequencies and are synchronized, the mobile host can receive an acknowledgment during the following downlink slot. However, in our uplink/downlink-sharing scheme, it is not the case: i.e. multiple reservation slots may be broadcast without any downlink slot. It results in a flow, which has been admitted in the previous reservation slot, requesting a reservation again in the following reservation slot since it has not received the acknowledgment. Thus, in our protocol, acknowledgements are broadcast during the same uplink slot following the RSs as depicted in Figure 6.2 (b) and (c) in order to prevent the duplicated request phenomenon. The common channel model adopted in FQMA enables a base station to send base-to-mobile acknowledgements even during an uplink slot period.

6.2.1 Reservation request of periodic traffic flows

Due to the different characteristic between periodic and random traffic, reservation phases for both in FQMA are designed slightly differently. When a base station broadcasts an RP slot, a mobile host that has a periodic source starting a new talkspurt\(^4\) requests a reservation to the base station during a randomly chosen RS subslot by transmitting ID, \(W\), \(T\) and \(U\) parameters. The parameters contain information about the flows as follows:

- **ID**: mobile host ID
- **\(W\)**: waiting time in slots of the packet\(^5\) at the head of the queue from its generation time
- **\(T\)**: packet generation interval time in slots
- **\(U\)**: packet lifetime in slots.

Upon measuring the parameters, downlink slots are also counted. The uplink scheduler uses the parameters to estimate expected packet generation time of each source. If the request has been successfully received without channel conflict, the base station replies in the corresponding ACK subslot (ACK \(n\) for RS \(n\)) with the ID of the mobile host as an acknowledgement. The mobile station

\(^4\) The typical example of periodic traffic is a speech source, so we call a backlogged period in a periodic source a talkspurt in this dissertation.

\(^5\) This may not be the first packet of the talkspurt if packets have been dropped for exceeding their lifetimes.
host that has not received the ACK message sends a request again in the next reservation slot. The issuing rate of RP slots varies according to the periodic traffic load.

### 6.2.2 Control of the RP slot issuing rate

During over-loaded periods, packet drops are likely to occur in an uplink periodic flow. There are two causes of the packet drops. Firstly it may occur upon delayed RP slot broadcast or repeated request conflicts. In this case the packet drops occur from the beginning of a talkspurt. Secondly, after a flow is admitted, it may occur because the base station cannot serve all packets within their lifetimes due to congestion. During such over-loaded periods, every uplink slot should be used only for conveying data rather than reservation requests to maximise channel utilization. Issuing RP slots should be delayed until the periodic traffic is under-loaded because it just increases the packet drop rates even though it may admit additional flows.

Based on the number of admitted periodic flows and their packet generation rates, we can compute whether the link is over-loaded or not. Using the computed load, the issuing rate of RP slots should be controlled in the inverse proportion to the periodic traffic load. Thus, a RP slot is not broadcast at all during over-loaded periods and the interval of RP slots varies according to the periodic traffic load. But there is a minimum-issuing interval, so RP slots are issued at a constant interval during low traffic load. Note that the computed load only includes the admitted flows. The whole load could be larger than that because more flows may start to be backlogged in the next slot or have failed to contend to obtain the reservation. Apparently, these additional loads should be considered for change issuing rate. We estimate it using the request conflicts. We will show the detailed algorithm for controlling the RP slot issuing rate in §6.4.4.

### 6.2.3 Reservation request of random traffic flows

A random traffic flow uses the same reservation procedure except that it sends different information to the base station as indicated in Figure 6.2 (c). Note that random traffic also includes the best-effort class traffic as we defined in §4.2. Unlike the periodic traffic, it is difficult to estimate the packet generation time of random traffic. Although they provide their average packet generation rate (share rate) that may be used for load estimation, it is inappropriate to use it for computing slot-level load since random traffic loads are likely to be dynamic. Thus, we do not adopt the dynamic slot rate scheme used for periodic traffic. Instead, RR slots are issued at a fixed rate within the slot period allocated to random traffic, so the practical issuing rate varies and it depends on the available slots for random traffic.
6.3 Proxy Packet Generator

Whenever an uplink flow begins a new backlogged period, its MH requests a slot reservation for the service during the backlogged period. When a flow successfully added, the PPG (Proxy Packet Generator) creates a proxy packet for it, which contains the information transmitted from the mobile host and represents the real packet at the head of the MH queue. As can be seen in Figure 4.4, the created proxy packet is delivered to the uplink scheduler, and is queued in either PEDQ or PCAWFQ' depending on the traffic class of the flow – PC (Periodic Class) packets to PEDQ and other classes packets to PCAWFQ'. Thus, the PEDQ or PCAWFQ' serves the proxy packet as if it is a real packet. The STS (Slot Type Selector) decides which one of both the queues owns the next uplink slot, and the assigned queue serves a proxy packet at its head and hands it to SAC (Slot Allocation Controller). Then SAC allocates a DP slot or DR slot depending on the queue so that the corresponding real packet of the proxy packet can be conveyed to the BS. After being served, the proxy packet is re-queued in the queue where it was, representing the next real packet at the MH queue. Thus, an uplink flow being served has only a proxy packet, which represents the next real packet to be served, in either of both the queues.

6.4 Uplink scheduler

In FQMA, an uplink scheduler in a base station plays a significant role to achieve our proposed class service in the uplink, resolving the three design issues 1 - 3 pointed in §6.1.1. An uplink scheduler consists of STS (Slot Type Selector) and two queues for periodic and random traffic (PEDQ and PCAWFQ') as shown in Figure 4.4. Using the parameters from reservation requests, PEDQ determines the service order of periodic traffic flows and PCAWFQ' random traffic flows. The STS determines the type of an uplink slot and controls the issuing rate of the reservation request slot for periodic traffic. To summarize, the two queues resolve the third design issue and the STS the first and second design issues. We will present those components in the following sections.

6.4.1 PEDQ (Packet Endurable Delay Queuing)

PEDQ schedules proxy packets of periodic sources according to the remaining lifetimes of their corresponding real packets which can be computed using the parameters delivered from MHs upon a request, and of the 4 flow parameters, W changes in each request while the others keep the same value during the whole period. When a flow is admitted, the created proxy packet for the flow represents the packet at the head of the MH's queue, which may not be the first
generated packet in the talkspurt as the packet exceeding its lifetime is removed. Let \( p'_k, i = 0,1,2,\ldots,n \) be the sequential real packets starting from the packet at the head of the \( k \)th admitted flow's queue. Let \( VP'_k \) be the proxy packet representing \( p'_k \) and \( MET'_k \) be the Maximum Endurable Time of \( p'_k \) at the time \( t_k \) where \( t_k \) is the moment that the flow \( k \) is admitted. The MET represents the maximum period of time that a real packet keeps valid without being dropped when it is measured: i.e. it is the remaining lifetime of a packet at the time \( t_k \). When a proxy packet \( VP'_k \) arrives, the PEDQ calculates \( MET'_k \) as follows:

\[
MET'_k = \begin{cases} 
U_k - W_k, & i = 0 \\
MET_{k}^{i-1} + T_k, & i = 1,2,3,\ldots,n.
\end{cases}
\]

where \( T_k, U_k \) and \( W_k \) represents the packet generation interval, the packet lifetime and the waiting time of the flow \( k \). Note that \( W_k < U_k \) for all \( k \) since an MH drops packets that exceed their lifetimes.

![Diagram of MET and PDT](image)

**Figure 6.3 Comparison between MET (Maximum Endurable Time) and PDT (Packet Drop Time)**

To schedule proxy packets in increasing order of the remaining lifetime, each flow's \( METs \) should be compared each other to decide service order. However \( METs \) of different flows cannot be compared each other directly since each flow's \( MET \) is computed based on the admitted time
of the flow. Figure 6.3 (a) shows one example of the situation. For instance, \( MET_i \) in the Figure 6.3 (a) is not correct remaining lifetime of \( p_i \) at \( t_i \) since it is measured at \( t_i \). Thus, we introduce a PDT (Packet Drop Time) to transform MET into a single referenced time. PDT is computed based on a single referenced time, SL (Slot Counter), and is defined as follows:

\[
PDT_k^i = \begin{cases} 
MET_k^i + SL(t_k), & i = 0 \\
PDT_k^{i-1} + T_k, & i = 1, 2, 3, ..., n 
\end{cases}
\]

where \( t_k \) indicates the time when the \( k^{th} \) flow is admitted. \( SL(t) \) is clocked by each occurrence of a slot (including uplink and downlink slots) and is reset to zero whenever the PEDQ queue is empty. As can be seen in Figure 6.3 (b), as the PDT of a packet is computed based on the slot counter at which the packet’s flow is admitted, a packet’s PDT represents the slot counter that the packet’s lifetime runs out. Consequently, we can use PDT as a remaining lifetime index to decide packet service order among different flows. PEDQ tags \( PDT_k^i \) on \( VP_k \) and serves the proxy packets in increasing order of PDTs.

### 6.4.2 Re-queuing to PEDQ

After a proxy packet is served, it is re-queued in the PEDQ to serve next real packet as long as the reservation is valid. There are two routes of re-queuing as indicated in Figure 4.4. One is from STS and the other is from SAC.

**From STS (Slot Type Selector):** When STS allocates the current uplink slot to periodic traffic, PEDQ hands the proxy packet at its head to the STS. Then the STS judges whether the corresponding real packet of the proxy packet has been dropped in the mobile host due to the expiration of the lifetime by comparing PDT of the packet and current SL. If it has been dropped: i.e. \( PDT \leq SL \), then the STS re-queues the proxy packet instead of sending it to SAC. The re-queued proxy packet represents the next real packet of the dropped real packet. The STS repeats the same procedure until finding an undropped real packet. If found, it sends the corresponding proxy packet of the found one to SAC.

**From SAC (Slot Allocation Controller):** Once SAC receives a proxy packet, it assigns the current uplink slot to the mobile host owning the proxy packet. A proxy packet functions as a ticket to obtain an uplink slot for its corresponding real packet. The proxy packet given to the SAC is not destroyed even after the slot allocation. It waits for the end of the uplink slot, and then the proxy packet is re-queued to PEDQ to serve the next real packet if the mobile host has sent a data packet during the allocated slot. However, the proxy packet is removed if the mobile host has not responded to the allocated slot. The
proxy packet removal means that the flow loses its reservation. So the flow should request a reservation again when the next talkspurt starts.

When a proxy packet is re-queued, the PEDQ updates \( PDT \) of the proxy packet as follows so that the re-queued proxy packet is tagged with the next real packet’s \( PDT \):

\[
PDT_i = PDT_{i-1} + T_k \quad \text{where } i = 1,2,3 \ldots n
\]

The re-queued proxy packet is served based on the new \( PDT \).

### 6.4.3 PCAWFQ’ (Packet-by-packet Combined Absolute & Weighed Fair Queuing’)

A random traffic source generates packets irregularly unlike a periodic traffic source, so the PEDQ that uses the periodic feature of flows is not suitable to schedule the uplink random traffic. To schedule the uplink random traffic, we propose PCAWFQ’ scheme that is a modified version of the PCAWFQ and provides uplink random flows with the class based service as the PCAWFQ does. It has the same queuing discipline as PCAWFQ except for the use of proxy packet. In PCAWFQ, a packet is queued as it arrives, but in PCAWFQ’ it is not possible since the real packets are queued in the mobile hosts. As PCAWFQ’ cannot access the arrival times of packets, which is necessary information for most queuing disciplines, it uses proxy packets to emulate packet arrivals.

When an uplink random flow is admitted, the PPG (Proxy Packet Generator) generates a proxy packet for the flow and inputs it to the PCAWFQ’. Then PCAWFQ’ regards it as a packet arrival and performs as PCAWFQ. When a DR slot is assigned to PCAWFQ’, it picks a proxy packet according to the PCAWFQ discipline and sends it to the STS. Unlike the treatment to the PEDQ packets, the STS just hands it to SAC without considering the re-queuing since the random traffic packets do not have lifetimes.

A proxy packet of PCAWFQ’ arriving at the SAC is also re-queued in the same way as PEDQ’s proxy packets. That is, if the mobile host does not respond during the allocated slot, the proxy packet is removed. If not, the proxy packet is re-queued to the PCAWFQ’ representing the next real packet. The proxy packet re-queuing emulates the real packet arrivals during a backlogged period for PCAWFQ’. Consequently, the only difference between PCAWFQ’ and PCAWFQ is in the packet arrival times. Apart from using the proxy packet arrival times, it works in the same way as PCAWFQ.

In rate-based queuings, including PCAWFQ, the bandwidth guarantee is independently of the packet arrival time. A rate-based queuing guarantees the share rate of a flow as long as the
flow is backlogged. Thus, the arrival times emulated by proxy packets enable the PCAWFQ to guarantee the same share rate of flows as PCAWFQ. However, PCAWFQ cannot guarantee the allocated bandwidth only after a backlogged flow is admitted, while PCAWFQ guarantees it from the moment that it starts to be backlogged.

In addition, PCAWFQ cannot provide delay guarantees. There are two reasons for this. First of all, when a flow starts to be backlogged, PCAWFQ does not know it until the reservation request arrives successfully at the base station. The delay time between the backlog start and admitted time is dependent on the current traffic loads of the system and reservation scheme, and is not bounded deterministically. Secondly, the artificially generated arrival time prevents PCAWFQ from guaranteeing the delay bound.

6.4.4 Slot Type Selector (STS)

The STS determines a slot type and also decides whether the proxy packet received from PEDQ should be re-queued or not. In the case of re-queuing, it returns the proxy packet to the PEDQ. The proxy packet from PEDQ or PCAWFQ is passed to the SAC when STS assigns a DP or DR slot. When STS assigns an RP or RR slot, it gives a RP or RR slot signal to SAC instead of a proxy packet.

STS refers to the Redundant Slot Time (RST) derived from PEDQ to figure out the amounts of redundant slots which can be allocated to PCAWFQ without affecting the performance of the periodic traffic service. Let $RST_p$ be the Redundant Slot Time of $p^{th}$ placed proxy packet in the PEDQ. Then $RST_p$ is defined as follow:

$$RST_p(t) = PDT_p - (SL(t) + p), \quad p = 0, 1, 2, \ldots$$

where $PDT_p$ represents the PDT time of the $p^{th}$ proxy packet.

$RST_p(t)$ represents the number of redundant slots of the system at the time $t$ in servicing the $p^{th}$ queued proxy packet providing that no additional flows join and leave afterward. Thus, the $p^{th}$ proxy packet will be served without packet drop even though the system allocates the redundant slots to RP, DR, or RR slots. $RST$ is similar to $PDT$ in terms of representing the endurable delay of individual proxy packet. However, $RST$ indicates the number of extra slots that a proxy packet can endures taking into account the service of all queued proxy packets ahead of it, whereas $PDT$ shows the absolute slot position that a proxy packet becomes invalid. Consequently, $RST$ varies depending on the queue load while $PDT$ is a fixed value when admitted.

For example, assume that the $PDT$ of a proxy packet is 110 at slot counter 100 and there are 8 proxy packets queued ahead of the proxy packet. Then the proxy packet can live until slot
counter 110 and the $RST$ of the proxy packet is 2 at the slot counter 100 since it has 2 extra slots after the 8 proxy packets ahead are served. The service for the proxy packet is not affected even if the extra two slots are used for other queue services. The $RST$ does not change as long as new proxy packets are not inserted ahead of the proxy packet, but if a proxy packet is newly queued ahead of the proxy packet, then the $RST$ reduces to 1 while $PDT$ still remains unchanged.

In addition to individual $RST$, the smallest $RST$ in a PEDQ - we define it as $\lambda$ - is also meaningful to the system to minimize packet drop since the $RST$ for each proxy packet varies. Thus we define $\lambda$ as the follow:

$$\lambda(t) = \begin{cases} \min_{p \in Z}(RST_p(t)) & Z = \{1,2,\ldots, j\}, j \geq 1 \\ \infty & j = 0 \end{cases}$$

where $j$ is the queue size.

$\lambda(t)$ represents the maximum amount of redundant slots at the time $t$ that can be used for RP, DR or RR slots without any packet drop. $\lambda(t)<0$ means that packet drops will occur from $t$ onwards. Figure 6.4 shows the flowchart that describes how the STS determines a slot type at the beginning of an uplink slot using $\lambda$ and $RST$. The following shows the criterion for each step in the flowchart.

**Criterion 1:** $\lambda(t_c) > n$

where $t_c$ indicates the current time and $n$ is a design factor determined depending on the number of RS subslots per RP slot.

This judges whether the traffic loads of flows queued in PEDQ are not-overloaded such that a RP slot can be broadcast. If there was a conflict in the previous RP slot or it is time to broadcast a RP slot based on the minimum RP slot interval, a RP slot should be broadcast at the current slot. According to the analysis in §6.6.2, however, it is not beneficial to broadcast a RP slot during an over-loaded period regardless of the previous conflict status or RP slot interval. $n$ determines the traffic load to accept a RP slot broadcast. Ideally $n$ should be set by 0. But in the multiple request strategy, a reservation slot may accept multiple successful requests, which number relies on the RS subslot number $m$ in a reservation slot. If multiple flows are admitted, the total admitted traffic load might become over-loaded. To avoid the situation, we adjust $n$ depending on the number of $m$. Therefore, $n$ is a design factor chosen based on $m$. 

Figure 6.4 Slot type decision flowchart

Criterion 2: $SL(t_s) - SL(t_c) \geq \begin{cases} \sum_{j=1}^{\infty} \frac{T_p}{j}, & j > 0 \\ T_m, & j = 0 \end{cases}$

where $t_c$ indicates when the previous RP slot is issued, $T_p$ represents $T$ period of $p^{th}$ placed packet in the PEDQ and $j$ is the queue size. $T_m$ represents the default interval for RP slot broadcast which determines the minimum-issuing interval as explained in §6.2.2.

This criterion determines the time interval for RP slot broadcast. The right upper hand term represents the average packet generation interval of the queued flows.
**Criterion 3:** $\lambda(t_s) \leq 0$

This examines whether the traffic loads of the flows queued in PEDQ are over-loaded or not. If it is over-loaded, STS proceeds to issue a DP slot. When criterion 3 meets the condition $\lambda(t_s) < 0$, it shows that there will be at least a packet drop among all flows queued in PEDQ onward provided that no additional flow join and leave. When it meets the condition $\lambda = 0$, there will not be a packet drop. However, if the current slot is not assigned to the DP slot, a packet drop will occur. Namely, they mean that STS should issue a DP slot to avoid packet drops. Criterion 3 prevents the STS from issuing the RP, RR or DR slots in an over-loaded period. Note the difference between criterion 1 and 3.

**Criterion 4:** $RST_{o}(t_s) \leq 0$

Criterion 4 examines whether the corresponding real packet of the proxy packet at the head of PEDQ has been already dropped in the mobile host. If it has been, then STS returns the proxy packet to PEDQ for re-queuing. Following the re-queuing, PEDQ sends the next proxy packet to STS. If STS just assigns a DP slot to the proxy packet without the re-queuing under that situation, the slot allocated to the dropped packet will be used by the next packet queued in the mobile host, which has later PDT than the dropped packet. It is not desirable since there are packets with earlier PDT in the queue than the next queued packet. This slot misuse results in increasing the packet drop rate. Thus, in that case, the proxy packet should be re-queued so that the earlier PDT tagged packet can be served first.

**Criterion 5:** $RST_{o}(t_s) < U_0$

where $U_0$ represents $U$ period of the packet queued at the head of the PEDQ.

This criterion examines whether the corresponding real packet of the proxy packet at the head of PEDQ is already generated or not. If a DP slot is assigned to the packet that has not been generated, the issued DP slot is not utilised. At a low-loaded condition, the packet generation rates of periodic sources may lag behind the DP slot supplying speed for the sources. Consequently, many slots may be wasted without being used in such a low-loaded condition. The wasted slot should be allocated to random traffic to maximise the channel utilisation. It is certain that allocating the extra slots to random traffic does not affect the packet drop rate of periodic traffic. Therefore, criterion 5 enables a periodic flow to be served as long as it has a packet to send without slot wasting, allowing random traffic to exploit the extra slots. In this criterion, we may use 0 instead of $U_p$ to maximise the slots allocated to random traffic. It certainly does not
affect the performance of the periodic traffic service though it keeps the RSTs of all queued packets to their minimum. In this case, however, a problem arises when the periodic traffic changes into over-loaded. Since all RSTs remains minimum, packets start to drop as soon as the traffic changes into over-loaded. In contrast, if we keep the RST to its maximum, the packet drop is delayed and especially may not occur when an over-loaded period is short enough to endure. In this case, the maximised RST functions as a bandwidth buffer. Therefore, we keep RST to its maximum using $U$. The effect of maximising $RST$ will be explained in detail in the next section.

**Criterion 6:** $ASR(t) - ASR(tr) > d$

where $ASR(t)$ represents the number of slots allocated to random traffic up to the time $t$ and $tr$ indicates the time when the previous RR slot is issued. $d$ is an issuing interval of RR slots.

This determines the issuing interval of RR slots. The interval $d$ is a fixed value and is a design factor determined depending on the slot size and link speed. Within the slot period assigned to the random traffic, it works like the frame-based protocol C-CRA, where a reservation slot is broadcast once at the beginning of a frame. However, as can be seen in criterion 6, ASR does not count the slots allocated to periodic traffic. Thus, the RR slot is not issued regularly. In practice, the interval varies according to the periodic traffic load. We cannot set a regular interval for RR slots issuing in FQMA since the available slots for random traffic are not guaranteed at regular position. Also, random traffic cannot share the reservation slot with periodic traffic not to affect the successful rate of reservation request of periodic traffic. For the second best, we could try to make it as regular as possible within the random traffic slot period, which results in increasing the RR slot issuing rate at the cost of reducing the DR slot issuing rate. Thus, in that case we cannot control the allocation rate between RR slots and DR slots.

### 6.5 The effect of maximising RST

Bianchi [13] recently proposed the C-PRMA protocol using the scheduling scheme called the polling register (see Chapter 3). In the scheduling scheme, data slot issuing for periodic traffic is delayed for the maximum amount of time permissible without dropping packets. This maximising service delay scheme reduces $RST$s for queued packets close to zero even during a low-loaded period, consuming most slots on issuing reservation slots. Thus, packets are likely to be dropped as soon as the traffic loads exceed the link capacity.
The number of admitted flows are severely fluctuating in the PRMA-type protocols such as C-CPRMA, C-CRA and FQMA. It is because in such protocols a reservations is assigned based on a talkspurt of a flow even though a talk path sets up. Thus, the traffic load is dependent on the number of admitted flows not the number of set-up flows. The number of admitted flows is quite dynamic while the number of set-up flows is relatively static. Figure 6.5 shows an example of the number of admitted flows in FQMA, depicting the admitted flow number changing against the
time when 40 flows with sources modeled by the slow speech activity detector [68] are set up. In the example, all sources have a same packet generation rate and the capacity of the system is 20 flows. As can be seen, the system has several over-loaded periods. In the C-PRMA, packet drops must occur whenever the system is over-loaded.

However, in FQMA, the STS serves a packet as soon as it is queued in mobile hosts without delaying using criterion 5. It maximises RSTs for queued packets unlike in C-PRMA. When the number of admitted flows exceeds its capacity, the maximised RST delays packet drops at the cost of RST reduction of packets, though packets will be eventually dropped when their RSTs reduce to zero. Consequently, due to the bandwidth buffering effect, the system can serve more flows without packet drop than the capacity for short period during an over-loaded period. If an over-loaded period is short enough to tolerate, packet drops may not occur during an over-loaded period. Owing to the reasons, FQMA can reduce packet drop possibility. We will show the comparison with other protocols in the next section. Meanwhile, Figure 6.6 illustrates how the buffering effect by maximising RSTs works.

The y-axis in Figure 6.6 shows RSTs of served packets when the periodic traffic shown in Figure 6.5 is served under FQMA. In the example, the packet generation intervals of all flows are 20 slots and the packet lifetimes are 40 slots. The wireless link can support up to 20 flows without packet drop. During over-loaded periods, for instance two peaks between 5 – 7.5 sec, the RSTs of most served packets approach to the minimum. Some packets are dropped, as RSTs become to zero during the two peak periods. The period 2.5 – 5 forms contrast to the period 5 - 7.5. During the period 2.5 – 5, there are also two peaks that have over 20 flows, but the period length and peak load are shorter and smaller than those during 5 - 7.5. Thus, the RSTs are recovered soon after passing the peaks not to plummet to the zero. It is because RSTs have been kept high as passing the previous low-loaded period. It is certain that there is no packet drop during the two peak periods between 2.5 - 5. In C-PRMA or other schemes, packet-drops must occur during both the peak periods. Consequently, the buffering effect by maximising RST enables a base station to serve more flows than its capacity for a short period without packet drops. It improves the performance of FQMA over other schemes.

6.6 Evaluation study

In this section, we present the performance result of FQMA, simulating the proposed protocol in both uplink/downlink sharing and partitioning scheme using the OPNET simulator. This simulation targets a cellular network system where speech connections and data connections are served simultaneously. Figure 6.7 shows the wireless network topology used in this
simulation. As can be seen, there are 52 MHs: 44 MHs for periodic traffic, 4 MHs for AC random traffic and 4 MHs for WC random traffic. We use a speech source for periodic sources, and the speech source is implemented with a slow speech activity detector [68], which consists of talkspurts and silence gaps modeled as two-state Markov process with 32kbits/s coding rate. Figure 6.8 shows the speech source model implemented in the OPNET simulator. The random traffic of each MH is generated in different intervals according to each simulation case.

Figure 6.7 The wireless network topology used in the simulation

Figure 6.8 The speech source model implemented in OPNET simulator
For FQMA parameters, we select $m=5$ for comparing other protocols which generally adopt $m=5$, and we use $n=2$ for $m=5$. To compare other protocols, we choose the same system environment used in [4] as shown in Table 6.1. However, the channel rate is double and slot duration is half compared to those in [4] because we adopt a common channel scheme. In the simulation, the capture effect is ignored.

<table>
<thead>
<tr>
<th>Table 6.1 System Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel rate</td>
</tr>
<tr>
<td>Slot duration</td>
</tr>
<tr>
<td>Speech packet valid time</td>
</tr>
<tr>
<td>Type of speech activity detector</td>
</tr>
<tr>
<td>Mean principal talkspurt duration</td>
</tr>
<tr>
<td>Mean principal silence duration</td>
</tr>
</tbody>
</table>

6.6.1 The performance results of speech traffic service

In the PRMA-type protocol including the proposed FQMA, important factors are how many flows can be supported within the limitation of a packet drop rate and how much bandwidth can be allocated to the random traffic. We measure the packet drop rate of speech flows and the channel rate allocated to random traffic as a function of the number of connected speech flows. The RP slot issuing rate is also observed to see the efficiency of reservation slot issuing.

![Figure 6.9 Packet Drop Probability in the uplink/downlink partitioning scheme](image)
We simulate both directional services: the downlink and uplink service. Both cases are simulated in both uplink/downlink sharing and partitioning scheme respectively. For the downlink service, again we simulate two cases. Firstly, we placed all downlink sources in the base station so that no packet experiences network delay. We also drop a packet if its queuing delay in the base station has exceeded its lifetime. This case shows us the ideal performance of FQMA though no packet delay and packet dropping in the base station do not occur in practice. The figures from this simulation enables FQMA to be compared with the ideal performance. Secondly, we simulate a practical downlink service case. We still locate the downlink sources in the base station, but generated packets artificially experience various random network delay between 1 slot to 79 slots to emulate packet delays depending on network load and path.

We compared FQMA with C-CRA that has been proved to have the best performance among the frame-based RRA (Reservation Random Access) protocols (RRA-T, PR-CRA-D and C-PRMA) by Babich [4]. Figure 6.9 shows the packet drop probability of FQMA, C-CRA and two downlink service cases under uplink/downlink partitioning scheme. As can be seen, FQMA achieves an improvement over C-CRA by supporting approximately one more conversation at the limitation condition $P_{\text{drop}} < 0.01$ (1%). For reference, the quality of speech is audible up to 1% packet loss [44]. The Figure also shows that FQMA supports one less conversation than the ideal case which does not have reservation burden. As the connection number decreases, the performance gap between FQMA and C-CRA increases such that FQMA supports two more conversations at the lower limitation condition $P_{\text{drop}} < 0.001$. In Figure 6.9, we can also see the performance of downlink service when network delays are applied to packets. It supports approximately one less conversation at $P_{\text{drop}} < 0.001$ than the ideal case without network delay. Interestingly, the packet drop probability of the downlink service with network delay is similar to FQMA. It means the negative effect on performance by the FQMA protocol burden such as reservation slot issuing is similar to that by the inefficient scheduling for downlink due to not using the packet generation time information. Since a speech call sets up two connections – one incoming and one outgoing - in a cellular network, the lower number of supportable connections between the uplink and downlink service decides supportable number of calls in a cell. Therefore, the similar performance between uplink and downlink is desirable.

Figure 6.10 shows the bandwidth allocated to random traffic in the uplink/downlink partitioning scheme. FQMA assigns a slightly less bandwidth to random traffic than C-CRA, especially at higher number of conversations. The reduction of allocated bandwidth in FQMA occurs since FQMA allocates more slots to the speech connections to decrease packet drop probability. However, as the connection number decreases, both the protocols share a similar
allocation rate. This is because the number of dropped packets reduces significantly at lower conversation loads. Consequently, the reduction effect lessens as the connection number decreases. Meanwhile, the downlink service allocates more bandwidth to random traffic by around 4.5% than uplink service for FQMA spends some bandwidth for issuing reservation slots as indicated in Figure 6.11 which shows the bandwidth used by RP slot issuing. As explained in §6.2.2, FQMA broadcasts RP slots depending on the traffic load. Figure 6.11 illustrates the strategy that RP slot issuing rate decreases as the number of conversations increases while it keeps a regular rate at lower traffic load.

![Figure 6.10 Bandwidth allocated to random traffic in the uplink/downlink partitioning scheme](image)

**Figure 6.10 Bandwidth allocated to random traffic in the uplink/downlink partitioning scheme**

We can see the distinct benefit of uplink/downlink sharing scheme in Figure 6.12. As can be seen, the sharing scheme can support two more conversations than the partitioning scheme at the limitation condition $P_{\text{drop}} < 0.01$. Comparing with C-CRA, FQMA in the sharing scheme achieves 3.5 more conversations. Figure 6.13 shows the bandwidth allocated to random traffic. In the sharing scheme, since the uplink and downlink can shares each other their available bandwidth for random traffic, each allocation rate for uplink and downlink is meaningless. Thus we plot the total allocation rate of uplink and downlink. As can be seen, the sharing scheme performs similarly compared to the partitioning scheme but there is a slight less allocation at higher load since the sharing scheme uses more slot to speech connections to decrease packet drops.
Figure 6.11 Used bandwidth rate for RP slot issuing

Figure 6.12 Comparison of packet drop probability between the uplink/downlink partitioning and sharing scheme
6.6.2 The performance results of random traffic service

We also simulate the uplink service for random traffic in the same environment above. We choose $d = 20$ for this simulation. We do not simulate the downlink service for random traffic since we already conducted it in Chapter 5. This simulation targets to identify the three performances in serving random traffic: rate-based service, local fairness and global fairness. To produce suitable situations for each case, we change random traffic parameters and the number of speech flows connected. We use constant packet generation intervals for random traffic to observe service differences among flows with easy. The traffic parameters of the speech flows follow the previous simulations for speech traffic service throughout all simulation cases in this section. We also use the terms defined in Chapter 5.

A. Rate-based service

We simulate 3 difference cases to see the performance of rate-based service in FQMA: surplus, semi-deficit, and deficit. Different number of speech flows are applied to produce the three kinds of traffic load situations, while the traffic parameters of all random flows are same in the three cases. Table 6.2 shows the parameters for random traffic flows used for this simulation.

For the surplus case simulation, 4 speech flows are connected. In this simulation case, since the total random traffic load is only 432 kbits/s, the available bandwidth to random traffic is high.

---

**Figure 6.13** Comparison of the total bandwidth allocated to random traffic between uplink/downlink partitioning and sharing scheme
enough to serve all random flows. Thus, as can be seen in Figure 6.14 (b) all random flows receive their allocated bandwidth. Figure 6.14 (a) shows that w-virtual time equals a-virtual time all the time as it is a surplus period. We can also find that both virtual times drop down to zero frequently. It means that the service rate for all random flows is high enough to empty the queue of PCAWFQ\textsuperscript{'} frequently. Figure 6.14 (d) shows that among the same class flows the flow with higher allocated bandwidth experiences less delay than lower one. As I explained in §3.1.6, in the finish time based scheduling scheme a flow with higher allocated bandwidth is served earlier because of its smaller normalised service time. However, as can be seen, the rule does not work between WC and AC flows since there is a priority between both in PCAWFQ\textsuperscript{'}.

Table 6.2 Random traffic flows used in the rate-based service simulation

<table>
<thead>
<tr>
<th>Flow</th>
<th>Packet Generation Interval (sec)</th>
<th>Allocation (bit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC flow 1</td>
<td>0.00457 (112035 bit/s)</td>
<td>112000</td>
</tr>
<tr>
<td>WC flow 1</td>
<td>0.00533 (96060 bit/s)</td>
<td>96000</td>
</tr>
<tr>
<td>AC flow 2</td>
<td>0.0064 (80000 bit/s)</td>
<td>80000</td>
</tr>
<tr>
<td>WC flow 2</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
<tr>
<td>AC flow 3</td>
<td>0.01067 (47985 bit/s)</td>
<td>48000</td>
</tr>
<tr>
<td>WC flow 3</td>
<td>0.016 (32000 bit/s)</td>
<td>32000</td>
</tr>
</tbody>
</table>

Figure 6.15 shows the results of the semi-deficit case. 18 speech flows are connected to make this situation. In this case, the available bandwidth for WC flows is not high enough to serve them all while it is enough for AC flows. Figure 6.15 (b) shows that AC flows receive their allocated bandwidth independently of speech traffic loads while WC flows are served less than their allocation depending on the available bandwidth. Figure 6.15 (a) shows virtual times indicating the service status. As all AC flows receives exactly their allocated bandwidth during this period, the slope of a-virtual time is 1, while the slope of w-virtual time is fluctuating according to the available bandwidth maintaining that the total increment is less than a-virtual time. Due to the lower rate service than packet generation rate, the delays of WC flows keep increasing during the simulated period, while AC flows do not as can be seen in Figure 6.15 (c) and (d). In this case, also, the delays of flows do not conform to the delay rule explained in §3.1.6 since the traffic is over-loaded: i.e. in an over-load period, the delays are more dependent on how early successfully admitted through the reservation slot rather than the size of allocated bandwidth as the access delay rapidly increases due to frequent conflicts and longer interval of RR slot issuing.
Figure 6.16 shows the results of the deficit case simulation. 34 speech flows are connected, so the AC flows as well as WC flows hardly receive their full allocated bandwidth. As can be seen in Figure 6.16 (a), the service for WC flows stops after around 1.5 sec due to the shortage of bandwidth. The delays of AC flows keep increasing depending on the speech traffic loads as the AC flows receive less than their allocated bandwidth. The average delay gap among AC flows is higher than that in the semi-deficit case. It is because the variation of the access delay is longer due to very severe congestion.

B. Local Fairness

As I defined in Chapter 5, the local fairness is the fairness among the same class flows. To observe the local fairness performance, we simulate similar cases to section A above except that all flows have the same allocation rate. Table 6.3 shows the parameters used in this simulation.

<table>
<thead>
<tr>
<th>Flow</th>
<th>Packet Generation Interval (sec)</th>
<th>Allocation (bit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC flow 1</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
<tr>
<td>WC flow 1</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
<tr>
<td>AC flow 2</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
<tr>
<td>WC flow 2</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
<tr>
<td>AC flow 3</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
<tr>
<td>WC flow 3</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
</tbody>
</table>

Figure 6.17 shows the results of a surplus case. As can be seen in Figure 6.17 (b) all flows received the same amount of service. In this simulation, all flows have the same allocation rate and same packet size so ideally they should have the same delay. But as can be seen in (d) they experience different delays. It is because of the conflicts during the reservation phase. The earlier admitted flow experiences less delays than others. However, due to the priority service between AC and WC, AC flows receive earlier services than WC flows.

Figure 6.18 shows the results of a semi-deficit case. As can be seen in (b), the all AC flows receive their allocation rates while all WC flows receive less than their allocation. However, all same class flows are served at the same rate. Figure 6.19 shows the result of a deficit case. As can be seen in Figure 6.19 (b) the service for WC flows are stopped due to the bandwidth shortage. The service rates of AC flows are the same.
**C. Global Fairness**

As I defined in Chapter 5, the global fairness is the fairness between different class flows. The AC flows should be served with higher priority, but it is allowed within their allocation rate. Thus, if the available bandwidth is not high enough to serve all WC flows as committed, the AC flows can be served only within their allocated bandwidth. We simulate a case to observe this property. Table 6.4 shows the parameters used in this simulation, where the service period is semi-deficit. All AC flows are greedy - i.e. they send data at higher rates than allocated one - while the WC flows transmits data at their allocated rates. Figure 6.20 (b) show the global fairness performance. AC flows receives only their allocated bandwidth even though they are greedy, while WC flows share the remaining bandwidth according to their allocated rates.

<table>
<thead>
<tr>
<th>Flow</th>
<th>Packet Generation interval (sec)</th>
<th>Allocation (bit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC flow 1</td>
<td>0.004 (128000 bit/s)</td>
<td>112000</td>
</tr>
<tr>
<td>WC flow 1</td>
<td>0.00533 (96060 bit/s)</td>
<td>96000</td>
</tr>
<tr>
<td>AC flow 2</td>
<td>0.004 (128000 bit/s)</td>
<td>80000</td>
</tr>
<tr>
<td>WC flow 2</td>
<td>0.008 (64000 bit/s)</td>
<td>64000</td>
</tr>
<tr>
<td>AC flow 3</td>
<td>0.004 (128000 bit/s)</td>
<td>48000</td>
</tr>
<tr>
<td>WC flow 3</td>
<td>0.016 (32000 bit/s)</td>
<td>32000</td>
</tr>
</tbody>
</table>
Figure 6.14 Simulation results for uplink random traffic service with various allocation rates in a surplus period - 4 speech uplink flows are connected
Figure 6.15 Simulation results for uplink random traffic service with various allocation rates in a semi-deficit period - 18 speech uplink flows are connected.
Figure 6.16 Simulation results for uplink random traffic service with various allocation rates in a deficit period - 34 speech uplink flows are connected.
Figure 6.17 Simulation results for uplink random traffic service with the same allocation rate in a surplus period - 3 speech uplink flows are connected
Figure 6.18 Simulation results for uplink random traffic service with the same allocation rate in a semi-deficit period - 19 speech uplink flows are connected.
Figure 6.19 Simulation results for uplink random traffic service with the same allocation rate in a deficit period - 35 speech uplink flows are connected
(a) Virtual Times (VTs) and Service time

(b) Received packet number

(c) End-to-End delay

(d) Average End-to-End delay

Figure 6.20 Simulation results for uplink random traffic service with greedy AC flows in a semi-deficit period - 18 speech uplink flows are connected
6.7 Summary

In this Chapter, we have presented a reservation multiple access protocol, called FQMA. As a slot-based scheme, FQMA can serve periodic traffic and random traffic keeping the de-coupling between both traffic. To do so, we introduced an uplink scheduler that schedules periodic flows based on PEDQ (Packet Endurable Delay Queuing) and random flows based on PCAWFQ. The scheduler controls uplink slot sequence using the information transmitted from mobile hosts upon reservation request phase. The scheduler is capable of supporting more flows than channel capacity for a short period owing to the bandwidth buffering effect of maximised RST, which reduces the packet drop probability of periodic traffic. It also supports uplink random traffic service in the same way as the downlink random traffic service conducted by PCAWFQ. The proposed protocol is simulated in the OPNET simulator, and the results show that FQMA performs better than other protocols at the cost of random traffic bandwidth reduction at high traffic load. The bandwidth reduction is not significant since it is less than 1%. The simulation results also show that the uplink/downlink sharing scheme performs better than the partitioning scheme. The random traffic service is also simulated in terms of rate-based service, local and global fairness. We identified that PCAWFQ in FQMA can serve random traffic conforming to the definition of CAWFQ. We validated this simulation by comparing its results with those predicted by mathematical analysis in Chapter 5 and there was sufficient agreement to give us confidence of a correct simulation implementation.
Chapter 7

Handover management

In a cellular packet switched network, a flow may move around among cells during a connection period. A handover between cells results in the scheduler that has served the flow being switched to a new one located in the new base station (BS). This migration affects the fairness of the scheduler. In this chapter, we investigate the packet scheduling service problems that arise due to the handover and then propose solutions to enable our proposed service scheme to support continuous fair service to all flows regardless of handovers. We also propose a handover protocol to correct out-of-order packet delivery during a handover co-operating with the proposed scheme.

7.1 The effects of handover on scheduling performance

When the receiver in a flow moves to another cell, the source of the flow needs to switch its destination to another BS. The flow migration may also require packet forwarding if it occurs when its flow is backlogged: i.e. if packets are queued in the scheduler buffer of the old BS on handover, these should be forwarded to the new BS so as not to be lost. The packet forwarding always occurs in some handover protocol such as Caceres’s handover protocol [19] protocol where an old BS always forwards few last packets, which have been already transmitted to the mobile host by the old BS, to the new BS so that the new BS transmits them to the mobile host again to reduce packet loss during handover especially between non-overlapped cells. A handover protocol deals with the migration functions to migrate a flow seamlessly during a handover. Now let us investigate what happens in the old and new BSs on a handover.

7.1.1 Service distortion in the old BS

The handover of a connection usually causes both uplink and downlink flows to migrate. In terms of packet scheduling service, services for both flows are affected in different ways as follows.

7.1.1.1 Parameter errors in a downlink scheduler

As I introduced in Chapter 3, there has been no lack of research effort on scheduling disciplines to run service queues for routers in a wired network. They are designed assuming that
a flow does not move while backlogged. If such disciplines are applied to a BS queue without modification, service distortion in the old BS occurs after a handover if there was a backlog. The service distortion (unfair bandwidth distribution) arises due to parameter errors caused by forwarded packets because the parameter values in the old BS have been calculated on the assumption that all received packets are serviced which is no longer the case. We discover that the effect of the parameter error lasts until the parameters are reset when the queue is empty, and the parameter errors are cumulative when multiple handovers occur. These will be explained in detail in §7.2.

The parameter errors do not always occur. It depends on how the scheduler’s parameters depend on the packet forwarding. The errors should occur in those schedulers which have system parameters computed based on the backlogged flows such as the virtual times of WFQ [30], WF-Q [9] and PCAWFQ - which we will call PFD (Packet Forwarding Dependent) type disciplines. In this type of scheduler, the virtual time is mis-computed when packets, which have contributed to tick the virtual time, are forwarded without being served. By contrast, the other type of schedulers such as SCFQ [39] and SFQ [43] - which we will call PFI (Packet Forwarding Independent) type disciplines - do not have the parameter error problems though they also adopt the virtual time concept. It is because their virtual time are not dependent on the packet forwarding: i.e. they approximate their virtual time with the current served packet’s virtual finish time instead of computing it precisely. However, as introduced in Chapter 3, this type of scheduler has unbounded fairness problem due to the virtual time approximation to reduce computational complexity, so the service performance is lower than the PFD disciplines. Consequently, as a handover affects the PFD scheduling service performance in an old BS, the handover effect should be considered when a PFD scheduling discipline is designed for a cellular packet switched network.

7.1.1.2 Parameter errors in uplink service

Unlike a downlink flow, the packet forwarding does not occur during a handover of an uplink flow since the incoming packets queued in the old BS can be sent to the destination through a wired network. However, parameter errors may still occur in the uplink scheduler if the multiple access protocol in BS adopts a PFD type discipline to schedule the uplink traffic.

In FQMA, the parameter error always occurs in uplink services, since the uplink scheduler always keeps a proxy packet for an uplink flow that admitted. Upon a handover, the uplink scheduler discards the proxy packet of the handover flow. As the discarded proxy packet produces inaccuracy in the virtual times of CAWFQ', a handover always affects the performance
of the uplink scheduler in FQMA unlike downlink service, where it does only when packets are forwarded.

7.1.2 The handover effects in new BS

While a handover flow affects the service for the remaining flows in the old BS, it does not affect the service for the existing flows in the new BS. Instead, the migrated flow receives damage to its service in the new BS. There are two service distortions: delay to the forwarded packets due to the new BS treating them as new arrivals and out-of-order packet delivery during a handover. These problems appear only during a handover period.

7.1.2.1 The arrival time changes of forwarded packets

When forwarded packets arrive at the new BS, the scheduler in the new BS uses their arrival times at the new BS in calculating their time stamps, which results in additional service delay for those packets. It is not possible to use the old arrival times for the packets since the flow did not have any allocation in the new BS when the forwarded packets arrived at the old BS.

7.1.2.2 Out-of-order packet delivery

A handover protocol notifies the movement of an MH to the corresponding source so that the source can change the destination to the new BS. This may result in the packets reaching the new BS earlier than the forwarded packets. Since the new BS serves all packets based on their arrival times at the new BS, it may result in the packets being misordered when delivered to the MH.

7.2 Service distortion of WFQ

The parameter errors in PFD discipline due to the packet forwarding cause unfair service among flows in the old BS. In this section, we analyse how it affects the fairness of WFQ, the representative PFD discipline.

7.2.1 Virtual time lag by packet forwarding

To measure the scheduling service fairness, Golestani [39] introduced a definition of fairness, termed relative fairness. It is defined as the maximum difference between the normalised service received by any two flows when they are continuously backlogged over an interval: i.e. for any two continuously backlogged flows $i, j$ in an interval $(\tau, t]$, the fairness is defined as follows:
This measure shows how well a scheduling discipline guarantees the share of a flow in a backlogged period.

Stiliadis [86] proved that WFQ has bounded fairness using a RPS (Rate-Proportional Server). According to Theorem 4 of [86], the relative fairness of RPS is bounded as follows:

\[
\frac{W_i(\tau, t)}{r_i} - \frac{W_j(\tau, t)}{r_j} \leq \Delta P
\]

\(\Delta P\) represents the maximum difference between \(P_i(t)\) (the potential of the flow being serviced) and \(P(t)\) (the system potential in a RPS), where \(P_i(t)\) represents the normalised service amount received by flow \(i\) and \(P(t)\) is defined as a function which can keep track of the progress of the total work done by the scheduler. Schedulers use different functions to maintain the system potential, giving rise to widely different delay- and fairness-behaviors. The packet-by-packet version of RPS, PRPS (Packet-by-packet Rate-Proportional Server) has the following fairness bound:

\[
\left| \frac{W_i(t_1, t_2)}{r_i} - \frac{W_j(t_1, t_2)}{r_j} \right| \leq \Delta P + C_j + \frac{L_{\text{max}}}{r_i} + \frac{L_j}{r_j}, \Delta P + C_i + \frac{L_{\text{max}}}{r_i} + \frac{L_i}{r_i}, \quad (1)
\]

where

\[
C_i = \min((V - 1)\frac{L_{\text{max}}}{r_i}, \max(\frac{L_n}{r_n}))
\]

GPS belongs to the RPS according to the RPS definition and in GPS the system potential is defined as the potential of the flow being served. Thus we have \(\Delta P=0\) in GPS. Then, by the Eq. (1), it is proved that WFQ, which is a packet-by-packet version of GPS, has bounded fairness. The system potential corresponds to the virtual time \(V(t)\) in WFQ and the function \(F(t)\) which connects the virtual finish times of the packets that have just finished their service, represents the potential of a flow currently being served in WFQ. Thus the difference between \(V(t)\) and \(F(t)\) indicates the fairness at \(t\) in WFQ.

We discover that Eq. (1) no longer holds in a cellular packet switched network with handovers because the fairness value of the old BS increases whenever a handover occurs and the increment allows a newly joined flow after handovers to monopolise the output link for a short period from when it joins. Moreover, the influence does not disappear completely after the monopoly service period even though the increase reduces. In other words, the fairness value of the old BS cumulatively increases as handovers occur. We will now show how the fairness value increases due to a handover.
**Theorem 1:** In a WFQ based system, if a backlogged flow $f_h$ moves to a neighbour cell at time $t_n$ forwarding remaining packets $P_0, P_1, P_2, \ldots$ to the new BS and $t_0$ is the time queued in the old BS of $P_0$, then the virtual time $V(t_n)$ of the old BS lags by $\delta$ where $\delta > 0$.

**Proof:** Let $t_k, k = 0, 1, 2, \ldots n$ define event sequence times of packet departure and arrival including the handover event in the old BS. Then, the virtual time $V(t_n)$ is given by

$$V(t_n) = V(t_0) + \sum_{k=0}^{n-1} \left( \frac{t_{k+1} - t_k}{\sum_{i \in B(t_k, t_{k+1})} r_i + r_{f_h}} \right)$$  \hspace{1cm} (2)

where $B(t_k, t_{k+1})$ represents a set of the backlogged flows excluding $f_h$ in the interval $(t_k, t_{k+1})$ and $r_i$ indicates the share rate of flow $i$. Let $	ilde{V}(t_n)$ be the virtual time when there is no $f_h$ in the interval $(t_0, t_n)$, then $	ilde{V}(t_n)$ is given by

$$\tilde{V}(t_n) = V(t_0) + \sum_{k=0}^{n-1} \left( \frac{t_{k+1} - t_k}{\sum_{i \in B(t_k, t_{k+1})} r_i} \right)$$  \hspace{1cm} (3)

Subtracting (2) from (3)

$$\delta = \sum_{k=0}^{n-1} \left( \frac{t_{k+1} - t_k}{\sum_{i \in B(t_k, t_{k+1})} r_i + r_{f_h}} \right) - \sum_{k=0}^{n-1} \left( \frac{(t_{k+1} - t_k)}{\sum_{i \in B(t_k, t_{k+1})} r_i} \right) = \sum_{k=0}^{n-1} \left( \frac{\sum_{i \in B(t_k, t_{k+1})} r_i + r_{f_h}}{r_{f_h}} \right)$$

Since all $r_i$ are positive, $\delta$ is positive. Upon the handover, as the remaining packets of the flow $f_h$ are forwarded to the new BS, $f_h$ no longer belongs to the backlogged set during the interval time $(t_0, t_n]$. Considering the situation, the correct virtual time must be $\tilde{V}(t_n)$, not $V(t_n)$ that has been computed including $f_h$. Therefore, theorem 1 follows. \( \square \)

While $V(t_n)$ lags behind, $F(t)$ has correctly increased until $t_n$ regardless of forwarded packets because it is dependent on the transmission rate of the link. Thus we conclude that if a handover occurs at $t_n$ then the fairness value of WFQ increases by $\delta$ after $t_n$ as the followings:

$$\left| \frac{W_i(t_1, t_2)}{r_i} - \frac{W_j(t_1, t_2)}{r_j} \right| = \begin{cases} 
\max(\Delta P + C_j + \frac{L_{\max}}{r_i}, \Delta P + C_i + \frac{L_{\max}}{r_j}), & t_2 < t_n \\
\max(\Delta P + C_j + \frac{L_{\max}}{r_i}, \Delta P + C_i + \frac{L_{\max}}{r_j} + \delta), & t_2 \geq t_n
\end{cases}$$
7.2.2 Unbounded fairness

The lag of virtual time in the old BS occurs because the forwarded packets abandon their services in the old BS even though the packets have been already included in computing the virtual time. The time lag affects the fair service of WFQ. For example, because the virtual finish times of a new flow's packets are computed using the virtual time that has been lagged, then the virtual finish times also lag behind. It results in giving higher service priority to the new flow than existing flows in the old BS as packets are served in increasing order of virtual finish time. Figure 7.1 describes the situation. As can be seen, when a flow comes in, its virtual finish time begins from the lagged virtual time which is earlier than the current served packet's virtual finish time by δ. Thus, the new flow is served exclusively until its virtual finish time equals the virtual finish time of the served packet. The fairness gap still remains after the monopoly service period. It means that fairness value diverges as the number of handover increases: i.e. the fairness value of WFQ is not bounded in the presence of handovers.

Figure 7.1 Service distortion due to the virtual time lag
7.2.3 Identification by simulation

In this section, we introduce the simulation results carried out using the OPNET simulator which show that WFQ produces unbounded fairness when handovers are taken account, even though it has bounded fairness property in a single cell operation. The network configuration for the simulation consists of two BSs (BS1 + BS2) which are connected to a 10Mbits/s Ethernet bus as can be seen in Figure 7.2. The 9 downlink flows are set up between 9 workstations and 9 MHs respectively. All workstations are attached to the same Ethernet segment and have UDP sources to transmit UDP packets to a corresponding MH located in a cell. WFQ scheduler is implemented in the MAC layer of BSs. For supporting IP level handovers, Caceres’s handover protocol [19] is implemented above the IP layer of BSs and MHs. The scheduler processes handover messages to obtain handover information for packet forwarding. The 2 Mbits/s wireless link is allocated to each flow as shown in Table 7.1.

To identify the unbounded fairness problem explained above section, we simulate the following scenario. Each flow becomes active for a short period and moves to BS2 one after another shown in Table 7.1. After finishing 7 handovers by 1900ms, flow 9 becomes active at 2000ms. Figure 7.3 shows $V(t)$ and $F(t)$ in WFQ located in BS1. As can be seen in Figure 7.3, the gap between both functions increases as handovers occur. Figure 7.4 plots the gap to show the fairness variation by handovers clearly. In the Figure, the reason that the gap oscillates is due to the packet-by-packet service which cannot serve multiple packets simultaneously even though they have the same virtual finish time. Thus the oscillations are inevitable in WFQ and the maximum amplitude of the oscillation can be computed by Eq. (1). In the Figure, compared to the previous activity burst, the one commencing at around 2 seconds has much higher frequency oscillation and much low amplitude. Figure 7.4 also shows that the gap increment size by a handover varies depending on the period length during which the handover flow has affected the virtual time.

Figure 7.5 shows how the gap increment affects the practical service. After 7 consecutive handovers, flow 9 joins the BS1 which has served flow 8. Since flow 8 and 9 have exactly the same reservation bandwidth and traffic parameters, both should be served alternately. However, as can be seen in Figure 7.5, flow 9 has received services exclusively during 0.012 sec (service time for 11 packets) due to the cumulative fairness gap. The exclusively served packets have earlier virtual finish times than the virtual finish time of the served packet of flow 8 upon flow 9’s joining. As Figure 7.4 shows that the gap still remains after the exclusive service period, the unfair service will occur again on next arrival flow even though the effect will be reduced.
Table 7.1 The parameters for flows used in the simulation

<table>
<thead>
<tr>
<th>Flow</th>
<th>Reserved Bandwidth</th>
<th>Packet Size</th>
<th>Flow start time</th>
<th>Handover time</th>
<th>Inter arrival time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 kbits/s</td>
<td>10240 bit</td>
<td>500 ms</td>
<td>700 ms</td>
<td>0.02 sec</td>
</tr>
<tr>
<td>2</td>
<td>100 kbits/s</td>
<td>10240 bit</td>
<td>720 ms</td>
<td>900 ms</td>
<td>0.02 sec</td>
</tr>
<tr>
<td>3</td>
<td>100 kbits/s</td>
<td>10240 bit</td>
<td>920 ms</td>
<td>1100 ms</td>
<td>0.02 sec</td>
</tr>
<tr>
<td>4</td>
<td>100 kbits/s</td>
<td>10240 bit</td>
<td>1120 ms</td>
<td>1300 ms</td>
<td>0.02 sec</td>
</tr>
<tr>
<td>5</td>
<td>100 kbits/s</td>
<td>10240 bit</td>
<td>1320 ms</td>
<td>1500 ms</td>
<td>0.02 sec</td>
</tr>
<tr>
<td>6</td>
<td>100 kbits/s</td>
<td>10240 bit</td>
<td>1520 ms</td>
<td>1700 ms</td>
<td>0.02 sec</td>
</tr>
<tr>
<td>7</td>
<td>100 kbits/s</td>
<td>10240 bit</td>
<td>1720 ms</td>
<td>1900 ms</td>
<td>0.02 sec</td>
</tr>
<tr>
<td>8</td>
<td>600 kbits/s</td>
<td>2048 bit</td>
<td>500 ms</td>
<td>No</td>
<td>0.001 sec</td>
</tr>
<tr>
<td>9</td>
<td>600 kbits/s</td>
<td>2048 bit</td>
<td>2000 ms</td>
<td>No</td>
<td>0.001 sec</td>
</tr>
</tbody>
</table>
Figure 7.3 The F(t) and V(t) in WFQ of BS1
7.3 Approaches to remedy the parameters errors in PFD disciplines

7.3.1 Three approaches

There are three possible approaches to remedy the parameter errors: Re-computing, Compensation and Consuming. For the first of the items, the distorted parameter can be recomputed excluding the handover flow from the last packet service time of the handover flow whenever packet forwarding occurs. This is an ideal remedy, but it requires high computational burden especially if multiple handovers occur over a short period. In the second case, we can estimate the correct value of the distorted parameter with simple computation so that the parameter can be re-configured. The estimation usually contains errors that could increase fairness value, so it would reduce overall performance in terms of fairness. In addition, it is difficult to estimate the correct parameter value in those systems that use multiple system parameters like PCAWFQ, while we can rather easily estimate it in the system with single parameter such as WFQ. Finally, we could make the inaccurate parameter correct by just pausing...
service during the corresponding service period for the forwarded packets or assigning the link to other traffic irrelevant to the queued traffic during the period. In the former case, the output link’s bandwidth is wasted. This is undesirable especially in a low rate link. Therefore the latter scheme is recommended if applicable.

7.3.2 Compensation scheme for WFQ

We here propose a simple compensation scheme for WFQ to fix the lagged virtual time. Assuming that the virtual time at \( t_n \) equals the virtual finish time of served packet at \( t_n \) when a handover occurs at \( t_n \) we can obtain the estimated virtual time \( V_e(t_n) \) as shown in Figure 7.6. Then \( V_e(t_n) \) can be easily calculated using the virtual finish times of being served packet at \( t_n \) and next served packet with the following equation:

\[
V_e(t_n) = F_{i}^{k} + (t_n - t_i)(F_{j+1}^{k} - F_{i}^{k}) * \frac{L_{i}^{k}}{R}
\]

<table>
<thead>
<tr>
<th>( t_s )</th>
<th>The service start time of current packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_m )</td>
<td>Last packet arrival time</td>
</tr>
<tr>
<td>( i )</td>
<td>The currently serviced flow</td>
</tr>
<tr>
<td>( k )</td>
<td>The currently serviced packet number</td>
</tr>
<tr>
<td>( j )</td>
<td>Next flow scheduled to be served</td>
</tr>
<tr>
<td>( L_{i}^{k} )</td>
<td>The size of ( k ) packet of ( j )th flow</td>
</tr>
<tr>
<td>( R )</td>
<td>Bit throughput of a wireless link</td>
</tr>
<tr>
<td>( F_{i}^{k} )</td>
<td>Virtual finish time of ( k )th packet of ( j )th flow</td>
</tr>
</tbody>
</table>

\[ F_{j+1}^{k} \]
\[ V_e(t_n) \]
\[ F_{i}^{k} \]
\[ V(t_n) \]

\[ t_s \]
\[ t_n \]

**Figure 7.6 Estimation of correct virtual time on a handover**
The last packet arrival time $t_m$ in WFQ system is also replaced with $t_n$ since the estimated virtual time $V_e(t_n)$ has been reflected up to $t_n$. The assumption of the proposed scheme introduces errors in the region of maximal fairness value of WFQ, so this estimation scheme may not compute accurate virtual time at $t_n$ all the time. In addition, this scheme cannot correct the miscalculated virtual finish times for packets that have arrived during the period between the last service time of the forwarded flow and $t_n$. However, with this one-off simple estimation, we can approximate the virtual time within a limited error bound.

![Graph 1](image1.png)

(a) Using the “Re-computing” approach

![Graph 2](image2.png)

(b) Using the proposed compensation approach

**Figure 7.7 The gap between F(t) and V(t) in corrected WFQs in BS1**

7.3.3 Simulation results of the “Re-computing” method and the proposed compensation scheme for WFQ

We simulated the “Re-computing” and proposed compensation schemes using the same configuration used in §7.2.3. Figure 7.7 shows the fairness gap of the two schemes. As can be seen, both schemes can prevent the cumulative increase of the fairness values regardless of
handovers unlike the unmodified WFQ as shown in Figure 7.4. Figure 7.7 (b) shows that the compensation scheme produces small errors comparing to the re-computing scheme after 2 second. It has larger amplitude of the oscillations. However, the error term is so small that it does not affect practical packet service as shown in Figure 7.8.

![Diagram](image)

(a) Using the "Re-computing" approach

(b) Using the proposed compensation approach

**Figure 7.8 Service order of flow 8 and 9 in corrected WFQs in BS1**

Figure 7.8 shows the packet services of flow 8 and 9. As can be seen, both schemes serve packets fairly. The packets of flows 8 and 9 are alternately serviced from 2000ms as we expected since they have the same reserved bandwidth and traffic parameters. From Figure 7.7 and 7.8, we conclude that the proposed scheme has similar performance compared to the "re-computing" scheme.

### 7.4 Protecting the service distortion of PCAWFQ and PCAWFQ'

The PCAWFQ and PCAWFQ' are virtual time based disciplines as WFQ, so they have the same virtual time lag problem upon a handover. Since both have two virtual times for AC and WC flows respectively which tick based on the set of each class’s backlogged flows, AC packet forwarding causes a-virtual time lag and WC packet forwarding w-virtual time lag. The effect of the virtual time lag in PCAWFQ is the same as I explained in the WFQ case. These are easily proven using Eq (22) and (23) in Chapter 5. In this section, we propose solutions to protect the
virtual time lags in both the scheduling disciplines. As explained in §7.1.1, the virtual time lags in PCAWFQ and PCAWFQ’ have different causes. The virtual time lag of the former is caused by packet forwarding, while that of the latter is caused by proxy packet elimination. In this section, we consider the “consuming” approach as a remedy for the virtual time lags which does not accompany the output link wasting.

7.4.1 “Consuming” approach for PCAWFQ

In our proposed service scheme, PCAWFQ is the scheduling discipline for the downlink random traffic service, so packet forwarding may occur upon a handover of a downlink random flow. To correct the virtual time lag due to the packet forwarding, the virtual time associating the handover flow should be re-computed according to one of the approaches introduced in §7.3.1. It is not easy to re-compute the virtual times since we should know a-delay times and w-delay times for all packet arrival and departure events. Instead of re-computing them, we choose the “consuming” approach to keep the virtual times correct regardless of packet forwarding. In our proposed “consuming” approach, the downlink scheduler in the old BS allocates the slots supposed to be used by the forwarded packets to other traffic flows which do not belong to downlink AC and WC. Since there are several types of the other traffic, the available slots are assigned according to the traffic service priority decided by traffic characteristic, giving higher priority to the downlink traffic for the same kind of traffic. Table 7.2 shows the assignment priority among the other traffic.

Table 7.2 Assignment priority for PCAWFQ and PCAWFQ’

<table>
<thead>
<tr>
<th>Priority in PCAWFQ</th>
<th>Priority in PCAWFQ’</th>
<th>Direction</th>
<th>Traffic</th>
<th>Condition for assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>Downlink</td>
<td>Periodic</td>
<td>Non-empty queue</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Uplink</td>
<td>Periodic</td>
<td>Non-empty queue &amp; Criterion 5 in Chap 6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Downlink</td>
<td>AC and WC</td>
<td>Non-empty queue</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Uplink</td>
<td>AC and WC</td>
<td>Non-empty queue</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>Downlink</td>
<td>Best-effort</td>
<td>Non-empty queue</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>Uplink</td>
<td>Best-effort</td>
<td>Non-empty queue</td>
</tr>
</tbody>
</table>

Thus, an available slot is assigned to the traffic with the lowest priority number which satisfies its associated assignment condition as shown in Table 7.2. The assignment condition
ensures that a slot is assigned to an MH with non-empty queue. For the uplink periodic traffic, there is an additional condition, Criterion 5 in Chapter 6 which is used to decide whether or not the real packet supposed to receive the assigned slot has been generated. This is necessary in order not to waste the assigned slot. Such slot consumption by other traffic keeps the virtual times correct regardless of packet forwarding without wasting slots.

7.4.2 “Consuming” approach for PCAWFQ’

PCAWFQ’ uses the proxy packets representing their corresponding real packets. Thus, in PCAWFQ’, the packet forwarding does not occur upon a handover. Instead, the uplink scheduler eliminates the proxy packet for the handover flow. However, the effect of the proxy packet elimination is the same as the packet forwarding in PCAWFQ. The difference compared to the packet forwarding is that packet elimination always occurs upon a handover and only a proxy packet is eliminated. Except for the two facts, the handover effect of PCAWFQ’ is identical to that of PCAWFQ. Therefore, the solution for virtual time lag is the same as PCAWFQ except that the slot assignment priority is different as shown in Table 7.2 since the uplink should have higher priority than downlink traffic for PCAWFQ’.

7.5 A handover protocol without out-of-order packet delivery

A handover protocol performs seamless migration of flows from one cell to another cell. To support a seamless handover for a cellular packet switched network, there has been much research effort in different ways [19], [74], [7]. They achieve various packet delays, and packet loss rates depending on its policies such as a retransmission packet scheme to reduce packet loss [7] or Mobile IP [74] to support transparent routing of IP datagram to mobile nodes, as well as the control flow itself of the protocol. Provided that a BS adopts a rate based scheduling discipline to ensure a committed QoS to a flow, a handover protocol influence the performance of the discipline. As a consequence, an applied handover protocol should be harmonised with the scheduling discipline.

7.5.1 Caceres’s handover protocol

Figure 7.9 shows how packet delivery may be mis-ordered in Caceres’s handover protocol [19]. In the figure, packet ① which departs from a source before receiving the redirect message is sent to the old BS. However, MH has already moved to the new BS when packet ① arrives at the old BS, so packet ① is forwarded to the new BS to be delivered to the MH. Meanwhile, packet ② is sent to the new BS directly because of the redirect message. Since packet ① travels one hop
more compared to packet ②, it arrives at the new BS later than packet ② if the inter-departure time between the two packets is smaller than the journey time difference between the two packets or packet ① is queued in the old BS longer than (inter-departure time - the journey time difference). The scheduler in the new BS serves packets based on arriving time at the new BS, so such out-of-order arrival of packets results in out-of-order delivery to the MH. If a handover protocol does not consider such cases, the out-of-order packet delivery could occur on handover.

7.5.2 The effect of out-of-order packet delivery

Out-of-order packet delivery can be fixed in the upper layer protocol at the receiver such as TCP. However, TCP assumes that only a maximum reordering of 3 packets will occur in normal operating condition. In other words, if a source receives more than two duplicated acknowledgements from the receiver due to out-of-order packet delivery, it regards it as packet loss. Then, it reduces its congestion window by half and doubles the time-out parameter value. Those parameter changes decrease the data transmission speed of TCP. Thus, out-of-order packet
delivery may degrade the performance of the TCP connections. Therefore out-of packet delivery should be minimized.

7.5.3 A forwarding-aware handover protocol

We propose a modified version of Caceres's handover protocol to fix the out-of-order packet delivery problem as shown in Figure 7.10. On receiving a Greeting message, the new BS locks the service for redirected packets until packet forwarding is completed. The service locking lasts until a Forwarding End message arrives from the old BS. This mechanism prevents redirected packets from being served before the forwarded packets are served.

Another difference from the Caceres's handover protocol is the timing of a Redirect Message. The new BS sends a Redirect message to the source just after it receives a Greeting message from an MH rather than after receiving the Notify ACK message from the old BS. This strategy minimises the number of forwarded packets, redirecting packets to the new BS as quickly as possible.

After receiving a Notify message, the old BS starts to forward queued packets of the handover flow to the new BS following sending a Forwarding Start message which acts as a Notify ACK message. During a packet-forwarding period, the old BS may still receive packets from the source due to travelling time difference between a Redirect message and Notify message. The packets arrived during the period are also forwarded to the new BS. After finishing packet forwarding, the old BS sends the Forwarding End message to signal the end of packet forwarding. If the old BS has no packet to be forwarded, then the old BS sends the Forwarding End message directly without Forward Start message. The Forwarded End message causes the new BS to release the service lock for redirected packets.

If the route between old and new BSs is congested or the Forwarding End message is not delivered due to an error, the redirected packets will be locked for an unacceptable period or forever. To avoid this situation, the new BS sets a maximum waiting period for a service lock. The period is decided considering the average round trip from the old BS when it receives the Greeting message from MH. The waiting period starts after sending the Notify message and resets whenever a message or packet arrives from the old BS. The first maximum waiting period is determined by the round trip time because it should consider the travelling time of the Notify message to old BS, while following waiting periods by the half round trip time.

To support the service strategy above, the downlink scheduler processes the Forwarded End message and uses its arrival time as the arrival time of the redirected packets which have arrived during the service lock period, while it does not adjust the arrival times of the forwarded packets.
at the new BS. Using this forwarding-aware protocol, the out-of-order packet delivery can be remedied. As shown in Figure 7.10, the redirected packets ①②③ are served following servicing the forwarded packets regardless of their arrival times.

Figure 7.10 The proposed handover protocol

7.5.4 Simulation results

We simulated the proposed handover protocol with the same configuration used in §7.2.3. In the simulation, flow 1 and 2 experienced out-of-order packet delivery after its handover among the 7 handover flows as can be seen in Figure 7.11 (a) and (b). Our proposed handover protocol remedied the out-of-order deliveries, as can be seen Figure 7.11 (c) and (d). Figure 7.12 shows that end-to-end delay of both cases. As can be seen in Figure 7.12 (a) and (b), the first redirected packet which arrives during the packet forwarding period has lower delay than those of forwarded packets.
Figure 7.11 The packet numbers of flow 1 and 2 arrived at corresponding MHs

Figure 7.12 End to End delay of flow 1 and 2
7.6 Summary

This chapter has addressed the issues regarding a handover when a PFD-type scheduler is adopted in a BS. Unbounded fairness of WFQ and out-of-order packet delivery are identified. The former occurs due to packets being forwarded or removed without consuming outgoing link bandwidth even though the packets have influenced the virtual time. The latter occurs due to the travelling time gap between forwarded packets and re-directed packets. We propose solutions for PCAWFQ and PCAWFQ' to remedy the unbounded fairness along with handover compensated WFQ. We also propose a handover protocol to fix the out-of-order packet delivery, which co-operates with a rate-based scheduler. The simulation shows the behaviour of WFQ, a representative PFD-type scheduler during handover. The results demonstrate an increase in the fairness gap of WFQ at each occurrence of handovers. The bounded fairness and correctly ordered packet delivery are observed in the simulation studies using the proposed compensated scheme and the proposed handover protocol respectively.
Chapter 8

Conclusions

This chapter summarizes the work presented so far reiterating the principal points, and draws some contributions. The way towards opportunities for further investigation is also discussed here.

8.1 Thesis summary

This dissertation deals with QoS issues in a cellular packet switched network. Chapter 2 described what kinds of cellular data networks have been developed and will be deployed to provide diverse QoS for integrated traffic in wireless environment. The survey showed us that future cellular data networks require higher data rates than current commercially available ones to accommodate the integrated services, and a packet switched scheme will be suitable rather than current widespread circuit switched schemes to utilise the limited wireless bandwidth efficiently.

A range of technology is required to provide integrated traffic with committed QoS in micro/pico cellular packet switched networks. The two big issues among them are a reservation protocol and practical resource distribution. This dissertation has focused on the latter issue and set the four aims of this work to provide committed QoS for integrated traffic in micro/pico cellular packet switched networks:

1) Designing a service architecture to support the guaranteed QoS
2) A packet scheduling discipline for downlink service
3) A multiple access protocol embodied packet scheduling disciplines for uplink service
4) Investigating handover related problems associated with packet scheduling.

Chapter 3 drew on recent related work regarding those four topics, and analysed their advantages and disadvantages with respect to our theme. The main body of this dissertation (Chapter 4 - 7) has outlined our approaches to achieve the research aims. Chapter 4 described the framework of our proposed service architecture, which can serve 4 kinds of traffic classes: Periodic, Absolute, Weighted and Best-effort. Chapter 5 presented the CAWFQ and its packetized version PCAWFQ, which is a general-purpose scheduling discipline for a varying bandwidth link but can be used in the proposed service architecture. Chapter 6 described a multiple access protocol called FQMA, which accommodates the uplink scheduling disciplines: PEDQ and PCAWFQ'. Chapter 7 discussed the handover-related issues occurring when the
scheduling disciplines are applied to a BS. We investigate how the handover affects the performance of a scheduler.

8.2 Research Contributions

In assessing the contributions of this work, it is first necessary to assess where it stands in relation to the previous work. Thus, in this section, we compare main components of this work with their related previous work and present the contributions of this work.

8.2.1 Comparison with the previous work

This work contains two main research areas: scheduling discipline and multiple access protocol. The most closely related contributions in the scheduling discipline area are WFQ [71] and SFQ [43]. WFQ is a rate-based queuing where the share rate of a flow is guaranteed all the time. Thus, if an output link is fixed, then it provides a flow with a guaranteed bandwidth service and also commits a bounded delay when the source of a flow is complied with the token bucket filter [92]. However, as the WFQ is designed for a constant output link, it does not operate properly in a varying output link: i.e. it produces unfair service. SFQ is designed to overcome the flaw of WFQ, but it has still two shortcomings in supporting diverse QoS in a cellular packet switched network. Firstly, since SFQ uses only share rates as the reference of flows’ requirements, the guaranteed bandwidth rate of a flow varies under conditions of fluctuating available bandwidth even though it can produce fair service among flows regardless of the bandwidth fluctuation. Thus, with the simple rate-based service, a flow that should require a constant or minimum bandwidth cannot be served in a varying rate link. By contrast, CAWFQ in this work can provide a flow with a constant bandwidth – which can be guaranteed under certain conditions - by developing a concept that merges the rate-based queuing and conditional priority-based queuing\(^6\). Secondly, SFQ cannot accommodate various kinds of services depending on the traffic characteristic of a flow or user preference. For instance, periodic traffic is intolerant of delay, so all packets should be served within their lifetimes in order not to lose them. By contrast, some applications dealing with random traffic such as FTP applications are unaffected by packet delay and the average bandwidth is a more important factor. Thus, for the diverse QoS, multiple schedulers should be needed to provide various services so that a user can choose a service depending on its traffic characteristics. In this work, we proposed a service architecture that can

\(^6\) A priority in a conditional priority-based queuing works on a conditional status, while that in a priority-based queuing works absolutely.
support diverse QoS by adopting several different schedulers depending on traffic types and the direction of a wireless link as shown in Table 8.1.

**Table 8.1 The scheduling disciplines used for each class**

<table>
<thead>
<tr>
<th>Class type</th>
<th>Direction</th>
<th>Scheduling Discipline</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Periodic</strong></td>
<td>Downlink</td>
<td>FCFS</td>
</tr>
<tr>
<td></td>
<td>Uplink</td>
<td>PEDQ</td>
</tr>
<tr>
<td><strong>Absolute, Weighted, Best-Effort</strong></td>
<td>Downlink</td>
<td>PCAWFQ</td>
</tr>
<tr>
<td></td>
<td>Uplink</td>
<td>PCAWFQ'</td>
</tr>
</tbody>
</table>

For the second research area, C-PRMA [13] is the most closely related work. C-PRMA adopted a scheduler called PR (Polling Register) which serves flows based on their packet inter-arrival time and packet discard priorities, latter of which is applied when available slots are not enough. Thus it is possible to provide not only fixed bandwidth service, but also a priority-based service using the packet discard priority. However, it cannot support fair service as well as rate-based service. There are two fairness problems in the C-PRMA: among flows in the same priority group and between different priority groups. The former occurs because there is no fair mechanism to determine the owner when flows with the same priority compete each other for a slot. The latter occurs because the priority is locally applied to a specific slot location, not with global view. For instance, although two flows have the same priority and allocation, one flow may be served more than the other flow if the other flow had met higher priority flows more frequently when competing for slots. By contrast, the proposed service scheme supports fair service without both the problems since each traffic class is decoupled each other so that a class cannot encroach the others. In addition, each class is served with a specific scheduling discipline considering the class traffic characteristic. The decoupling between the periodic and random traffic is performed by the STS, giving higher priority to periodic traffic as long as allocating slots to random traffic does not degrade the service performance of periodic traffic. The decoupling among the random traffic classes is provided by the PCAWFQ discipline which is designed to support fair services between them using two virtual times that indicate the received service amount of their corresponding classes. PCAWFQ also maintains the fairness among the same class flows, forcing a new flow to have its reference virtual time computed based on its associated virtual time.
8.2.1 Contributions

The major contributions of this research can be listed as follows:

♦ **Pioneering new service architecture that can provide integrated traffic with guaranteed QoS using multiple scheduling disciplines in a micro/pico cellular packet switched network, taking into account available bandwidth fluctuation and handover**

We proposed a new service architecture to accommodate multiple scheduling disciplines for integrated traffic. Previous work tried to support QoS using a scheduling discipline. The research represented in this thesis, I discovered that a system with multiple scheduling disciplines is very efficient handling various kinds of traffic. As can be seen in Table 8.1, 4 different scheduling schemes are used to maximise service performance of each traffic group. Each scheme has an own unique service policy supporting the characteristics of the traffic that it aims to serve. Especially, scheduling disciplines for uplink and downlink are designed in a different way. One of the difficulties in implementing multiple scheduling schemes is to assign slots to each scheduler. STS and SAC were developed for that purpose. Both assign the current slot to one of the four schedulers, determining the service priority depending on traffic characteristics and loads.

♦ **Developing a packet scheduling scheme called PCAWFQ which has a mixture of the functions of rate-based and conditional priority-based discipline**

A packet scheduling discipline is one of the principal issues in the practical resource distribution area. While packet scheduling disciplines for a fixed network have been heavily researched especially for high speed networks, those for a cellular packet switched network have received little attention. Most scheduling disciplines have been designed assuming that the output link bandwidth is fixed and flows being served do not move out during their backlogged periods. However, they are not the cases in a cellular network. In addition, a cellular network is relatively slow compared to a fixed network, so the service performances of a scheduler such as fairness and delay bound are more important rather than low complexity that is especially necessary in a high-speed wired network. To overcome those problems, the CAWFQ scheduling discipline and its a packet by packet version, PCAWFQ were developed.

♦ **Designing a slot-based multiple access protocol called FQMA which can decouple periodic and random traffic, accommodating multiple scheduling disciplines**
Actually, developing a multiple access protocol frame was not what was planned. Further my interest was to support multiple scheduling disciplines in a multiple access protocol, so that an existing multiple access protocol could be used for my work. However, during performance testing of several candidates, it was realised that a slot-based protocol is more efficient than a frame-based scheme in decoupling traffic if a proper algorithm for assigning reservation slots is provided. Although C-PRMA is a slot-based protocol, it cannot support the decoupling among different traffic classes. Furthermore, it was realised that an uplink/downlink sharing scheme can produce higher performance gain than a partitioning scheme that was intended to be used. Thus, also a mediator called SAC which is required for both links to share a common channel was developed.

- **Developing a scheduling scheme called PEDQ for uplink periodic traffic service using remaining lifetime of a packet**

  This is an enhanced version of PR scheduling scheme of C-PRMA. Both use the lifetimes of packets in scheduling packets. The difference is that PEDQ always schedules packets in increasing order of their remaining lifetimes, while PR does not since an allocated slot for a packet cannot be given way to following packets with shorter remaining lifetimes once allocated. Thus, a packet with longer remaining lifetime could be served earlier than a packet with shorter one. It results in increasing packet-drop rate in over-loaded periods. Consequently, PEDQ performs better than PR.

- **Introducing the proxy packet-scheduling concept to serve distributed packets in MHs as if they are queued in a single queue**

  The proxy packet concept is efficient to serve distributed packets in MH queues. It enables a scheduling scheme for downlink to be converted into the one for uplink, so downlink and uplink can use the same scheduling discipline. This property is important to support a consistent QoS in both links. One problem in the proxy packet concept is that it cannot emulate the exact packet generation time, so in the case of PCAWFQ the delay bound cannot be guaranteed even though throughput bound is guaranteed.

- **Discovery of the handover effects that affect the scheduling performance in a BS and proposals to remedy the problems.**

  It was discovered that a handover causes unfair service in an old BS that adopts a PFD type scheduler. We proposed a compensated scheme to protect the service distortion in the BS.
Extending a handover protocol which can sense packet forwarding to adjust out-of-order packet delivery

A typical handover protocol is interested in fast and seamless migration of a host. It does not concern the packet service performed in lower level. However, it was found that the communication between scheduler and a handover protocol is needed to remove the out-of-order packet delivery that causes due to the one hop more travelling of forwarded packets compared to the redirected packets. Thus, a packet-forwarding aware handover protocol was proposed to fix the out-of-order packet delivery.

8.3 Future work

Although the service architecture proposed in this dissertation has shown its potential and validity on cellular packet switched services, it can be still improved further in some aspects. However, the further research investigation for the improvements is too massive to add into current work considering the scope of Ph.D. We here just suggest the directions regarding the further work.

8.3.1 A clustered scheduling system

Designing the proposed service disciplines, we focused on the services within one cell even though the handover effect was considered. Thus the service disciplines were designed to operate in an independent BS without communicating with others located neighbour cells in spite of flows moving around among the cells. Conceptually, a BS has potential relationship with those neighbour schedulers. Therefore we could consider a system of PCAWFQs in a set of neighbour BSs as a single system with a slow queue (high latency fixed delay in the handover). With such a system, a flow can obtain an allocation in advance before it moves to the cell and we could also fundamentally fix the unfairness problem on handover. PCAWFQ could be evolved to an element of such a clustered scheduling system.

8.3.2 Reservation request sharing

FQMA is designed such that random and periodic traffic have their separated reservation phases to prevent the reservation request of random traffic from degrading the successful requesting rate of periodic traffic. The segregation approach is essential to decouple both traffic services, which commits the limit of packet drop rate for periodic traffic. However, if an MH has both traffic sources which try to request their reservations at the same time, it is efficient for both
sources to share a same reservation request slot. In this way, the performance of multiple access protocol could be improved.

8.3.3 Interface with a reservation protocol

For a scheduling discipline to negotiate with an application regarding the service commitment, we need a reservation protocol in the middle of both. While several leading reservation protocols such RSVP for a fixed network have been developed, the reservation protocol for a mobile network has not been actively researched – however, there is one called MRSVP which is still developing. The IETF transport working group is producing a functional model [37] for the purpose which includes a scheduling discipline for RSVP on shared and switched LAN infrastructure. We cannot directly apply it to the cellular network due to handover even though wireless link is shared infrastructure. We did not explore the framework that accommodates our proposed service scheme for supporting a reservation protocol in a mobile network.

8.4 Limitation and Concluding Remarks

In this dissertation, we have taken a set of insights into QoS regarding cellular data networks and proposed a new approach. In doing so, we have developed not only several packet scheduling disciplines suitable for the service architecture, but also multiple access protocol for uplink service. In addition, the problems of packet service regarding handover have been analyzed and a packet forwarding-aware handover protocol introduced. The proposed multiple access protocol has been designed for a micro-cellular network, so it cannot be used in a macro-cellular network where there are long round trip delays. However, other parts of the proposed system such as scheduling disciplines and the proxy packet concept can be applied to a macro-cellular data network. In addition, we assumed that the effect of errors in the wireless link were negligible, which is probably a reasonable assumption, given that short distance and low cell loading normally found there. Further work to validate these assumptions is needed.

This research, although limited, does demonstrate feasible service architecture for providing integrated traffic service in a micro/pico cellular packet switched network for the next generation. ITU and ETSI are making the standards for the third generation mobile system. At the moment, we cannot draw the whole picture of the service, but it is certain that circuit- and packet-oriented service will be supported. The proposed service architecture can be applied to a scheduling scheme on a packet-oriented service in the next generation mobile system or can be implemented
as a pure packet-oriented system. It is recommended that further research be done to enable the proposed service model to achieve its potential.
Appendix

A.1 Proof of Lemma 2

Proof: A leading sub period, \((\tau, t]\) can be classified into three cases: deficit, semi-deficit and surplus period.

Deficit period: \(\int_{\tau}^{t} C(t)dt < (t - \tau) \cdot \sum_{i \in B(\tau, \tau') \cap A} r_i\)

Substituting \(t = \tau^+\) as \(t_s\) into (11) and (12) and subtracting between them, we have

\[ v_a(t) - v_w(t) = \left( \int_{\tau^+}^{t} C(t)dt \right) \sum_{i \in B(\tau^+, t) \cap A} r_i \]

By the lemma 1, \(v_a(\tau^+) - v_w(\tau^+) = 0\). Therefore, we have \(v_a(t) \geq v_w(t)\)

Semi-deficit period: \((t - \tau) \cdot \sum_{i \in B(\tau, \tau') \cap A} r_i \leq \int_{\tau}^{t} C(t)dt < (t - \tau) \cdot \sum_{i \in B(\tau, \tau') \cap W} r_i\)

Substituting \(t = \tau^+\) as \(t_s\) into (11) and (12) and subtracting between them, we have

\[ v_a(t) - v_w(t) = \frac{(t - \tau) \cdot \sum_{i \in B(\tau^+, t) \cap A} r_i - \int_{\tau}^{t} C(t)dt}{\sum_{i \in B(\tau^+, t) \cap A} r_i} \]

From the definition of semi-deficit period, \(\int_{\tau}^{t} C(t)dt - (t - \tau) \sum_{i \in B(\tau, \tau') \cap W} r_i < (t - \tau) \sum_{i \in B(\tau, \tau') \cap W} r_i\)

By the lemma 1, \(v_a(\tau^+) - v_w(\tau^+) = 0\)

Therefore we have:

\[ v_a(t) - v_w(t) = (t - \tau) \sum_{i \in B(\tau^+, t) \cap A} r_i > (t - \tau^+) \sum_{i \in B(\tau^+, t) \cap W} r_i + (t - \tau^+) \sum_{i \in B(\tau^+, t) \cap W} r_i = 0 \]

Therefore, we have \(v_a(t) > v_w(t)\)

Surplus period: \(\int_{\tau}^{t} C(t)dt \geq (t - \tau) \cdot \sum_{i \in B(\tau, \tau')} r_i\)

Substituting \(t = \tau^+\) as \(t_s\) into (13) then we have \(v_a(t) = v_w(t)\)

\[ \square \]

A.2 Proof of Lemma 3

Proof: We will prove the lemma by two steps. Firstly, we will show that \(v_a(t_m)\) and \(v_w(t_m)\) are non-decreasing functions, where \(v_a(t_m)\) and \(v_w(t_m)\) represent approximated linear virtual
functions that connects \( t_m \) for \( m = 0,1,2,\ldots \) where \( t_m \) is the start time of \( m^{th} \) membership-step period. Next, we will prove that \( v_a(t) \) and \( v_w(t) \) are non-decreasing during any membership-step period.

Step 1: Consider virtual times over a membership-step period \((t_m,t_{m+1}]\), replacing \((t_s,t_e] \) with \((t_m,t_{m+1}]\) in (11) and (13). Since the link rate, \( C(t) \) and the share rate \( r \) are always non-negative and \( t_{m+1} - t_m > 0 \), the right terms of eq (11) and (13) are also non-negative. It follows that

\[
v_a(t_{m+1}) - v_a(t_m) \geq 0 \text{ for } m = 0, 1, 2, \ldots
\]

According to the initial condition of \( v_a \), \( v_a(t_m) \leq v_a(t_{m+1}) \) for \( m = 0, 1, 2, \ldots \)

Therefore, \( v_a(t_m) \) are non-decreasing functions.

The right terms of eq (12) and (13) are always non-negative. It follows that

\[
v_w(t_{m+1}) - v_w(t_m) \geq 0 \text{ for } m = 0, 1, 2, \ldots
\]

According to the boundary condition of \( v_w \)

\[
v_w(t_m) \leq v_w(t_{m+1})
\]

Therefore, \( v_w(t_m) \) are non-decreasing functions.

Step 2: Firstly, we can obtain that \( v_a \) and \( v_w \) are non-decreasing during any surplus membership-step sub-period via (16) since \( C(t) \) is always non-negative. Next, we will prove that the virtual times, \( v_a \) and \( v_w \) are non-decreasing during any semi-deficit or deficit membership-step sub-period, showing that the right terms of (14) and (15) are always non-negative for any membership-step sub-period \((t, T]\) where \( t_m < t < T \leq t_{m+1} \).

For eq (14), consider the following four possible combinations

Case 1: \( \int_{t_s}^{t_e} C(t)dt \leq \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \) and \( \int_{t_s}^{t_e} C(t)dt \leq \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \)

the right term of (14) is non-negative since \( C(t) \geq 0 \) for all \( t \) and \( \int_{t_s}^{t_e} C(t)dt > \int_{t_s}^{t_e} C(t)dt \)

Case 2: \( \int_{t_s}^{t_e} C(t)dt \leq \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \) and \( \int_{t_s}^{t_e} C(t)dt > \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \)

the right term of (14) is non-negative since \( \int_{t_s}^{t_e} C(t)dt \geq t \int_{t_s}^{t_e} C(t)dt > \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \)

Case 3: \( \int_{t_s}^{t_e} C(t)dt > \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \) and \( \int_{t_s}^{t_e} C(t)dt > \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \)

the right term of (14) is non-negative since \( \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) > \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \)

Case 4: \( \int_{t_s}^{t_e} C(t)dt > \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \) and \( \int_{t_s}^{t_e} C(t)dt < \sum_{i \in B(t_s,t_e) \setminus A} r_i(t-t_s) \)


the right term of (14) is non-negative since

\[ \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) > \sum_{i \in B(t_s, T) \cap A} \int_{i} C(t) dt \]

In eq (15), from the definition of a membership-step period, there is no such case that

\[ \int_{i} C(t) dt \leq \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \quad \text{and} \quad \int_{i} C(t) dt > \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \]

therefore, consider the following possible three cases.

Case 1: \[ \int_{i} C(t) dt \leq \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \quad \text{and} \quad \int_{i} C(t) dt \leq \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \]

the right term of (15) is zero

Case 2: \[ \int_{i} C(t) dt > \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \quad \text{and} \quad \int_{i} C(t) dt > \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \]

the right term of (15) is

\[ \int_{i} C(t) dt - \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) - \left( \int_{i} C(t) dt - \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \right) \]

\[ = \int_{i} C(t) dt - \sum_{i \in B(t_s, T) \cap A} r_i (t - t) > 0 \]

because, according to the definition of membership-step period,

\[ \int_{i} C(t) dt > \sum_{i \in B(t_s, T) \cap A} r_i (t - t) \quad \text{if} \quad \int_{i} C(t) dt > \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \]

Case 3: \[ \int_{i} C(t) dt > \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \quad \text{and} \quad \int_{i} C(t) dt < \sum_{i \in B(t_s, T) \cap A} r_i (t - t_s) \]

Apparently, the right term of (15) is non-negative

For the above cases, we conclude that \( v_a \) and \( v_w \) are non-decreasing during any membership-step sub-period. Thus, lemma 3 follows from steps 1 and 2. \( \square \)

A.3 Proof of Lemma 4

Proof: Combining (17) and (1)(2), we have

\[ \sum_{j \in B(t_1, t_2) \cap A} W_j (t_1, t_2) = \sum_{j \in B(t_1, t_2) \cap A} r_j \cdot (t_2 - t_1) \] (A.1)

\[ \sum_{j \in B(t_1, t_2) \cap W} W_j (t_1, t_2) = \sum_{j \in B(t_1, t_2) \cap W} r_j \cdot (t_2 - t_1) \] (A.2)

where \((t_1, t_2)\) is a membership-step period

Since the CAWFQ is a fluid work-conserving server and \( B(t_1, t_2) \) does not change, the following must hold for any period \((t, \tau)\) where \( t_1 < \tau < t \leq t_2 \) from (A.1)(A.2)

\[ \sum_{j \in B(\tau, \tau) \cap A} W_j (\tau, t) = \sum_{j \in B(\tau, \tau) \cap A} r_j \cdot (t - \tau) \] (A.3)
\[ \sum_{j \in B(\tau,t) \setminus W} W_j(\tau,t) = \sum_{j \in B(\tau,t) \setminus W} r_j \cdot (t - \tau) \] (A.4)

Let the increases of a- and w-virtual time during \((\tau, t]\) be \(\theta^a\) and \(\theta^w\), respectively as followings

\[ \frac{W_j(\tau,t)}{r_j} = \frac{W_j(\tau,t)}{r_j} = \theta^a \quad i,j \in B(\tau,t) \cap A \] (A.5)

\[ \frac{W_j(t,\tau)}{r_j} = \frac{W_j(t,\tau)}{r_j} = \theta^w \quad i,j \in B(\tau,t) \cap W \] (A.6)

Combining (A.3)-(A.4) and (A.5)-(A.6), we conclude that

\[ \theta^a = \theta^w = t - \tau, \]

\[ v_a(\tau) - v_a(t) = v_w(\tau) - v_w(t) \] (A.7)

From the definition of virtual time, \(v_a(0^+) = v_w(0^+)\) (A.8)

Therefore, the lemma 4 now follows from (A.7) and (A.8) \(\square\)

A.4 Proof of Theorem 1

**Proof:** Let us first prove the relationship for AC. A packet could arrive before or after the departure of the last arrival packet. Therefore we will distinguish two cases:

Case 1: \(a_j > d_{k-1}' \)

\[ w_k^a( a_j', d_j') = v_a( a_j') - v_a( a_j) = L_k/r_k \] (A.9)

Case 2: \(a_j < d_{k-1}' \)

\[ w_k^a( a_j', d_j') = v_a( a_j') - v_a( a_j) = L_k/r_k \] (A.10)

Combining (A.9) and (A.10)

\[ v_a(d_j') - \max(v_a(d_{k-1}'), v_a(a_j')) = L_k/r_k \]

From (19), \(v_a(d_j') = F_{k}^i\), thus we have

\[ F_{k}^i = \max(F_{k-1}^i, v_a(a_j')) + L_k/r_k \]

For the WC, similarly, we have

\[ v_w(d_j') = \max(v_w(d_{k-1}'), v_w(a_j')) + L_k/r_k \]

From (19), \(v_w(d_j') = F_{k}^i\), thus we have

\[ F_{k}^i = \max(F_{k-1}^i, v_w(a_j')) + L_k/r_k \]

Thus, Theorem 1 follows \(\square\)
References


[94] I. J. Wakeman, "Congestions Control for Packetised Video in the Internet," PhD these, University College London, 1995


