A QoS-Sensitive SDMA-TDMA Access in Broadband Fixed Wireless Networks

Qiang Wang

A Thesis

In

The Department

Of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

September 2002

0-612-77692-1

ABSTRACT

A QoS-Sensitive SDMA-TDMA Access in Broadband Fixed Wireless Networks

Qiang Wang

This thesis addresses both real-time and non-real time supports of communication services in a broadband fixed wireless network with Probing Process. In such a network, each cell is divided into multiple sectors, each of them served by a sector antenna colocated with the base station (BS), and user terminals use directional antennas mounted on the rooftops and pointed to their respective BS antennas. With the use of directional antennas to suppress interference, the Probing Process is proposed to improve data packet transmission without information exchange and coordination among base stations. The Probing Process detects available slots unused by other sectors to provide a higher capacity in sectorization cells. To support real time and non-real time traffic and efficiently implement Probing Process in FDD method, the base station (BS) adopts a demand-assignment channel access protocol named Prioritized Access with Centralized Polling Control that shares the bandwidth for different requests of real-time and non-real time users. The simulation results presented in this thesis show the quantitative improvement for voice traffic and data traffic with Probing Process.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis supervisor, Prof. Anjali Agarwal, for her invaluable guidance and support without which it would have been impossible to complete this thesis. She was a constant source of inspiration and confidence throughout my study.

I wish to thank my friends, especially Lu Yang, GaoFang Lao, LiKe Zhang and Hai Lin, for their continuous encouragement and support.

My special thanks to my parents, sisters and extended family members who have always been by my side whenever it is good times and bad.

I am also in debt to my wife, XiaoYan Yang, who did all the housework.without any complaint and endured the same stress I experienced in my study period.

Finally, I would also like to extend my gratitude to faculty and staff members of the department of Electrical and Computer Engineering of Concordia University who helped provide a great work environment.

Table of Contents

List of Figures

CHAPTER 1 INTRODUCTION

1.1 Introduction to Fixed Wireless Access Network

As the growth of the Internet [1], broadband access has received much attention lately [2]-[4]. Furthermore, much of this attention has been focused on providing wired broadband access using existing copper line [5] or coaxial cable [6]. Wireless broadband access is emerging as an attractive alternative or complement [4], [7]-[9]. Customers are expecting high-quality, reliability and easy access to high-speed communications from homes and small businesses. High-speed services are needed in the very near future for: 1) accessing the World Wide Web for information and entertainment; 2) providing data rates comparable to local area networks for telecommuters to access their computer equipment and data at the office; and 3) supporting various traffic with quality-of-service (QoS) guarantees as the growth occurs for broadband multimedia communications involving digital audio and video [10].

We consider the wireless approach for broadband multimedia services in this thesis. Specifically, we focus on a fixed broadband packet-switched network (Figure 1.1) using time-division-multiple-access (TDMA) techniques with user data rates of 10Mb/s, link lengths typically less than 10 km, and operating frequency in the range of 1-5 GHz. The whole service area in the network is divided into cells. Each cell is further divided into multiple sectors, each of which is covered by a sector antenna co-located with a base station (BS) at the center of the cell. Sector antennas are also called BS antennas because of their co-location. Terminals (users) use directional antennas mounted on the rooftop and pointed to their respective BS antennas. The beam width of each BS antenna should be just wide enough to serve the whole sector, while a terminal antenna can have a smaller beam width to suppress interference. The ratios of front-to-back lobe gain (abbreviated by FTB ratio below) for BS and terminal antennas are assumed to be finite. Time is slotted such that a packet can be transmitted in each slot. In addition, the downlink and uplink between terminals and BS can be provided by frequency-division duplex (FDD).

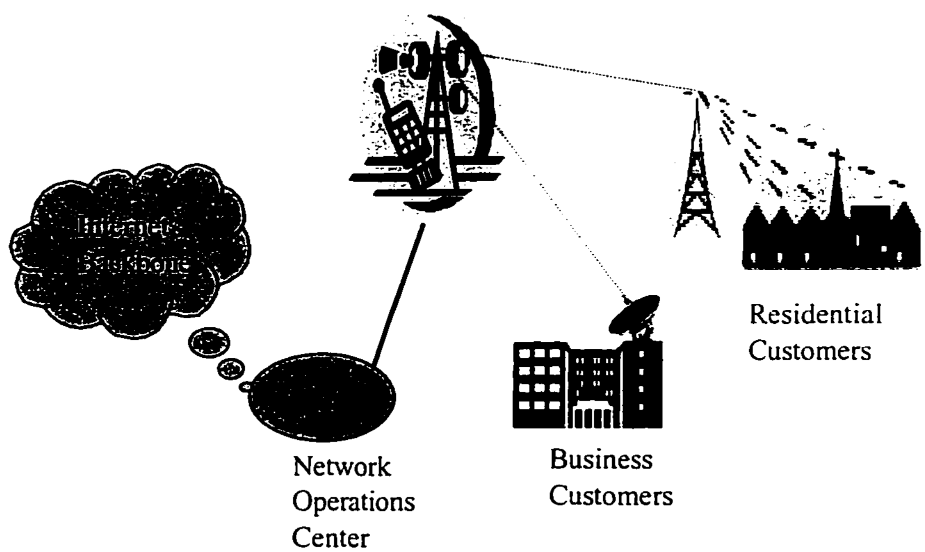


Figure 1.1 Fixed wireless Access Networks

1.2 Resource Allocation and MAC layer protocols

In wireless networks, microwave spectrum is not only expensive but inherently limited, so efficient strategies for reusing frequencies and managing cochannel interference are critically important. Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) are three major access techniques used to share the available bandwidth in a wireless communication system. In addition to FDMA, TDMA, and CDMA, two other multiple access schemes are used for wireless communications. These are packet radio (PR) and space division multiple access (SDMA). These techniques can be grouped as narrowband and wideband systems, depending on how the available bandwidth is allocated to the users. The duplexing technique of a multiple access system is usually described along with the particular multiple access scheme [11].

1.2.1 Frequency Division Multiple Access (FDMA)

Frequency Division Multiple Access (FDMA) assigns individual channels to individual users. It can be seen from Figure 1.2 that each user is allocated a unique frequency band or channel. These channels are assigned on demand to users who request service. During the period of the call, no other user can share the same frequency band. In FDD systems, the users are assigned a channel as a pair of frequencies, one frequency is used for the forward channel, while the other frequency is used for the reverse channel.

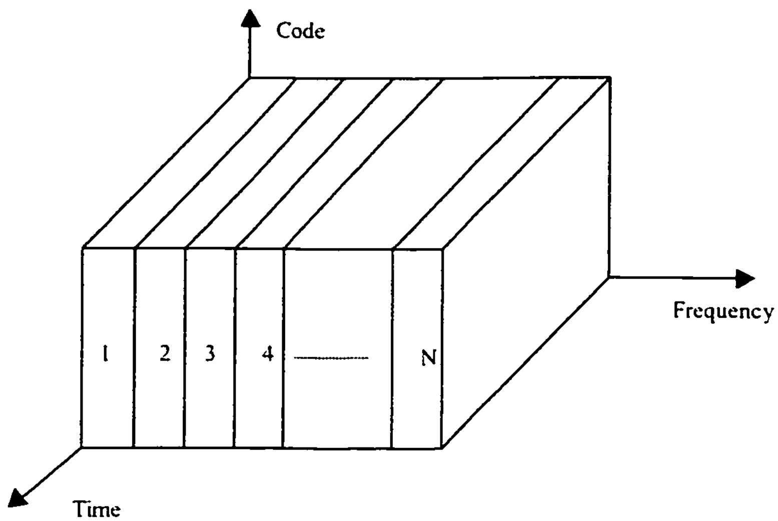


Figure 1.2 FDMA where different channels are assigned different frequency bands

1.2.2 Time Division Multiple Access (TDMA)

Time Division Multiple Access (TDMA) systems divide the radio spectrum into time slots, and in each slot only one user is allowed to either transmit or receive. It can be seen from Figure 1.3 that each user occupies a cyclically repeating time slot, so a channel may be thought of as particular time slot that reoccurs every frame, where N time slots comprise a frame. TDMA systems transmit data in a buffer-and-burst method, thus the transmission for any user is noncontinuous. This implies that, unlike in FDMA systems that accommodate analog FM, digital data and digital modulation must be used with TDMA. The transmission from various users is interlaced into a repeating frame structure. It can be seen that a frame consists of a number of slots. Each frame is made up of a preamble, an information message, and tail bits. In TDMA/TDD, half of the time slots in the frame information message would be used for the forward link channels and half would be used for reverse link channels. In TDMA/FDD systems, an identical or similar frame structure would be used solely for either forward or reverse transmission, but the carrier frequencies would be different for the forward and reverse links. In general, TDMA/FDD systems intentionally induce several time slots of delay between the forward and reverse time slots of a particular user, so that duplexers are not required in the subscriber unit.

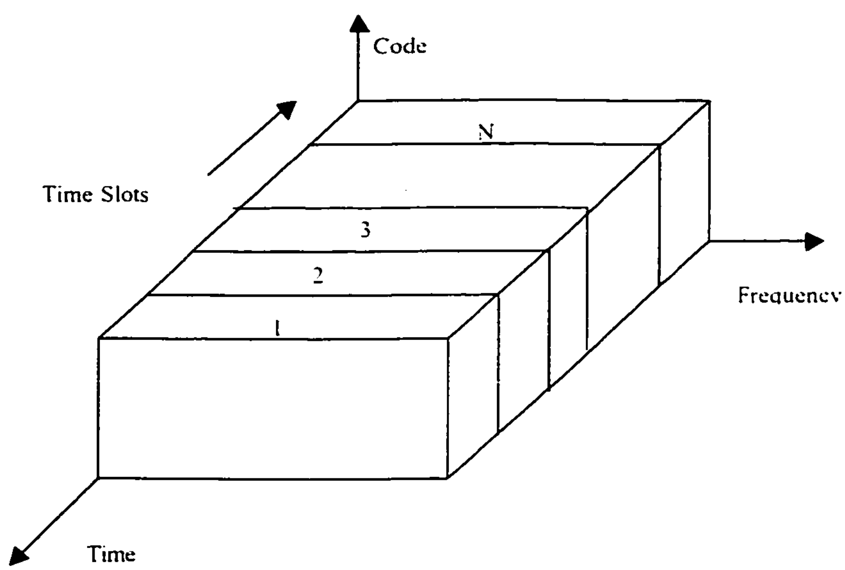


Figure 1.3 TDMA scheme where each channel occupies a cyclically repeating time slot

In a TDMA frame, the preamble contains the address and synchronization information that both the station and the subscribers use to identify each other (figure 1.4). Guard times are utilized to allow synchronization of the receivers between different slots and frames. Different TDMA wireless standards have different TDMA frame structures. The features of TDMA include the following:

- TDMA shares a single carrier frequency with several users, where each user makes use of non-overlapping time slots. The number of time slots per frame depends on several factors, such as modulation technique, available bandwidth, etc.
- Data transmission for users of a TDMA system is not continuous, but occurs in bursts. This results in low battery consumption, since the subscriber transmitter can be turned off when not in use (which is most of the time).
- Because of discontinuous transmission in TDMA, the handoff process is much simpler for a subscriber unit, since it is able to listen for other base stations during idle time slots. An enhanced link control, such as that provided by Movable Assisted Handoff (MAHO), can be carried out by a subscriber by listening on an idle slot in the TDMA frame.
- TDMA uses different time slots for transmission and reception, thus duplexers are not required. Even if FDD is used, a switch rather than a duplexer inside the subscriber units is all that is required to switch between transmitter and receiver using TDMA.
- Adaptive equalization is usually necessary in TDMA systems, since the transmission rates are generally very high as compared to FDMA channels.
- In TDMA, the guard time should be minimized. If the transmitted signal at edges of a time slot are suppressed sharply in order to shorten the guard time,

the transmitted spectrum will expand and cause interference to adjacent channels.

- High synchronization overhead is required in TDMA systems because of burst transmissions. TDMA transmissions are slotted, and this requires the receivers to be synchronized for each data burst. In addition, guard slots are necessary to separate users, and this results in the TDMA systems having larger overheads as compared to FDMA.
- TDMA has an advantage in that it is possible to allocate different numbers of time slots per frame to different users. Thus bandwidth can be supplied on demand to different users by concentrating or reassigning time slots based on priority.

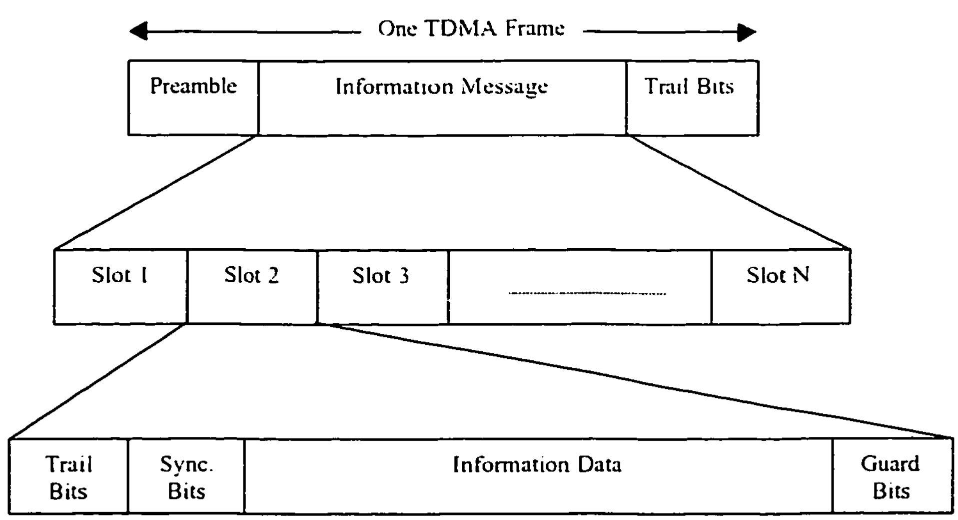


Figure 1.4 TDMA frame structure

1.2.3 Code Division Multiple Access (CDMA)

In Code Division Multiple Access (CDMA) systems, the narrowband message signal is multiplied by a very large bandwidth signal called the spreading signal. The spreading signal is a pseudo-noise code (PN code) sequence that has a chip rate which is an order of magnitudes greater than the data rate of the message. All users in a CDMA system, as seen from Figure 1.5, use the same carrier frequency and may transmit simultaneously. Each user has its own pseudoransom codeword which is approximately orthogonal to all other codewords. The receiver performs a time correlation operation to detect only the specific desired codeword. All other codewords appear as noise due to decorrelation.For detection of the message signal, the receiver needs to know the codeword used by the transmitter. Each user operates independently with no knowledge of the other users.

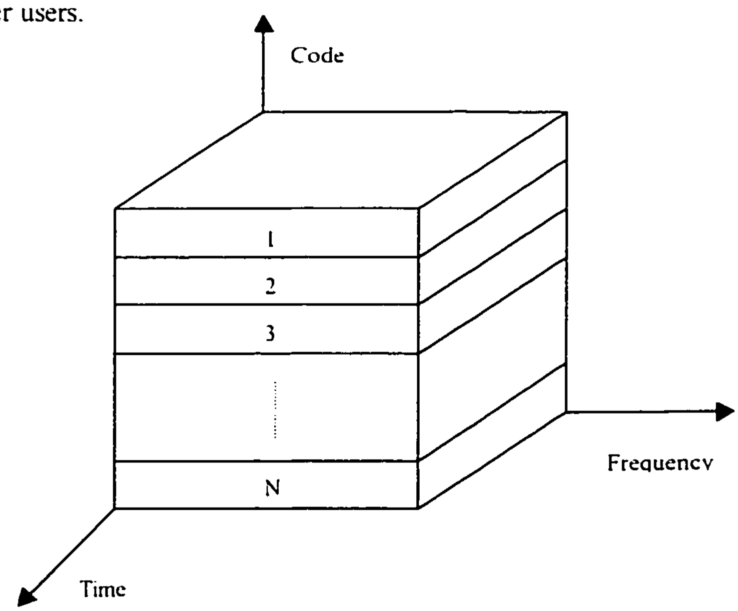


Figure 1.5 CDMA in which each channel is assigned a unique PN code which is orthogonal to PN Codes used by other users

1.2.4 Space Division Multiple Access (SDMA)

Space division multiple Access (SDMA) controls the radiated energy for each user in space. It can be seen from Figure 1.6 that SDMA serves different users by using spot beam antennas. These different areas covered by the antenna beam may be served by the same frequency (in a TDMA or CDMA system) or different frequencies (in an FDMA systems). Sectorized antennas may be thought of as a primitive application of SDMA. Adaptive antennas is used to simultaneously steer energy in the direction of many users at once and appear to be best suited for TDMA and CDMA base station architectures.

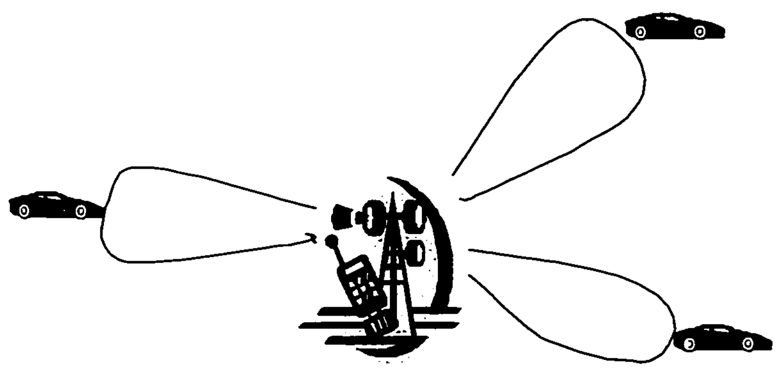


Figure 1.6 A spatially filtered base station antenna serving different users by using spot beams

1.2.5 Packet Reservation Multiple Access (PRMA)

Packet reservation multiple access (PRMA) is a transmission protocol proposed by Goodman, et.aL, for packet voice terminals in a cellular system. PRMA is a time division multiplex (TDM) based multiple access protocol that allows a group of spatially dispersed terminals to transmit packet voice and low bit rate data over a common channel. The key feature of this protocol is the utilization of user transmission to gain access to the radio resources. Once the radio resource has been acquired, it is up to the transmitter to release the reservation. PRMA is a derivative of reservation ALOHA, which is a combination of TDMA and slotted ALOHA. A reservation protocol like PRMA has an advantage in that it can utilize the discontinuous nature of speech with the help of a voice activity detector (VAD) to increase capacity of the radio channel.

To support a user data rate of 10Mb/s for multimedia services in an interference-limited wireless environment, a bandwidth of several megahertz is needed for TDMA. Employing Code Division Multiple Access (CDMA) techniques require more bandwidth than TDMA does. Since radio spectrum is expensive, efficient strategies for reusing frequencies while adequately managing cochannel interference as a means to provide enough network capacity is critically important. In contrast to narrowband cellular networks where radio spectrum is divided into multiple channel sets, which are reused only in relatively distant cells (see, e.g., [12] and references cited in [13]), broadband wireless networks must re-use bandwidth very aggressively, ideally reusing the same frequency band in every cell. The need for reuse of a common radio spectrum in all cells has also been noted by [14] and [15] for mobile broadband wireless networks. Having a shared communication channel with high data rate allows more efficient sharing of the limited bandwidth than in narrowband cellular networks.

1.3 Challenges

In the context of networks, time slots naturally become the bandwidth resources. We need to allocate time slots dynamically to various transmitters to send data packets such that a given signal-to-interference ratio (SIR) can be guaranteed at the intended receiver for successful reception. This results in the concept of dynamic resource allocation (DRA). The problem of time-slot assignment to achieve certain optimal performance while meeting an SIR requirement has been pointed out by [16] to be NP-complete or hard [17]. By using a central controller. [16] and [18] proposed approaches on assigning time slots. In the meantime, as the need for broadband multimedia communications involving digital audio and video grows, it is increasingly important for communication systems to support various traffic with QoS guarantees. In order to provide bandwidth on demand with QoS guarantees using scarce radio spectrum in a broadband fixed wireless network, the medium-access control (MAC) must:

1) efficiently reuse limited spectrum with interference avoidance;

2) handle dynamic and diverse traffic with high throughput.

In fixed wireless networks, cell sectorization and directional antennas at fixed terminal locations are key components in reducing interference from neighboring sectors and cells [23]. By exploiting this advantage, Fong et al. [7] proposed the staggered resource allocation (SRA) method as a distributed dynamic resource allocation (DRA) algorithm for the network where the same radio spectrum is shared by each sector in every cell on a dynamic time-division basis. With the use of directional antennas to suppress interference, the SRA method is particularly effective in avoiding both intercell and intracell interference.

Qiu and Chawla [19] made a novel observation: depending on terrain and fading, certain terminals (e.g., houses) due to their fixed locations, may be consistently unable to receive signals with satisfactory SIR while transmission for other terminals may always be successful. Thus, terminals at "good" and "poor" locations should be served according to different time-slot reuse patterns, which is called time slot reuse partitioning (TSRP) in [19]. The main idea is that many BS's can transmit simultaneously if the intended receiving terminals are located at good positions. On the contrary, when the receiving locations are poor, few BS's are scheduled to transmit at the same time so that a target SIR threshold can be met for successful reception at the receiving ends.

The specific implementation of TSRP in [19] divides the time frame (i.e., bandwidth) into a dedicated portion and a shared portion. At most, one packet is transmitted among four neighboring cells during each time slot in the dedicated portion and up to three packets can be transmitted simultaneously in every cell in the shared portion. The purpose is to allow terminals at "good" and "poor" locations to use time slots in the dedicated and shared portion, respectively.

Leung and Srivastava proposed Enhanced staggered resource allocation (ESRA) [20]. The new method, by considering the reception quality of terminals, has the capability of avoiding major interference as the SRA method does, and makes use of the knowledge of the reception quality at terminals to improve throughput and maintain the success probability of one (or as close to one as practically possible) for downlink transmission.

In the SRA method, as a conservative way, the traffic load of a sector has to be limited less than one-third of total channel capacity to avoid interference from major sources in the neighboring cells. In practice, not all sectors have same traffic load; specifically, some sectors own more users while others own fewer users. Can the sectors that own more users use the slots that are originally assigned to the other sectors but not utilized under the condition of interference avoidance? We propose a probing process to solve the problem without information exchange and coordination among base stations. The probing process is implemented in a FDD (frequency division duplex) medium access control (MAC) protocol. The novel protocol supports real time and non-real time traffic; meanwhile, it supports that the sectors dynamically use the slots that are assigned to the other sectors to transmit packets.

1.4 Thesis Outline

In this thesis, we implement C-PRMA (Centralized Polling Control Packet Reservation Multiple Access) in SRA method and propose a Probing Process to improve data packet transmission without information exchange and coordination among base stations. The simulation evaluates performance of the Probing Process.

In Chapter 2, we introduce SRA (Staggered Resource Allocation) as a Dynamic Resource Allocation in fixed broadband wireless access networks. In fact, there exists a flexible approach based on SRA that can perform aggressively or conservatively. We study a conservative way of SRA and find the traffic load of a sector has to be limited less than one-third of total channel capacity to avoid interference from major sources in the neighboring cells. We present CPRMA and implement it in the conservative way of SRA.

Chapter 3 presents the principle of our Probing Process. With sending the probing commands from base stations, Probing Process detects available slots unused by other sectors to provide a higher capacity in sectorization cells. The relative detail procedures including Main Procedure, Call Request Reservation Procedure and Packet Transmission Procedure are also introduced in Chapter 3.

Chapter 4 presents the simulation traffic models and results. A well-known ON-OFF voice traffic model is used in our simulation; however, the data users are modeled to generate packets according to a Poisson process. For voice traffic, we study the packet-dropping rate to find the capacity of a sector. For data traffic, we analyze the mean delay and find the breakdown point to determine the capacity of a sector. Some other characteristics are obtained from our simulations.

Chapter 5 concludes the thesis and suggests several future works.

CHAPTER 2 IMPLEMENTATION OF CPRMA IN SRA METHOD

In this chapter, we analyze the interference in fixed wireless access networks. SRA method is introduced as a way to avoid the major interference. We also present CPRMA and implement it in the conservative way of SRA.

2.1 Interference Analysis in Fixed Wireless Access Network

Let us consider a hexagonal cell layout. Each cell is divided into six sectors, each of which is served by a BS antenna with 60° beamwidth. Terminal antennas can have an angle smaller than 30°. For the hexagonal layout, Figures 2.1 and 2.2 show the interference sources for both downlink and uplink for a tagged sector under consideration (shaded in the figures). Using a simple path-loss model [11], we find that the major interference for the downlink in the tagged sector comes from other intracell sectors, the sector A and the opposite sector B. Similar observation can be made in Figure 2.2 for the uplink. Interference received from other neighboring cells is significantly attenuated because of the FTB ratio of directional antennas and distance. With this understanding, our challenge is to develop resource allocation methods that avoid most of the major interferers.

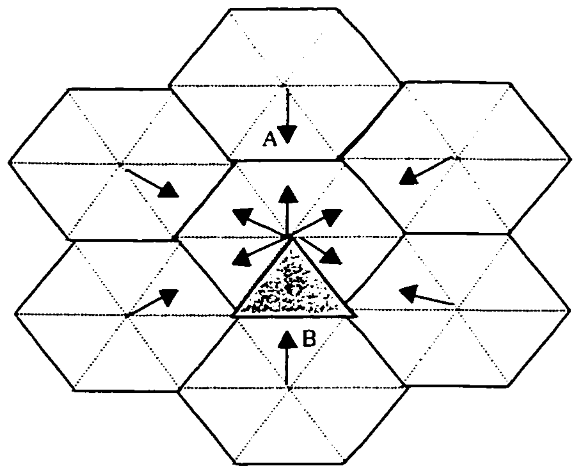


Figure 2.1 Major sources of interference for downlink

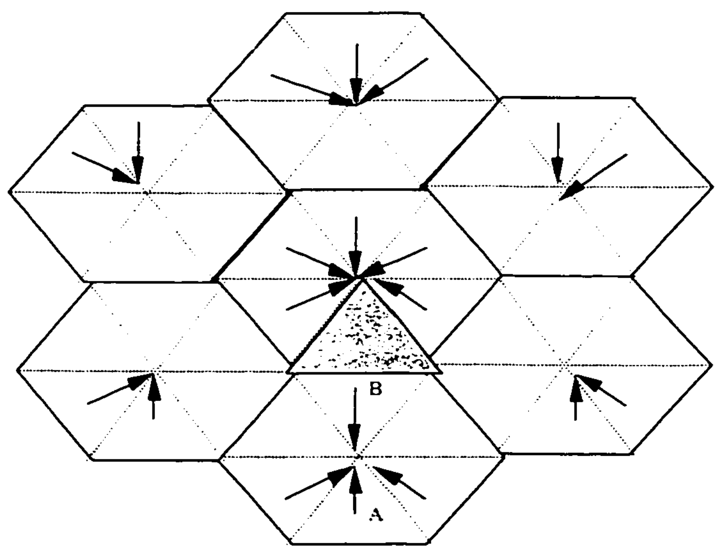


Figure 2.2 Major sources of interference for uplink.

2.2 SRA Method

Because of the colocation, sector antennas are also referred to as BS antennas. Terminals (users) use directional antennas mounted on the rooftop and pointed to their respective BS antennas. The beamwidth (angle) of each BS antenna should be just wide enough to cover the whole sector, while a terminal antenna pointing to a designated BS antenna can have a smaller beamwidth to avoid interference. Time is slotted such that a packet can be transmitted in each slot.

2.2.1 Time-Slot Assignment with Reuse Factor of Two

In this assignment method, as shown in Figure 2.3, a fixed number of time slots for a downlink (or uplink) are grouped into subframes and consecutive subframes are labeled alternately by 1 and 2. Each time frame consists of subframe 1 and the following subframe 2 (Note that the figure just shows the frame structure for a downlink or an uplink, excluding the TDD between the two). Sectors are also labeled by 1 and 2 such that no adjacent sectors share the same label. At the start of each time frame, sectors with label i can schedule packet transmission in time slots of subframe i. As a result, each sector can transmit on a 50% duty cycle, consuming at most half of the total bandwidth. Clearly, this assignment is time domain analog to a frequency reuse factor of two in existing cellular networks, although the reuse factor for our system and the cellular networks are expressed in terms of the number of sectors and cells, respectively. In the present case, collisions with interfering packets are expected to occur and, depending on their severity, may necessitate retransmission.

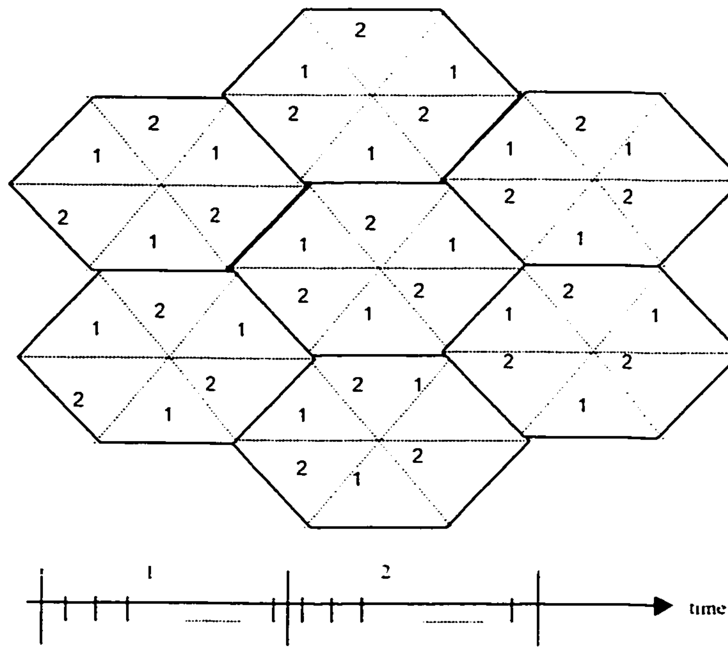


Figure 2.3 Time-slot assignment with reuse factor of two

There are two ways to improve system performance of this assignment method. First, a sector can borrow (use) time slots that its neighboring sectors do not need. Note that this approach does not increase the overall system capacity for uniform traffic load among sectors, but it does enable efficient bandwidth sharing, especially for transient surges of traffic load. However, slot borrowing requires information exchange and coordination among BS's, and that may not be desirable. A second way is to simply allow use of slots in a subframe not originally assigned to a given sector. A simple protocol can be applied to minimize concurrent transmission (thus reducing interference); specifically, label-1 sectors schedule their packet transmission in time slots, starting from the left-hand side of subframe 1 and continuing on to subframe 2 if needed, while label-2 sectors transmit in slots starting from the right-hand side of subframe 2 and continuing on to subframe1. This allocation algorithm is thus called the left-right protocol. Depending on the traffic load, this protocol yields as many as three to six concurrent packet transmissions in each time slot in each cell. In the extreme, all sectors in a cell can transmit simultaneously, yielding a reuse factor of one. Thus, this left-right protocol can be viewed as a very aggressive approach.

2.2.2 Time slot Assignment with Reuse Factor of Six

The assignment method with reuse factor of six is similar to that described above. As depicted in Figure 2.4, time slots are now grouped into six subframes (1'-6') and sectors are labeled 1-6 anticlockwise. The labeling patterns for adjacent cells differ by a 120° rotation, thus creating a cluster of three cells whose patterns can be repeated across the entire system. As described previously, sector i can schedule packet transmission in subframe I for I = 1 to 6. Without slot borrowing and overuse, this method, which has a reuse factor of six, represents a conservative approach because each sector can use only one-sixth of the total bandwidth. However, this may be appropriate for a radio environment where concurrent packet transmissions within the same cell can cause severe interference and, thus, should be prohibited.

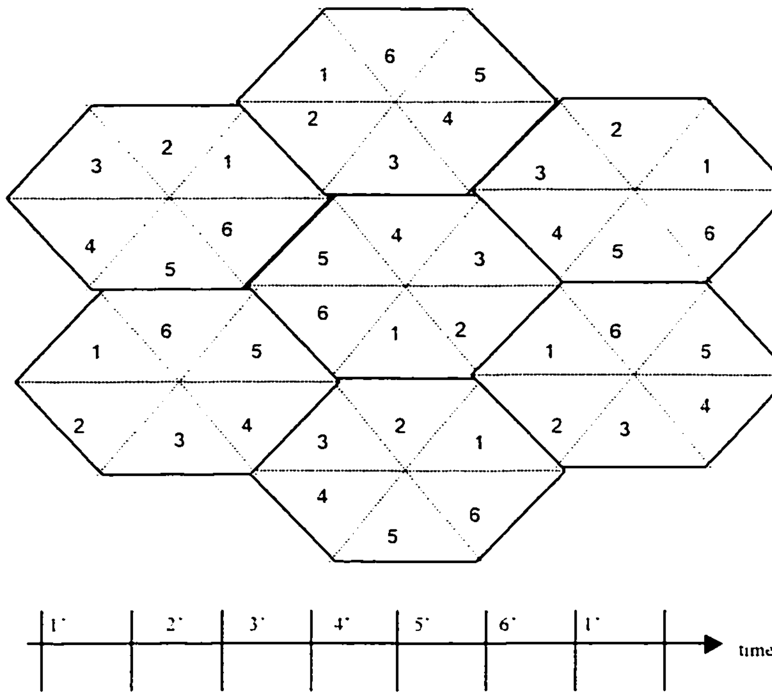


Figure 2.4 Time slot assignment with reuse factor of six

In fact, there exists a flexible approach based on the above ideas that can perform aggressively or conservatively, depending on the traffic load and a control parameter, with reuse factor ranging from one to six; it is the subject of the next section.

2.2.3 SRA

For illustration purposes, consider a regular, hexagonal cell layout in Figure 2.4. Each cell is divided into six sectors, each of which is served by a BS antenna with 60° beamwidth, and terminal antennas can have a beamwidth smaller than 60°. In the SRA method, time slots are grouped into six subframes (1'- 6') and sectors are labeled by 1 to

6 anti-clockwise, as shown in Figure 2.5. The sector labeling patterns for adjacent cells differ by a 120° rotation, thus creating a cluster of three cells whose patterns can be repeated across the entire system. Note that the time frame shown in the figure is applicable to both downlink and uplink, which are provided by the TDD or FDD technique.

Each sector assigns time slots for transmitting packets to or from its terminals according to a special order shown in Figure 2.5 (in case a terminal needs to send packets to its BS, it is assumed that BS is made aware of the need, perhaps via a separate dedicated channel or a contention channel). For example, a sector with label 1 first schedules packets for transmission in time slots of subframe 1' (denoted by a) in the figure. If it has more traffic to send, it then uses subframe 4' (**b**), subframe 5' (**c**), etc., until subframe 6' (**f**). Here, the scheduling order is denoted by "a", "b", "c", "d", "e" and "f". The reason for such an order is that if interference due to concurrent packet transmission in the same cell can be tolerated, then after using all slots in the first subframe **a**, a sector should use the first subframe of the opposite sector in the same cell, in order to make the best use of BS directional antennas. Following that, time slots in the first subframes for the sectors next to the opposite sector are used. To avoid interference due to overlapping antenna patterns of neighboring sectors, their first subframes are used as the last resort. For simplicity (while causing very minor throughput degradation), Figure 2.5 does not show the assignment from the left and right-hand side of the subframes as suggested in [7]. As depicted in the figure, the assignment order for the next sector is "staggered" by a right rotation by one subframe based on the order of the previous sector. The assignment order, regardless of the associated sector, is generally referred to as the staggered order in the following.

Sector	1'	2'	3'	4'	5'	6'
1	a	e	d	b	c	f
2	f	a	e	d	b	c
3	c	f	a	e	d	b
4	b	c	f	a	e	d
5	d	b	c	f	a	e
6	e	d	b	c	f	a

time →

Figure 2.5 Order of slot assignment for the SRA method.

It is easy to see from figure 2.5 that if all sectors have traffic loads of less than one-sixth of total channel capacity, all packets are transmitted in different time subframes, thus causing no interference within the same cell. Of course, as the traffic load increases, packets are transmitted simultaneously, thus increasing the level of interference. Nevertheless, the staggered order exploits the characteristics of directional antennas to allow multiple concurrent packet transmissions while reducing the intracell interference.

Besides managing intracell interference, the SRA method also helps avoid interference from major sources in the neighboring cells. This is particularly so when traffic load is low to moderate. To see this, let us consider the downlink for Sector 1 in the middle cell of Figure 2.4. Sector 2 in the bottom cell and Sector 3 in the upper cell are the major sources of interference. By examining the staggered order for Sector 1, 2 and 3, we find that they will not transmit simultaneously, so they will not interfere with each other provided that all of them have a traffic load of less than one-third of total channel capacity (i.e., using only subframes **a** and **b** for transmission). The same comment also applies to the uplink where Sectors 2 and 5 of the bottom cell in the figure now become the major sources of interference. Due to the symmetry of the staggered order and cell layout, this same applies to each sector in every cell. The character of SRA is our basis to design a new MAC protocol.

2.3 C-PRMA protocol

C-PRMA [21] is a packet switching multiple access protocol especially devised for a micro cellular environment. Within each cell, two separate time slotted channels are used for communication. The uplink channel conveys the information from the users to the BS, while the downlink channel is used to communicate in the opposite direction. The downlink channel is exclusively used by the BS, so that no multiple access problems exist. On the contrary, the transmissions of different users in the uplink channel must be coordinated. For this purpose, the BS transmits slot by slot commands, which allow managing a random access polling scheme among the users.

The polling commands generated by a scheduling algorithm specify whether a slot of the uplink channel is available or reserved and identify the user enabled to transmit. The random access is used for reservation transmission in available slots, while the polling mechanism provides the transmission coordination, in the reserved slots, among the active users. This is a proper approach in the microcellular environment because the small propagation delay allows a very efficient command/response communication mode, and the BS, being the interface between the radio channel and fixed network, already performs other centralized functions.

The major goal of the scheduling algorithm is to achieve efficient traffic integration (through a dynamic slot allocation) for services with different bandwidth and delay constraints. Furthermore, the flexibility in the slot assignment allows recovering corrupted packets by using retransmission techniques.

In order to efficiently operate the scheduling algorithm, each of the users must signal when it enters the active state, i.e., when it is ready to transmit packets. This signaling procedure is performed by sending a reservation request in an available slot on the uplink channel. Note that all available slots can be used for reservation request transmissions. The user remains in the contending state until its reservation request is successfully received by the BS. When this event occurs, the user starts transmitting data packets in the reserved slots, assigned by the BS, and keeps on transmitting as long as it remains active.

2.3.1 Description of CPRMA

The core functions of CPRMA are performed by the scheduling algorithm that dynamically assigns reserved slots in the uplink channel to active users. The information needed at the BS to enable the operation of the scheduling algorithm is provided by the users. The parameters that specify the service class (i.e., voice, data, packet rate, etc.) and service quality (i.e., priority level, maximum tolerable packet delay, etc.) are negotiated between users and the BS at the call-setup phase. Each reservation, in addition to a request to transmit, also contains the indication of the delay already suffered by the first packet of burst. The latter information is used by the scheduling algorithm.

Furthermore, in addition to the reservation information transmitted on the uplink channel, the CPRMA implementation requires the transmission of "commands" at each slot from the BS to users. To accommodate the transmission of this additional information, the slots on the downlink channel are taken long enough to contain a data packet and a command. The same slot length is assumed in the uplink channel. In fact, in such a slot, an acknowledgement for the data packet correctly received by the user on the downlink channel must be transmitted in addition to a data packet or reservation information. The immediate acknowledgement is needed at the BS if a quick recovery of errored packets is necessary to meet a given quality of service for real time traffic. Notice that the acknowledgement for an uplink data packet transmission is not needed in the downlink channel, as the BS can account for not correctly received packets by repeating the polling command to request the transmission of the failed packet.

Even if CPRMA can operate with independent time slots on the two channels, for sake of simplicity, we assume that both uplink and downlink channel slots are synchronous. In the sequel, we discuss the operation of the signaling and data channels in both the uplink and downlink cases.

2.3.1.1 Signaling Channel

In the C-PRMA protocol, available slots in the uplink channel are dedicated to the transmission of the reservation requests. As already mentioned, a reservation request must be transmitted. The reservation request can be structured as a minipacket, as it carries a limited amount of information (user identifier, the time spent by the first packet in the transmission buffer, and a CRC field). Consequently, the available slots can be split into m > 1 minislots, each one of a length equal to the minipacket transmission time.

Many multiple-access protocols existing in the literature can be used by the users in the contending state to transmit in the minislots. In [21], both the S-ALOHA algorithm, with a random retransmission of collided mnipackets, and a random-access splitting algorithm [21] have been considered. Both algorithms can be easily implemented because all the needed information, such as the slot-feedback information and the start of a contention cycle, is broadcasted by the BS.

After each available slot in the uplink channel, a slot in the downlink channel is dedicated to acknowledge the reservation requests. In such a slot, the BS transmits the identifier of all the new users that have successfully reserved. Figure 2.6 shows an example of a sequence of reservation minipackets and acknowledgement for the case of m = 3. The command CO identifies the uplink slot as available. More than three users attempt to reserve; a collision occurs in the third minislot. Only successful stations i and j are acknowledged in the next downlink channel slot.

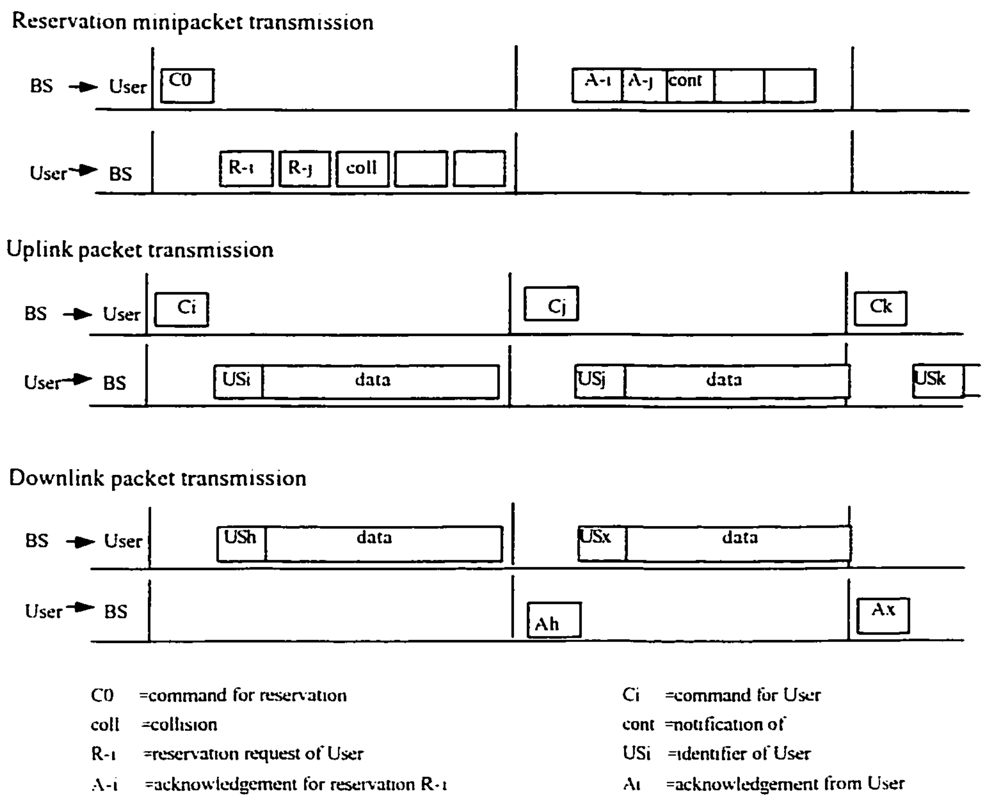


Figure 2.6 Data and signaling channels

2.3.1.2 Data Channel

In the uplink channel, data packets are transmitted on the reserved slots that are assigned by the BS using the polling commands. The polled user responds with a data packet transmission in the same time slot, as illustrated in Figure 2.6. This implies that the transmission delay and reaction time are negligible, as expected in the microcellular systems. When this is not the case, the data transmission is shifted by a fixed delay (number of slots) to account for the overall delay.

In addition to the user identifier, the polling commands contain a CRC field for error detection and a sequence command (SC) bit. The SC bit for the same user is alternatively set to zero and one. This procedure allows a correct exchange of commands and data packets between the BS and each user. In fact, if a command fails or the data packet is not correctly received by the BS in the assigned slot, the BS repeats the command with the same SC value to enable the user to recover from the error.

The data packet transmitted by the polled user has a header, which contains the user identifier, a CRC field, and a more indicator (MI) bit to signal the BS needs to keep on (MI = 1) or to stop (MI = 0) polling the user.

The transmission of data packets from the BS to users, illustrated in the example of Figure 12. can be done in any downlink channel slot after the command transmission. No signaling is needed for the slot assignment as all transmissions are performed by the BS. Note that the command and data packet in the same slot can be addressed to different users. The header of the data packet contains the destination user identifier and a CRC field. At the reception of a correct data packet, the user sends an acknowledgment in the next slot of the uplink channel. An example of the complete procedure on the uplink and downlink can be obtained by merging the different cases reported in Figure 2.6.

The correct operation on both channels requires time guards between slots to account for different propagation delays. In the uplink channel, a further time guard is needed within the slot between the acknowledgment and data packet, as they can be originated by different users. Note that the time-guard lengths involved in cellular systems are equivalent to a few bits of transmission time. Detailed slot structures used by C-PRMA are given in Figure 2.7.

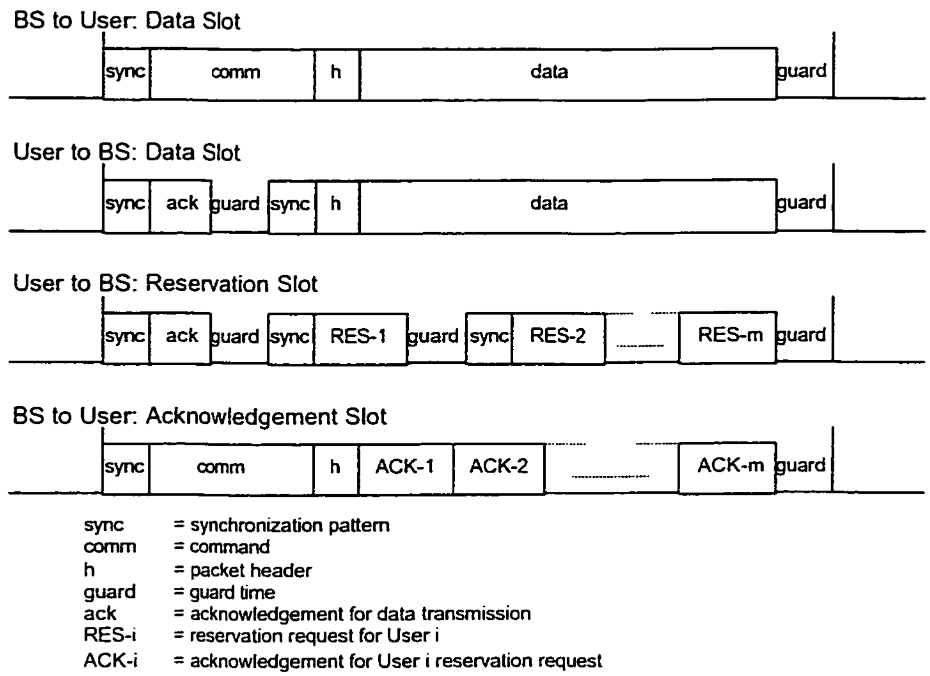


Figure 2.7 Slot structures

2.3.2 CPRMA Scheduling Algorithm

The scheduling algorithm (SA) is the core of the CPRMA protocol as it manages the radio resources in order to optimize the system performance. In particular, it allows:

1) Integrating traffic sources with different transmission rates, priorities, delays, and packet-loss requirements;
2) Managing the retransmission of corrupted packets to meet the quality of service, even in the presence of a noisy channel;
3) Optimizing the uplink channel utilization.

It operates on the basis of service parameters acquired by the BS for each user i either during the setup (type A parameters) or in the reservation (type B parameters). The type A parameters are:

1) L_i: maximum tolerable packet delay, in slots;

2) T_i: packet inter-arrival time when the source is transmitting, in slots;

3) P_i: packet discard priority.

The type B parameter w_i, updated at each silent-to-active transition, is defined as the time, in slots, spent by the first packet of the burst in the transmission buffer before its successful reservation.

Based on the above service parameters, the SA generates the following internal parameters.

1) C_i: lifetime, in slots, of the hand-of-the-line (HOL) packet in the transmission buffer of User i. The parameter C_i is set to the value L_i-W_i when a new reservation is accepted, and it decrements at each time slot. The parameter C_i of a packet in a burst other than the first one is equal to the C_i of the preceding packet plus T_i.

2) R_i: number of retransmissions of the HOL packet at User i.

3) F_c: contention flag. It is set to one when a contention resolution phase among users is in progress. Otherwise it is zero.

4) CRS: consecutive reserved slots issued since the last available slot.

The detailed use of these parameters will be given in the following subsections. In addition, the SA uses a shift register, called polling register (PR), of size L+1, where L is equal to the maximum L_i. Each position of PR is either empty or filled with the identifier

R of a user waiting to be polled, after its reservation has been successfully received. The positions in PR are numbered from right to left starting from zero. A new reservation is stored in PR in the highest available position smaller or equal to the C_i of the corresponding packet. Therefore, the user i identifier in position k of PR specifies that a packet at User i must be successfully transmitted within k slots, otherwise, it will be dropped.

As User i generates packets every T_i slots, the SA increments C_i by T_i when either a packet is correctly received by the BS or the user i identifier has reached the position zero in PR. If $C_i > L_i$, the next packet to be reserved has not yet been generated. Its reservation will be inserted in PR as soon as $C_i = L_i$. At each time slot, the scheduler decides whether to issue an available or reserved slot. If the position zero of PR contains an user identifier, a reserved slot is always issued.

2.3.2.1 Main Procedure

The main procedure of the scheduling algorithm specifies all the operations performed by the SA at each time slot (see Figure A.1 in Appendix A). According to the information received in the previous slot, the main procedure updates the internal parameters and the content of PR by using the reservation scheduling (RS) procedure. Finally, the Main Procedure ends by calling the command issuing (CI) procedure, dedicated to the slot assignment.

2.3.2.2 RS Procedure

This procedure, specified in Figure A.2 of Appendix A, is in charge of storing, in an available position of PR, the identifier of user i, which has a packet ready for transmission. An available position for User I is any position j in PR, with j <= C_i, that either is empty or contains a user identifier with a priority lower than Pi. The RS procedure is structured in such a way that each reservation occupies the highest indexed available position. In fact, this is the policy that minimizes the constraints on the succeeding reservations.

2.3.2.3 CI Procedure

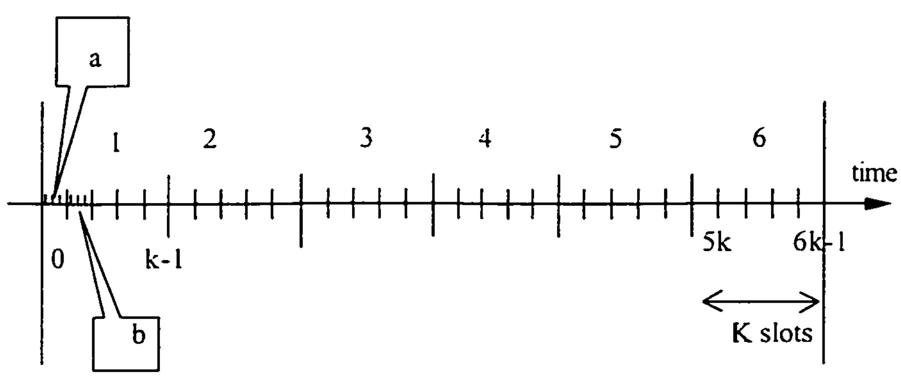
This procedure, specified in Figure A.3 of Appendix A, is in charge of managing the optimum slot assignment to reservation and data channels. Its operation is based on the status of the data channel, described by the PR content and the Ri parameter associated to each user in PR, and on the status of the reservation channel, characterized by the flag Fc and counter CRS.

To avoid a packet being lost by a user in the position zero of the PR, the procedure assigns a reserved slot to that station, with no regard to its service priority Ri. Otherwise, either an available or reserved slot is issued. A minimum rate of available slots, 1/Ct, is guaranteed by a threshold on the number of consecutive reserved slots, CRS. Moreover, an available slot is issued when the contention flag F_c, set by the reservation protocol, is on.

The reserved slots are issued according to the parameters (number of retransmissions of a packet), which acts as a dynamic service priority. A reserved slot is assigned to the user with the lowest r_i. If more than one, the user in the smallest position of the PR is selected. The combined use of position and service priority has been adopted to provide reserved slots for transmission of packets with a shorter lifetime, and to avoid that, a user with unsuccessful transmissions is granted too many slots, with a consequent waste of bandwidth.

2.4 C-PRMA in SRA method

This section describes the proposed protocol in which C-PRMA is implemented in SRA method. In the novel MAC protocol, to avoid interference from major sources in the neighboring cells, a frame is also divided into six sub-frames as shown in figure 2.11. However, subframes 1' and 4' (same with subframe **a** and **b** in figure 2.5.) are reserved for Sector 1 of central cell in Figure 2.4, subframes 2', 5' and subframes 3', 6' are reserved for Sector 2 and Sector 3 respectively.



a: for uplink, reservation request minislots
b: for downlink, request acknowledgement minislots
Figure 2.8 Frame Division

2.4.1 Transmission Request Access

In our protocol, we also use the S-ALOHA algorithm for the reservation of requests with a random retransmission of collided minipackets as used in CPRMA [22]. However, a simplified CPRMA is considered to study the effect of our probing process. The first slot of subframes 1', 2', 3' in the uplink channel is dedicated to the transmission of the reservation requests. The reservation request can be structured as a minipacket, as it carries a limited amount of information. Consequently, the available slots can be split into m>1 minislots, each one of a length equal to the minipacket transmission time. When BS issues a transmission-request command, each user with pending requests decides whether to make a request with probability $\mathbf{p_r}$ or not (i.e., in a pr-persistent manner). If it has decided to send the request, it does so in one of the request mini-slots. After each first slot of Subframe1', 2', 3' in the uplink channel, a slot in the downlink channel is dedicated to acknowledge the reservation requests. In the acknowledgement mini-slots, BS transmits the identifier of all the new user's that have successfully reserved.

The request includes the following service parameters [22]:

1) Li: maximum tolerable packet delay, in slots;
2) Ti: packet interarrival time when the source is transmitting, in slots;
3) Pi: voice or data packet;
4) Wi: waiting time of the first packet of the burst in the transmission buffer before its successful reservation.

Based on the above service parameters,

5) Ci: defined as life time, in slots, of the head-of-the-line (HOL) packet in the transmission buffer of user i.

The parameter Ci is equal to Li-Wi when a new reservation is accepted, and it is decreased at each time slot.

In the thesis, we do not discuss the packet discard priority as described in [22], but study only the data and voice traffic. BS schedules the transmission of downlink data packets or the polling of uplink data packet transmission permission with lower priority than the scheduled voice traffic.

2.4.2 Data Transmission

In the downlink channel, BS schedules data packet transmission to users. The data packet transmitted has a header, which contains the user ID. If the destined user receives the packet correctly, it responds with ACK signal in the next slot. If the transmission fails, the User responds with N-ACK signal in the next slot.

In the uplink channel, data packets are transmitted on the reserved slots that are assigned by BS using the polling commands. The polled user responds with a data packet transmission in the same time slot. The data packet transmitted by the polled user has a header, which contains the user identifier.

In addition to the User identifier, the polling commands contain a sequence number (SN) bit. The SN bit for the same user is alternatively set to zero and one. This procedure allows a correct exchange of commands and data packets between BS and each user. In fact, if a command fails or the data packet is not correctly received by BS in the assigned slot. BS repeats the command with the same SN value to enable the user to recover from the error.

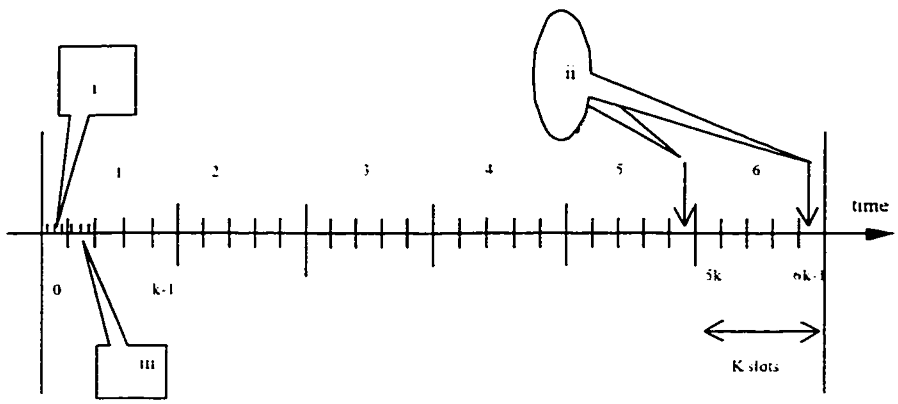
CHAPTER 3 PROBING PROCESS

3.1 Objective and Motivation

In the SRA method, as a conservative way, the traffic load of a sector has to be limited less than one-third of total channel capacity to avoid interference from major sources in the neighboring cells. In practice, not all sectors have same traffic load; specifically, some sectors own more users while others own fewer users. Can the sectors that own more users use the slots that are originally assigned to the other sectors but not utilized under the condition of interference avoidance? This problem leads to the Probing Process we propose here.

3.2 Description of Probing Process

In the scheduling process depicted in figure 3.1, it is assumed that subframes 1' and 4' are reserved for sector 1 of central cell in figure 2.4. Sector 1 schedules its packet transmission in time slots of subframes 1' and 4' beginning from slots S_0 to S_{k-1} and slots S_{3k} to S_{4k-1} where slot S_0 is used as reservation request minislots and slot S_1 is used as request acknowledgement minislots. Sector 1 can also schedule its packet transmission in time slots of subframes 5' and 2', or in time slots of subframes 6' and 3'. Scheduling is performed in the reverse order starting from S_{6k-1} to S_{5k} in case of subframe 6' or from S_{5k-1} to S_{4k} in case of subframe 5'. Similar order is followed for subframe 2' or 3'. Sector 1 should first execute the probing process to detect available slots before it schedules its packet transmission in subframes 5' and 2' or 6' and 3'.



i: for uplink, reservation request minislots
ii: probing packet scheduled in reverse order beginning here
iii: for downlink, request acknowledgement minislot

Figure 3.1 Probing Process

The probing process starts from the slots 5k-1 and 6k-1. Sector 1 sends a polling command to ask a user node to respond with a probing packet in the slot 5k-1 and 6k-1. If the transmission succeeds, the sector assumes the slot can be used. The process continues in the reverse order to determine more available slots. If a certain transmission fails, the sector gives up the slot and tries again after random frames.

3.3 Procedures

Figures 3.2-3.7 show the flow charts for the procedures used to evaluate the performance of the proposed MAC protocol. The flow charts for BS of sector 1 are presented in figures 3.2-3.6. BS after receiving the clock signal judges the kind of the present slot and calls the respective procedure. Figure 3.7 shows the procedure for the user stations. If a user receives the reservation command and there are packets to transmit, it will send call request to BS. If the reservation is successful, it will transmit packets in certain slots according to the command from BS. If not, it will try to send request again in the next frame.

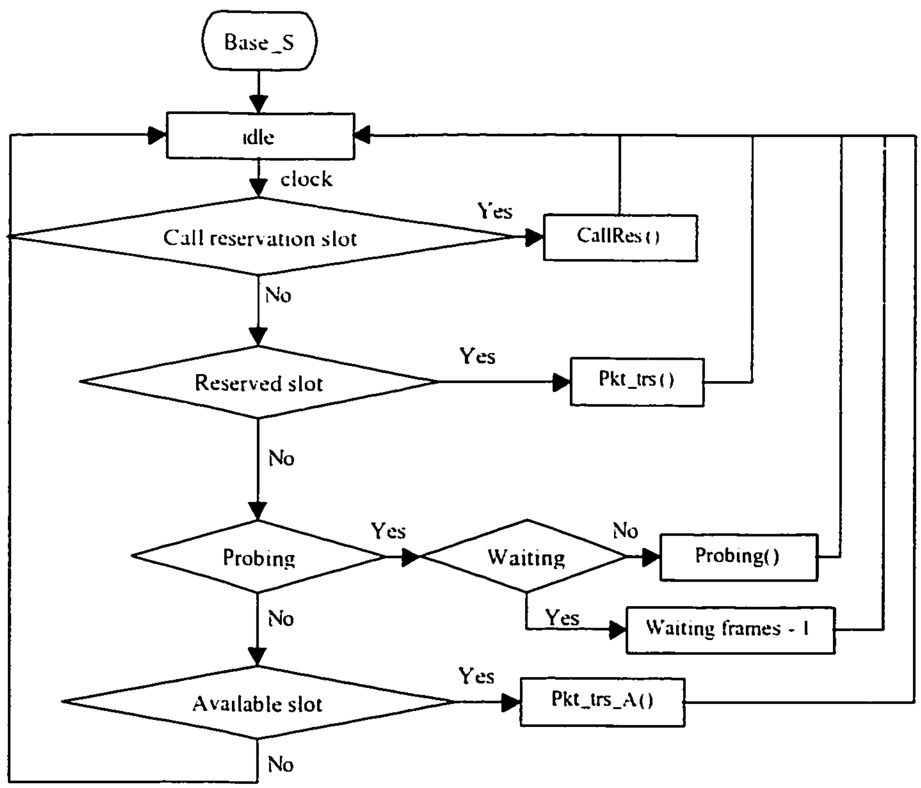


Figure 3.2 Main procedure for Base Station of sector 1

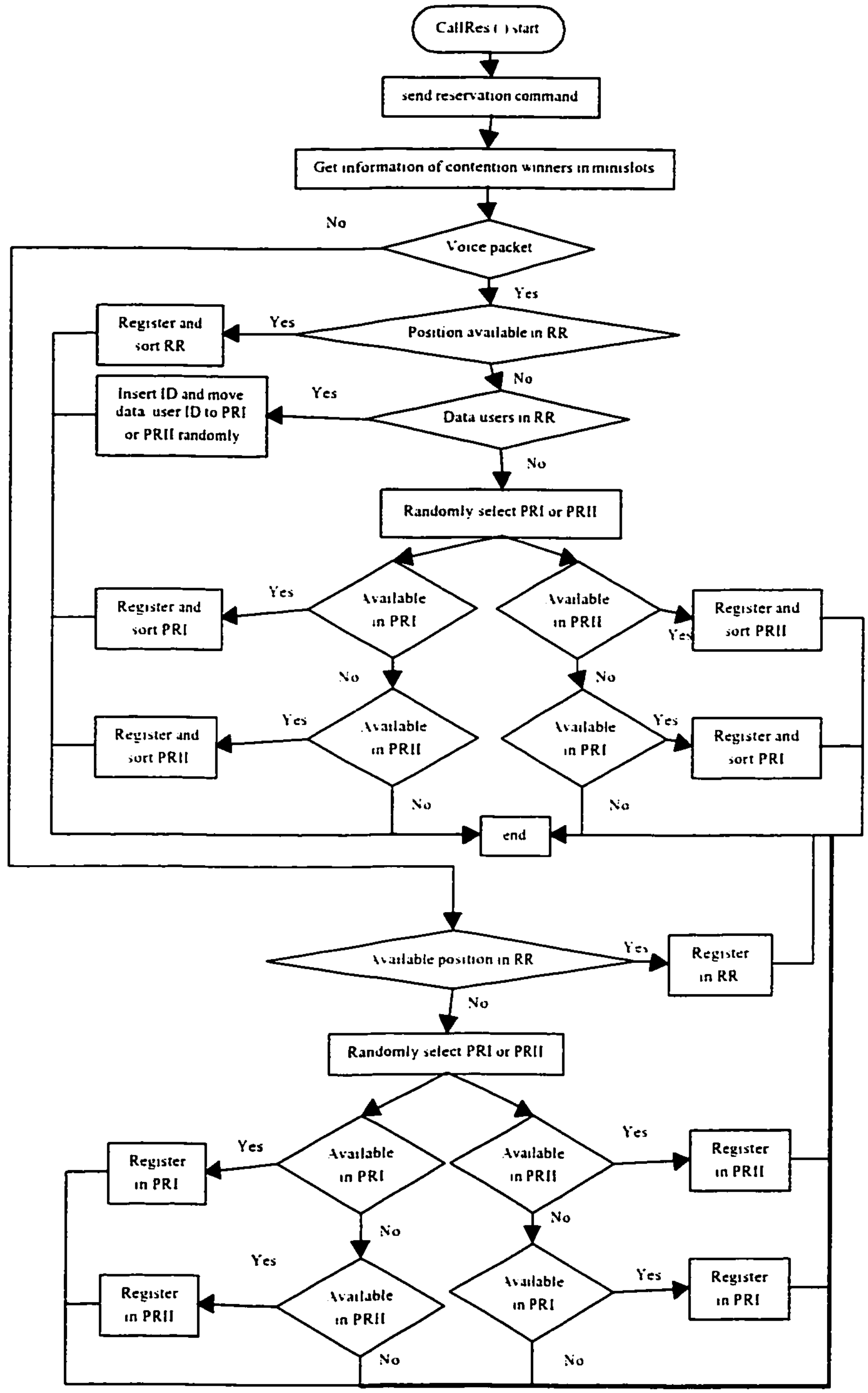


Figure 3.3 Call Reservation procedure for Base Station of sector 1

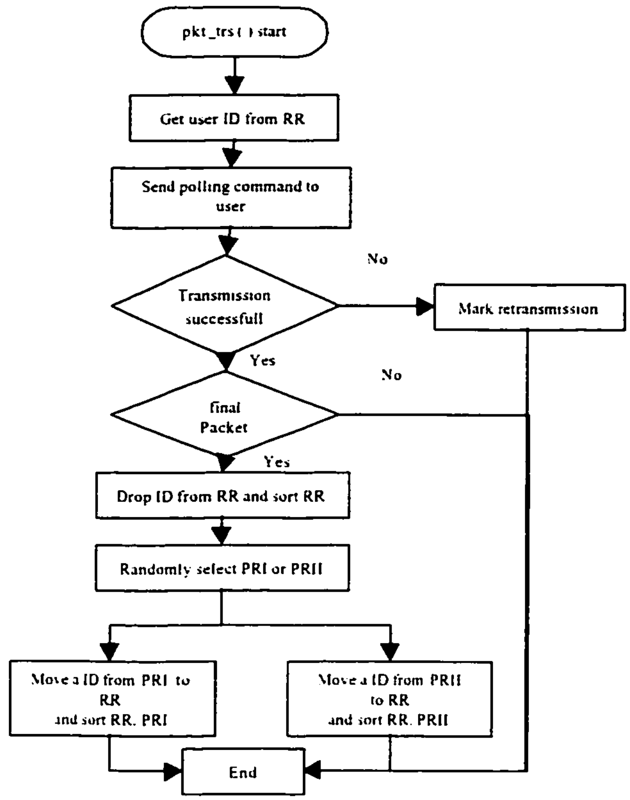


Figure 3.4 Packet Transmission procedure for sector 1

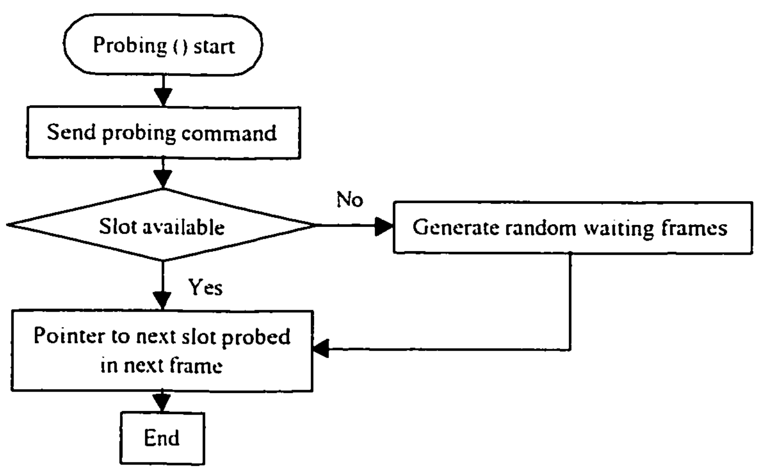


Figure 3.5 Probing procedure for Base Station of sector 1

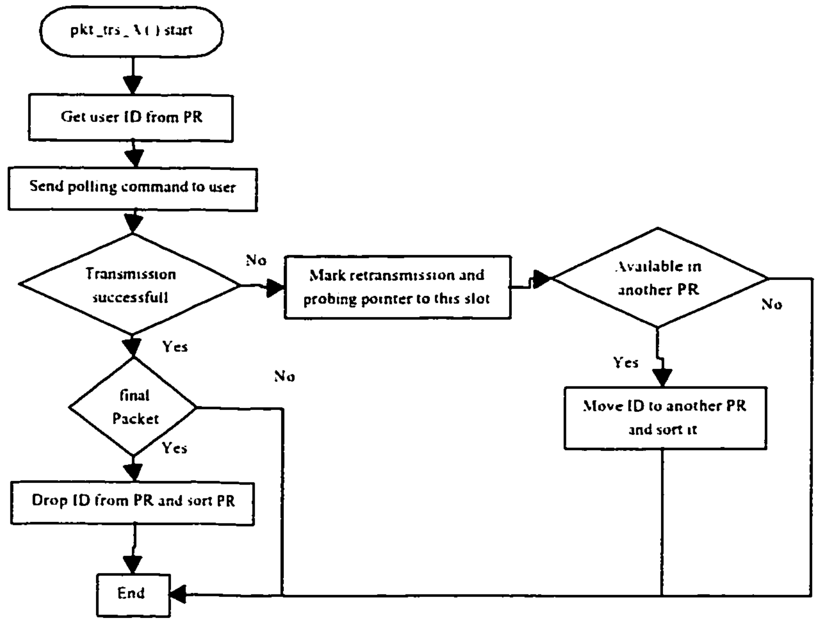


Figure 3.6 Sector 1 procedure for Packet Transmission in sector 2 or 3

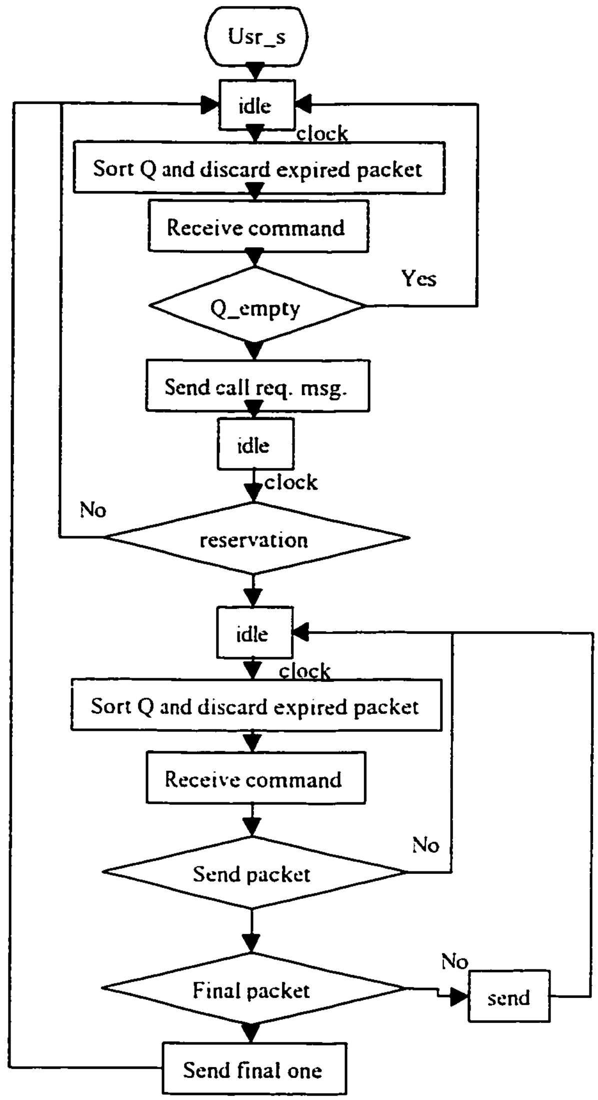


Figure 3.7 procedure for User Station of sector 1

In the following, the main procedures defined in the flow charts are given in detail with their C-like code.

3.4 Probing Procedure

The proposed probing procedure as described in figure 3.8 begins by the sector polling the user nodes to determine unassigned or available slots. NumProbingI and NumProbingII define the total number of available slots in subframes 6' and 3', and subframes 5' and 2' respectively for sector 1. The polling is done in the reverse order starting from the slot 6k-1 and slot 5k-1 for subframes 6' and 5' respectively. If user node responds with a probing packet, the sector assumes the slot can be used and it increases the corresponding number of slots by one. The process continues to determine more available slots. If slot is not available, the sector gives up the slot and tries again after random frames.

The probing process is forbidden to proceed in either of the Slot k, Slot k+1, Slot 2k or Slot 2k+1 because Slot k and Slot 2k are divided into the reservation request mini-slots for other sectors in uplink, and Slot k+1 and Slot 2k+1 are used for the reservation acknowledgement mini-slots by other sectors in downlink. Therefore, the maximum of NumProbingI and NumProbingII is 2k-2.

An essential parameter to be confirmed in the proposed simulation is the waiting time, in frames, of probing command retransmission after unsuccessful probing procedure. This parameter reflects how often the probing process starts. Too intensive probing packet transmissions with lower waiting times cause serious effect to the communications in sector 2 and sector 3 as shown in the simulation results in Chapter 4. To diminish the interference to the communications of sector 2 and sector 3 as much as possible, it is important to find the proper value for the waiting frames of probing command retransmission.

```
void Probing(){

 /* initializing */
 int NumProbingI=0;
 int PntProbingI=5k-1;         /* initially point to the probing slot in subframe 5'*/
 int NumProbingII=0;
 int PntProbingII=6k-1;        /* initially point to the probing slot in subframe 6'*/

/* Generate the command to ask for the probing packet response from a certain user in
subframes 2' and 5'*/
 Probing_I:
 /* Execute probing process in subframes 2' and 5' */
 if( k+1<PntProbingI<2k || 4k-1<PntProbingI<5k ){
   ProbingCommand(i);                 /* send the probing command to User i */
   if(slot available){
      NumProbingI++;
      PntProbingI--;                  /* probing slot for the next frame * /
      /*point to the probing slot in subframe 2' */
      if(PntProbingI == 4k-1) PntProbingI = 2k-1;
      goto Probing_I;            /*execute probing process again in the next frame */
```

```
    }
    else {
        j=rand();
        Wait(j);                    /* wait for random frames j if slot not available*/
    }
}
```

/* Generate the command to ask for the probing packet response from a certain user in subframes 3' and 6'*/

```
Probing_II:
/* Execute probing process in subframes 3' and 6' */
if( 2k+1<PntProbingII<3k || 5k-1<PntProbingII<6k ){
    ProbingCommand(i);              /* send the probing command to User i */
    if(slot available){
        NumProbingII++;
        PntProbingII--;             /* probing slot for the next frame * /
        /* point to the probing slot in subframe 2' */
        if(PntProbingII == 5k-1) PntProbingII = 3k-1;
        goto Probing_II;            /*execute probing process again in the next frame */
    }
    else {
        j=rand();
        Wait(j);                    /* wait for random frames j if slot not available*/
```

```
        }
    }
}
```

Figure 3.8 C-like code of Probing Procedure

3.5 Call Request Reservation Procedure

Figure 3.9 describes the procedure for Call Request Reservation. Three registers named Reserved Register (RR) of size 2k-1, Probing Register I (PRI) of size 2k-2, and Probing Register II (PRII) of size 2k-2 are defined. The positions of RR correspond to the slots S_1-S_{k-1} and S_{3k}. S_{4k-1} (S_0 reserved for the reservation request mini-slots). The positions of PRI correspond to the slots S_{k+2}-S_{2k-1} and S_{4k}-S_{5k-1} (S_k and S_{k+1} reserved for the reservation request mini-slots and the reservation acknowledgement mini-slots by other sectors). The positions of PRII correspond to the slots S_{2k+2}-S_{3k-1} and S_{5k}-S_{6k-1} (S_{2k} and S_{2k+1} reserved for the reservation request mini-slots and the reservation acknowledgement mini-slots by other sectors). Each position of RR, PRI and PRII is either empty or filled with User ID. NumRev, NumPI and NumPII express the number of users registered in RR, PRI and PRII.

When BS receives a real time reservation request from a voice user, it first checks if there is a position in RR, PRI and PRII to accommodate the user. If NumRev<2k-1, it marks the User ID in the empty position of RR and NumRev is increased by one. If NumRev>=2k-1, it checks if there are non-real time users in RR. If it is so and (NumPI<NumProbingI or NumPII<NumProbingII), it moves the leftmost non-real time

user in RR to PRI or PRII and marks the User ID in the position. If there is no non-real time user in RR and NumPI<NumProbingIor NumPII<NumProbingII, the sector also tries to mark the User ID in the leftmost position of PRI or PRII and move the non-real time users to higher positions. Generally, the sector randomly selects PRI or PRII to schedule users. The users may also be scheduled first in PRI and then in PRII. If NumPI>=NumProbingI and NumPII>=NumProbingII, all available slots are full and the request will not be registered and acknowledged.

When BS receives a reservation request from a non-real time user and there is an empty position (NumRev<2k-1) in RR, it marks the new User ID in the empty position of RR. If RR is full, it randomly selects PRI or PRII to find if there is an empty position in them, in which case it marks the User ID. BS repeats the process until an empty position is not found, in which case the request is refused. BS can also first select PRI to schedule packet transmission and then select PRII.

```
void CallRes() {

    /* schedule all contention winners in minislots */
    for (j=winners) {                       /* for all contention winners */
        c[j]=l[j]-w[j];             /* reset c[j] */



    /* If it is the voice traffic */
        if(Voice_packet&&c[j]>0){
```

```
if(NumRev<2k-1){     /* there is a position in RR */
        rr[NumRev]=j;          /* register the user j in RR */
        NumRev++;
        sort_array(rr); /* sort RR according to c[ ] in the ascending
                          order */
}
else {                              /* no position in RR */
        search_array(rr, data_packet);/*search data packets in RR*/
        if(Data_in_array){     /*there is a data user x in RR*/
                insert_array(rr,j);/* register j in RR replacing x */
                sort_array(rr);          /* sort RR */
                /* randomly select PRI or PRII */
                h=rand_sel_array(pri,prii);
                insert_array(h,x);/* insert x in PRI or PRII */
        }
        else {                      /* total voice users in RR */
                h=rand()%2;  /* randomly select PRI or PRII */
                if(h==0 and
                     /* select PRI and it's not full */
                     NumPI<NumProbingI){
                        pri[NumPI]=j; /* register j*/
                        NumPI++;
                        sort_array(pri);/* sort PRI[ ] */
```

```
                }
                /*if PRI is full */
                else if(h==0 and NumPII<NumProbingII){
                        prii[NumPII]=j; /*register j in PRII[ ]*/
                        NumPII++;
                        sort_array(prii); /* sort PRII[ ] */
                }
                /* select PRII and it's not full */
                if(h==1 and NumPII<NumProbingII){
                        prii[NumPII]=j; /*register j in PRII[ ]*/
                        NumPII++;
                        sort_array(prii);
                }
                /* if PRII is full */
                else if(h==1 and NumPI<NumProbingI){
                        pri[NumPI]=j;
                        NumPI++;
                        sort_array(pri);
                }
        }
    }
}
```

```
/* if it is the data traffic */
    if(Data_packet){
        if(NumRev<2k-1){                    /* RR[ ] is not full */
            rr[NumRev]=j;                   /* register j in RR[ ] */
        }
        else {                              /* RR[ ] is full */
            h=rand()%2;             /* randomly select PRI or PRII */
            /*select PRI and it's not full */
            if(h==0 and NumPI<NumProbingI){
                pri[NumPI]=j;           /* register j in PRI */
                NumPI++;
            }
            /* if PRI is full */
            else if(h==0 and NumPII<NumProbingII){
                prii[NumPII]=j;                 /* register j in PRII */
                NumPII++;
            }
            /* select PRII and it's not full */
            if(h==1 and NumPII<NumProbingII){
                prii[NumPII]=j;
                NumPII++;
            }
            else if(h==1 and NumPI<NumProbingI){ /*if PRII is full */
```

```
                    pri[NumPI]=j;
                    NumPI++;
                }
            }
        }
    }
}
```

Figure 3.9 C-like code of Call Request Reservation Procedure

3.6 Packet Transmission Procedure

Figures 3.10 and 3.11 describe the Packet Transmission Procedure in reserved and available slots respectively. BS decides the scheduling for the slots in the beginning of a frame. In its reserved slots, it sends polling commands according to the RR's User IDs in the order from low position to high position. In the unreserved slots, BS sends commands according to the User IDs of PRI or PRII in the reverse order from high position to low position. A polling command is not send by BS if the position of RR, PRI or PRII is empty.

In the process of message transmission, any interference caused by other sectors that originally reserved subframes 2', 3', 5' and 6' will result in sector 1 or its users stop the packet transmission.

When a user finishes the transmission, it will send the final packet with stop information in its header to BS. After receiving the packet, BS removes the User ID from RR or PR.

```
void pkt_trs() {
        int CommandPointer=0;          /* initalizing */
        /* in Subframe 1' and 4' */
        i=rr(CommandPointer);          /* get user ID from RR */
        if (i!=0){
                send_polling_command(i);    /* send polling command to user i*/
                CommandPointer++;           /* point to next user */
        }
        if(transmission_successful){
                c[i]=c[i]+t[i];
                schedul(i);                          /* schedule next packet for user i */
                if(Final_Packet) {                   /* user i reports final packet */
                        ShiftArray(rr, i);      /* drop user i by shifting RR from RR[i] */
                        if (NumPI!=0 and NumPII!=0) { /* PRI and PRII is not empty */
                                z=rand_select(PRI,PRII);/*randomly select PRI and PRII*/
                                rr[2k-2]=z[0];          /* move used ID from z to rr */
                                ShiftArray(z,0);        /* shift z */
                        }
                }
```

```
        }
        else{
                c[i]=c[i]+Long_Frame; /* re-transmission in next frame */
                MarkRetr[i]=1; /* mark re-transmission */
        }
}
```

Figure 3.10 C-like code of Packet Transmission Procedure in reserved slots

```
void pkt_trs_A() {
        /* in subframe 3' and 6' */
        int CommandPointerI=NumPI;      /* get user ID from high positions of PR */
        i=pri(CommandPointer);          /* get user ID from PRI */
        send_polling_command(i);        /* send command user i*/
        CommandPointerI--;              /* point to next user in reverse order */
        if(transmission_successful){
                c[i]=c[i]+t[i];
                schedul(i);             /* schedule next packet for user i */
                if(Final_Packet) {      /* user i reports final packet */
                        ShiftArray(pri, i);     /* drop user i by shifting PRI from PRI[i] */
                }
        }
        else{
        MarkRetr[i]=1;                                  /* mark re-transmission */
```

```
        if(NumPII<NumProbingII){                    /*PRII is not full*/
        prii[NumPII]=i;                              /*move user ID to PRII */
        pri[NumPI]=0;
        }
}
        /* in subframe 2' and 5' */
        int CommandPointerII=NumPII;
        i=prii(CommandPointer);
        send_polling_command(i);
        CommandPointerII--;
        if(transmission_successful){
                c[i]=c[i]+t[i];
                schedul(i);
                if(Final_Packet) {
                ShiftArray(prii, i);
        }
}
        else{
                MarkRetr[i]=1;
                if(NumPI<NumProbingI){              /*PRI is not full*/
                pri[NumPI]=i;                        /*move user ID to PRI */
                pri[NumPII]=0;
                }
```

```
    }
}
```

Figure 3.11 C-like code of Packet Transmission Procedure in available slots

CHAPTER 4 SIMULATION AND ANALYSIS

Simulation was done using OPNET (Optimized Network Engineering Tools) to study the sector capacity performance of the protocol for voice and data traffic.

4.1 Traffics Characteristics and models

The voice traffic model used here is based on the work done by Brady [24] and assesses that speech sources generate periods of talkspurts and gaps. By assuming that a voice activity detector can be used to differentiate between principle talkspurts and principle gaps, voice traffic is characterized by two states Markov chain model displayed in figure 4.1. The system alternates between the ON and OFF states, which correspond to the talkspurts and idle periods of speech. In the ON State, voice packets are generated at a constant rate. No packet is generated in the OFF State. Time spend in each state is exponentially distributed with means α^{-1} for the OFF state and β^{-1} for the ON state [11]. A voice source would therefore require a reservation while in the ON state, but then could release the reservation during the OFF state when no packets are available for transmission.

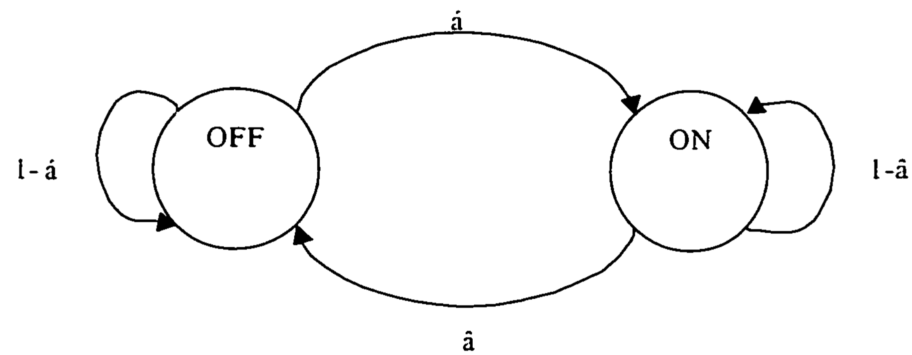


Figure 4.1 Two-state Markov model for voice

The ON-OFF voice traffic sources used in this study are modeled with the parameter values α^{-1} = 1.35s and β^{-1} = 1.0s [11]. Since voice packets must be delivered in real time, there is a maximum transmission delay allowed; any voice packet that has not been transmitted within 40ms of its generation time will be dropped at the source. For this study, the packet-dropping rate in this manner must not exceed 1%.

The data users are modeled to generate packets according to a Poisson process. Since data traffic is generally non real time traffic, no packets are discarded due to excessive delay. However, too many data users cause the infinite increase of mean delay. There is a breakdown value existing for the system capacity. We detect the breakdown value in our simulations.

All the settings for the simulations are listed in Table I.

Channel Data Rate (Mbps)	4.1472
Frame length (ms)	5
Slots per frame	54
Mini slots per first slot	24
Slots used per Sector	17
Slot size (bytes)	48
Command length (bytes)	4
Packet size (bytes)	44
Reservation Request length (bytes)	4
Acknowledgement length (bytes)	4
Average time spent in speaking for voice	1.0
Average time spent in silence for voice	1.35
Allowed transmission delay (ms)	40
Allowed packet dropping probability	1%

Table 1. Simulation Parameters

4.2 Simulation results

Figure 4.2 was simulated with 48 voice users in sector 1, 26 voice users in sector 2 and 26 voice users in sector 3. It shows that the packet dropping rate in sector 2 and 3 decreases with the increase of waiting frames of probing command retransmission. When the value of waiting frames is over 20, the interference becomes rather smaller and the packet-dropping rate is lower than 1%. Figure 4.3 also provides a similar result for data traffic. The mean delay of data packet transmission in sectors 2 and 3 is tested when there are 22 data users in sector 1, 15 in sector 2 and 15 in sector 3. In sector 1, the mean generating rate of data packet of each user is 1 packet per 0.006s. In sector 2 and 3, the mean generating rate is one packet per 0.008s. The results show that for the data traffic, the mean delay is very close to the value without probing process when the value of waiting frames is over 20.

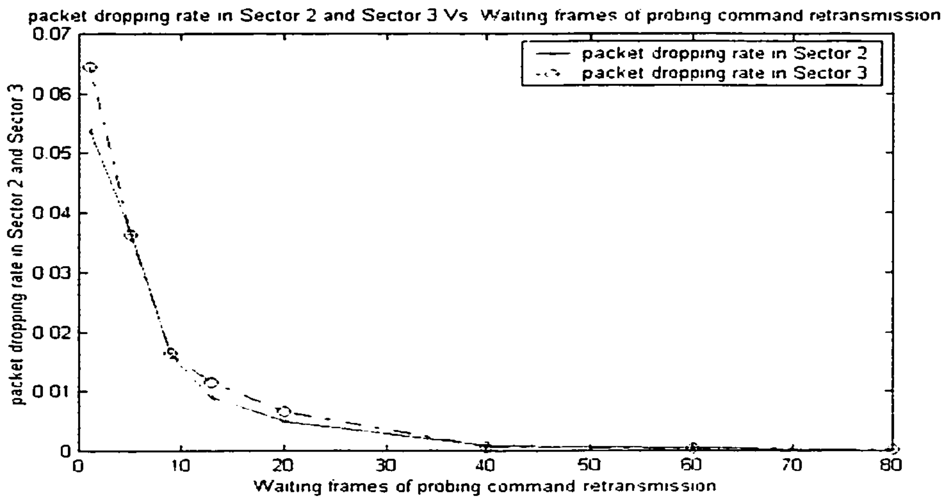


Figure 4.2 Packet dropping rate in sectors 2 and 3

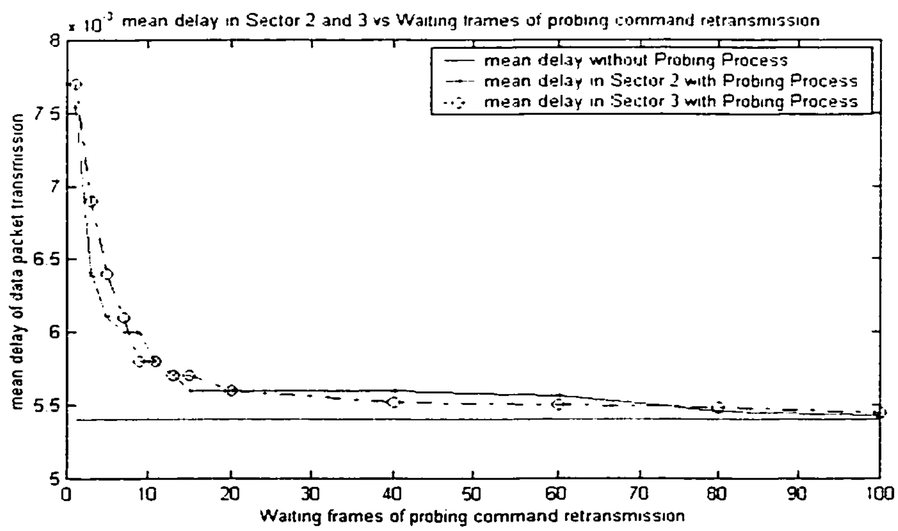


Figure 4.3 Mean delay in sectors 2 and 3

Figures 4.4 and 4.5 present the results when the packet transmission is first scheduled in PRI (the available slots in sector 2), then in PRII (the available slots in sector 3) for voice and data traffic with the proposed probing process. This means there is more traffic scheduled in subframes 2' and 5' than in subframes 3' and 6'. When the number of waiting frames to retransmit probing command is lower, the probing process is executed more frequently resulting in higher packet dropping in both sectors 2 and 3. However, it helps in accurately determining the available slots in both sectors 2 and 3. With increasing of waiting frames to retransmit probing command, the probing process is not executed as frequently, and the packet dropping rate is reduced. However available slots are not determined accurately resulting in higher packet dropping rate for voice traffic and longer mean delay for data traffic in sector 2 than in sector 3.

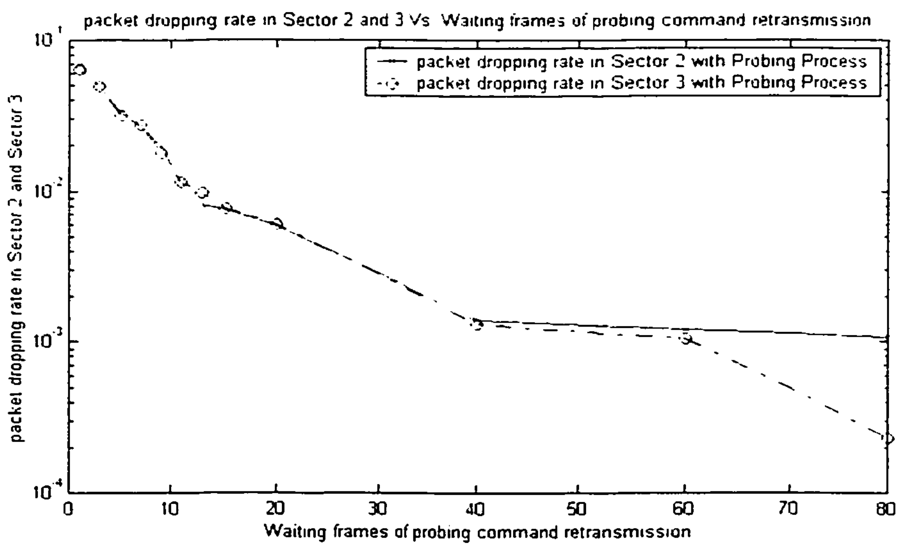


Figure 4.4 Packet dropping rate with sector 2 probed before sector 3

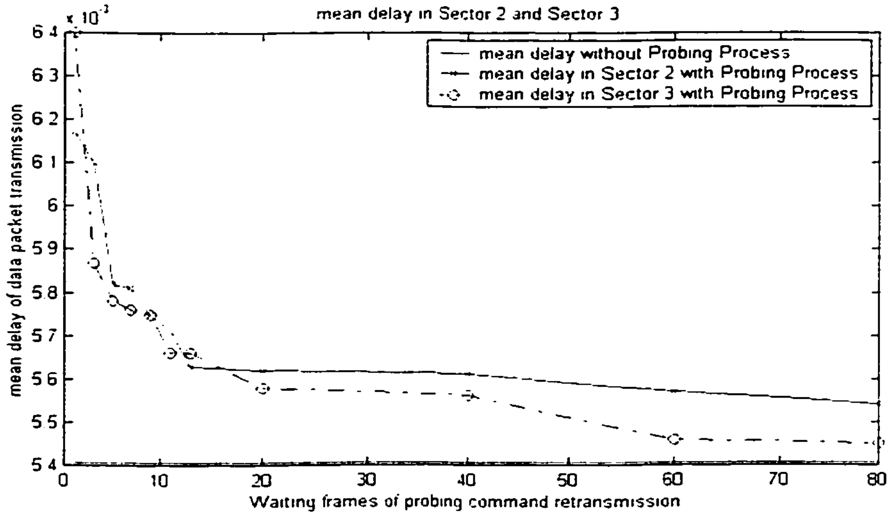


Figure 4.5 Mean delay with sector 2 probed before sector 3

From figures 4.2 to 4.5 it can be seen that a lower value of waiting frames of probing command retransmission results in the frequent probing packet transmission. This interferes with the communications of sector 2 and 3. To limit this interference a value of 25 is used as the waiting frames in the following simulations.

To take maximum advantage of the probing process, the traffic in other sectors should be low. The number of users in sector 2 and 3 is therefore taken as 20 for detecting the maximum capacity in sector 1. Figure 4.6 shows the packet-dropping rate with respect to the number of voice users in sector 1 with 20 voice users in sectors 2 and 3 separately. If a 1% packet dropping rate is allowed, CPRMA supplies 36 voice users capacity (see [22]), and CPRMA with SRA supplies 34 voice users capacity because the division of a frame into subframes in SRA method makes the users to transmit their packets in reserved subframes and caused more delay of transmission. However, CPRMA with probing process provides 44 users capacity.

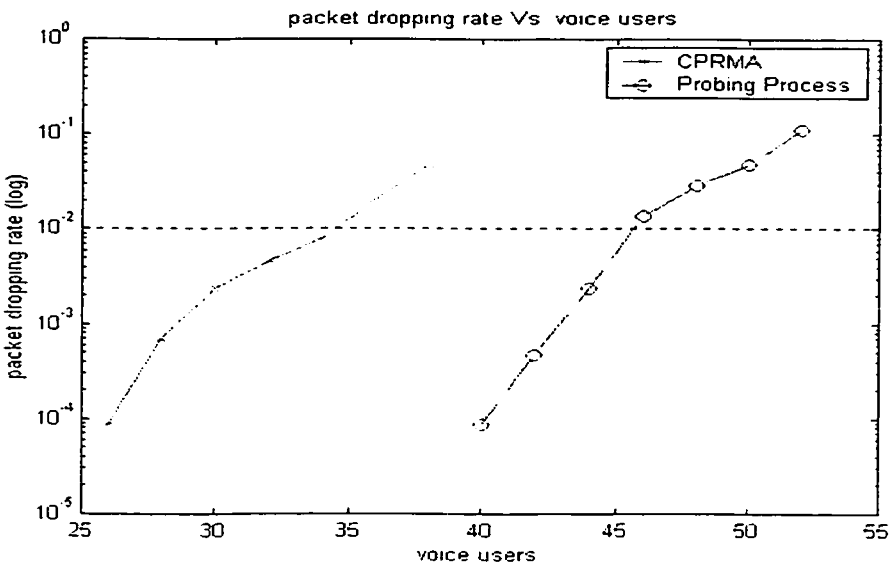


Figure 4.6 User Capacity for Voice traffic in sector 1

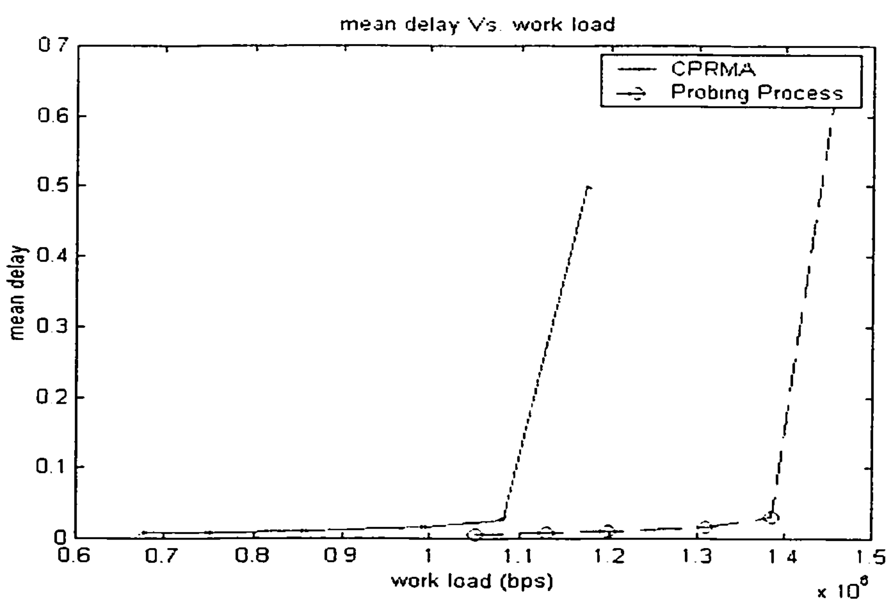


Figure 4.7 Link Capacity for Data traffic in sector 1

The aggregate traffic generated by all the data users in a sector is defined as the workload to the sector. The total channel capacity is assumed to be 4.1472 Mbps, so the link capacity of a sector is 1.3824 Mbps as 1/3 of the total channel capacity. In figure 4.7, sector 1 provides 1.08 Mbps workload as 78% of the link capacity that provides the breakdown point of system with CPRMA. CPRMA with probing process can support 1.38 Mbps workload as near 100% of the link capacity when the workload of sector 2 and 3 is 0.96 Mbps separately as 69% of the link capacity.

CHAPTER 5 SUMMARY AND CONCLUSIONS

We have proposed the probing process for both down link and uplink in fixed broadband wireless access networks, which is implemented in CPRMA with SRA radio resource allocation algorithm. We have shown that this approach provides a way to improve the bandwidth utilization with the avoidance of intracell and intercell cochannel interference when the network traffic is not balanced between different sectors in cellular, fixed, broadband wireless networks where a given frequency band is re-used in every sector of every cell.

The results shown in this study demonstrate that CPRMA with the probing process performs significantly better than CPRMA when we use them with SRA method. We see that the probing process detects the available slots unused by other sectors to provide a higher capacity for voice and data traffic.

The performance improvement that Probing Process offers can be seen in the simulation. For the voice traffic, Probing Process provides 23.5% more user capacity when there are 20 voice users in sectors 2 and 3 separately. For the data traffic, the system breakdown point increases 22% by Probing Process when the workload of Sector 2 and 3 is 0.96 Mbps separately as 69% of the link capacity.

Our study can be further extended in the following areas:

- The performance for the hybrid traffic including data and voice can be investigated by the simulation. The more complicated traffic models, such as web traffic model and video traffic model, can also be studied in the future.
- In our study, for the sake of simplification, we assigned certain slots and divided them into mini slots as Call Request Reservation mini slots. In real CPRMA, all available slots could be used as Call Request Reservation mini slots. As a possible continuation of our work, more dynamic slot assignment scheme can be studied to solve the bottleneck problem of Call Request Reservation.

REFERENCES

[1] J. Sanchez, R. Martinez, and M. W. Marcellin, " A survey of MAC protocols proposed for Wireless ATM", IEEE Network, pp. 52-62, Nov./Dec., 1997.

[2] N. Passas et al., "MAC protocol and traffic scheduling for wireless ATM networks," proposed for publication in ACM mobile networks and Appls. J., 1996.

[3] D. Raychaudhuri, "Wireless ATM networks: architecture, system design and prototyping," IEEE Pers. Commun., Aug. 1996, pp. 42-49.

[4] D. Raychaudhuri and N. D. Wilson, "ATM-based transport architecture for multiservices wireless personal communication network," IEEE JSAC, vol. 12, no. 8, Oct. 1994, pp. 1401-14.

[5] K. Maxwell, "Asymmetric digital subscriber line: Interim technology for the next forty years," IEEE Commun. Mag., vol. 34, no. 10, pp. 100-106, Oct 1996.

[6] C. Bisdikian, K. Maruyana, D. Seidman, and D. Serpanos, "Cable access beyond the hype: On residential broadband service over HFC networks," IEEE Commun. Mag., vol 34. pp. 128-135, Nov. 1996.

[7] T. K. Fong, P. S. Henry, K. K. Leung, X. Qiu, and N. K. Shankaranarayanan, "Radio Resource Allocation in Fixed Broadband Wireless Networks", IEEE Trans. Commun., vol. 46, pp. 808-818, June 1998.

[8] I. Brodsky, "What will LMDS hatch?" Telephony, pp. 22-30, Sept. 30, 1996

[9] M. Davis, "CATV installation basics," Telephony, pp. 42-44, Aug. 26, 1996

[10] H.P. Stern, "Design Issues Relevant to Developing an Integrated Voice/Data Mobile Radio System," IEEE Transactions on Vehicular Technology, vol. 39, no. 4, pp. 281-292, November 1990.

[11] T.S. Rappaport, *Wireless Communications Principles and Practice*. Prentice-Hall, 1996.

[12] W. C. Y. Lee, Mobile cellular Telecommunications Systems. New York: McGraw-Hill, 1989.

[13] I. Katzela and M. Naghshineh, "Channel Assignment Schemes for Cellular Mobile Telecommunication systems: A Comprehensive Survey," IEEE Personal Communications, pp. 10-31, June 1996.

[14] F. Borgonovo, L. Fratta, M. Zorzi, and A. Acampora, " Capture division packet access: A new cellular access architecture for future PCN's", IEEE Commun. Mag., pp.154-162, Sept. 1996.

[15] F. Borgonovo, M. Zorzi, L. Fratta, V. Trecordi, and G. Bianchi, "Capture-division packet access for wireless personal communications," IEEE J. Select. Areas Commun., Vol. 14, pp. 609-622, May 1996.

[16] J.F. Whitehead, 1996, private communications.

[17] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP Completeness. San Francisco, CA: Freeman, 1979.

[18] N. Kahale and P. E. Wright, "Dynamic global packet routing in wireless networks," in Proc. IEEE INFOCOM'97, Kobe, Japan, pp.1416-1423.

[19] X. Qiu and K. Chawla, "Resource Assignment in a Fixed Wireless System," IEEE Commun. Letters, Vol. 1, No. 4, pp. 108-110, July 1997.

[20] K. K. Leung and A. Srivastava, "Dynamic Allocation of Downlink and Uplink Resource for Broadband Services in Fixed Wireless Networks", IEEE Journal on Selected Areas in Commun., vol. 17, No. 5, pp. 990-1006, May 1999.

[21] D. Bertsekas and R. Gallager, Data Networks. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[22] G. Bianchi, F. Borgonovo, L. Fratta, L. Musumeci, and M. Zorzi, "C-PRMA: A Centralized Packet Reservation Multiple Access for Local Wireless Communications", IEEE Transactions on Vehicular Technology, vol. 46, No. 2, May 1997.

[23] A. Srivastava, N.K. Shankaranarayanan, and K.K. Leung, "Sector-Based Resource Allocation for Broadband Fixed Wireless Networks", IEEE VTC'98.

[24] P. T. Brady, "A model for on-off speech patterns in two-way conversation", Bell Syst. Tech. J. 48(7) (Sep. 1969) 2445-2472.

APPENDIX A

The following figures show the main procedure, RS procedure and CI procedure of C-PRMA.

```
void Main() {

    /* execute operations depending on the status of previous uplink slot*/

    if (previous_slot==AVAIL){

        if (contention_detected) Fc=1;/* update contention flag*/

        else Fc=0;

        for (j=vwinners){           /*for all contention winners j */

            c[j] = l[j] - w[j];     /* reset c[j]              */

            RS(j);                  /* schedule reservation j   */

        }

    }

    else {

        i = reserved_MS;

        read_data_packet();

        if ((successful_transmission) and (MI_bit == 1 ) ) {

            c[i] =c[i] + t[i];

            r[i] = 0;
```

```
            if ( c[i] <= l[i]) RS(i);  /* new transmission scheduled  */
            else {                 /* uncorrect packet received   */
                    r[i]++;              /* increase retransm. counter  */
                    RS(i);               /* schedule retransmission     */
            }
    }

/* update polling register and schedule users                     */
        j = PR[0];               /* temporarily store PR[0]     */
        shift_PR();              /*shift PR to the right        */
        if (j != 0) {            /* PR[0] reservation j lost;   */
                r[j] = 0;
                c[j] = c[j] + t[j];
                if (c[j] <= l[j] RS(j)  /* schedule new reserv. for j  */
        }
        for (all active MS's i ) {
                c[i]--;              /* decrease counters c[i]      */
                if (c[i] == l[i]) {   /* new packet arrived at MS i   */
                        r[i] = 0;
                        RS(i);           /* schedule reservation for user i */
                }
        }
        /* Generate the command to be issued in the current slot    */
```

```
        command = CI();          /* call command Issuing proc.   */

} /* end Main() */
```

Figure A.1 C-like code for the main procedure of CPRMA

```
void RS(i) {

        /* place reservation in PR, starting from maximum available position */

        k = c[i];

        while (k >= 0) {

                if (PR[k] = i) {    /* empty position found            */

                PR[k] = i;          /* MSi is scheduled                */

                return;

                }

                j = PR[k];          /* pos., not empty: check priority  */

                if (p[j] < p[i]) {  /* lower priority: replace j with i */

                        PR[k] = i;      /* MSi is scheduled             */

                        RS(j);          /* reschedule lower priority res. */

                        return;

                }

                k--;                /* a lower PR position     */

        }


        /* no available position found in PR: the packer is lost     */

        /* schedule reservation for next packet of MS i              */

        c[i] = c[i] + t[i];

        if (c[i] <= l[i] RS(i)  /* Res. schedule only if pck. ready */

} /* end RS(i) */
```

Figure A.2 C-like code for the RS procedure

```
int CI() {

        /* if position 0 of PR is not empty, issue a reserved slot    */

        if (PR[0] != 0) {

                CRS++;

                return PR[0];

        }


        /* if counter of consecutive reserved slots overflow the threshold, or  */

        /* the contention flag is on, issue an available slot                */

        if ((CRS > CT) || (FC == 1) {

        CRS = 0;

        return 0;

        }


        /* if none of the previous conditions is verified issue a reserved slot */

        k = 1;

        z = MAXINT;       /* very high integer value      */

        pos = -1;         /* "impossible" PR position     */

        while (k <= L+1) {   /* L+1 id the size of the PR  */


                if (PR[k] !=0){    /* a non-empty position found in PR  */

                        j = PR[k];

                        if (r[j] = 0){ /* MS j has highest service priority */
```

```
                CRS++;
                PR[k] = 0;
                return j;  /* return reserved slot for MS j     */
            }
            else if (r[j]<z){
                /* found res. with greater priority than previous  */
                pos = k;
                z = r[pos];  i  /* mark new reservation  */
            }
        }
        k++;
    }

    /* no reservation with highest priority found     */
    if (pos != -1) {     /* found at least one res. serve it  */
        CRS++;
        j = PR[pos];
        PR[pos] = 0;
        return j;
    }
    else {              /* PR empty - return available slot  */
        CRS = 0;
        return 0;
```

```
        }

} /* end CI() */
```

Figure A.3 C-like code for the CI prodedure

APPENDIX B

Our study was developed on a simulation package called Optimized Network Engineering Tools (OPNET). The simulation is built with independent building blocks called process models. The operations of those process models are specified by finite state machines, translated into C code.

Figure A.1 shows the network model of users employing packet generator and queue. We give the process model of voice packet generator in figure A.2 and the process model of base station in figure A.3. The code for the voice packet generator is given in the end of thesis.

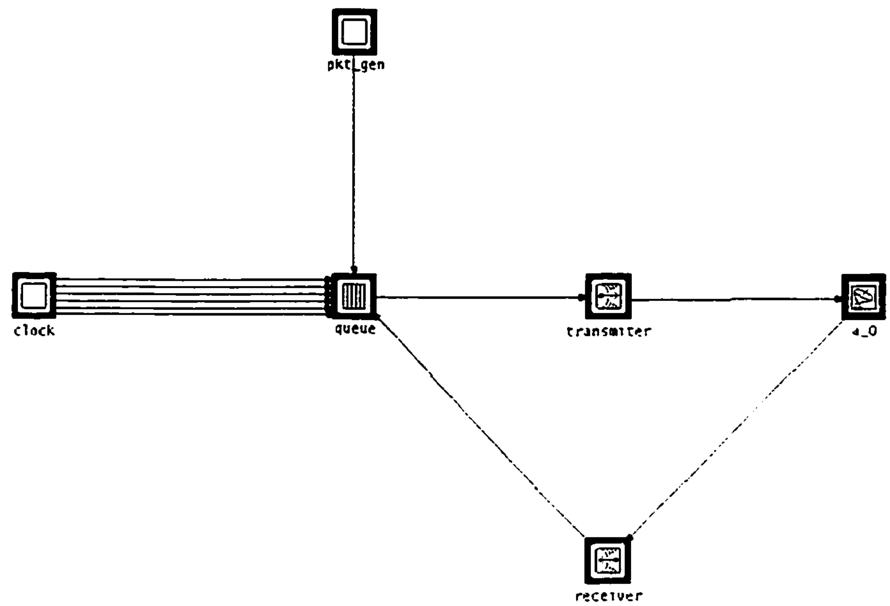


Figure B.1 network model of users

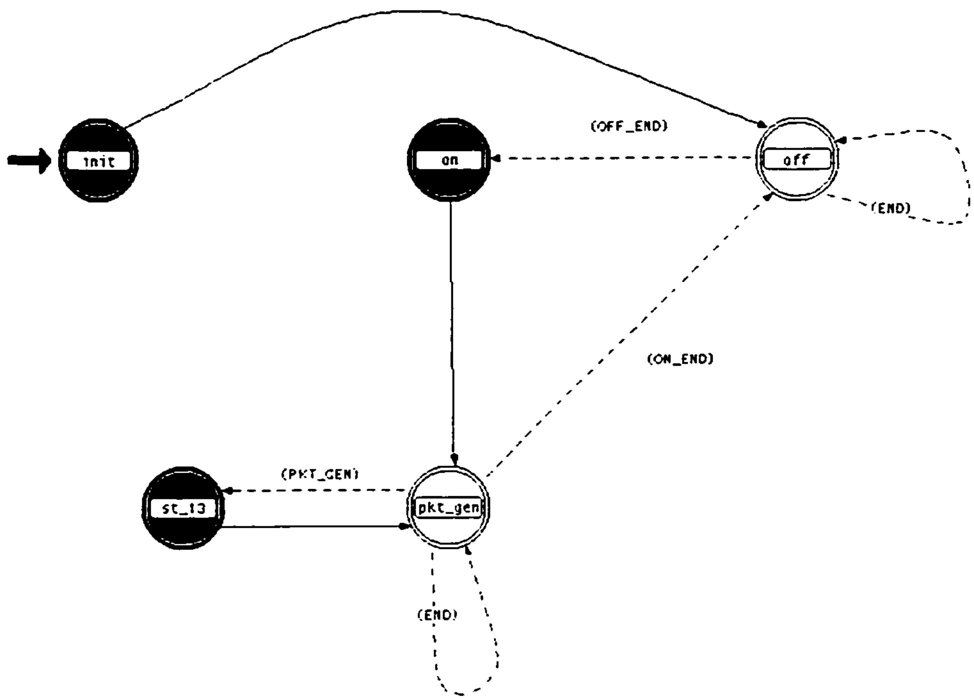


Figure B.2 Process model of voice packet generator

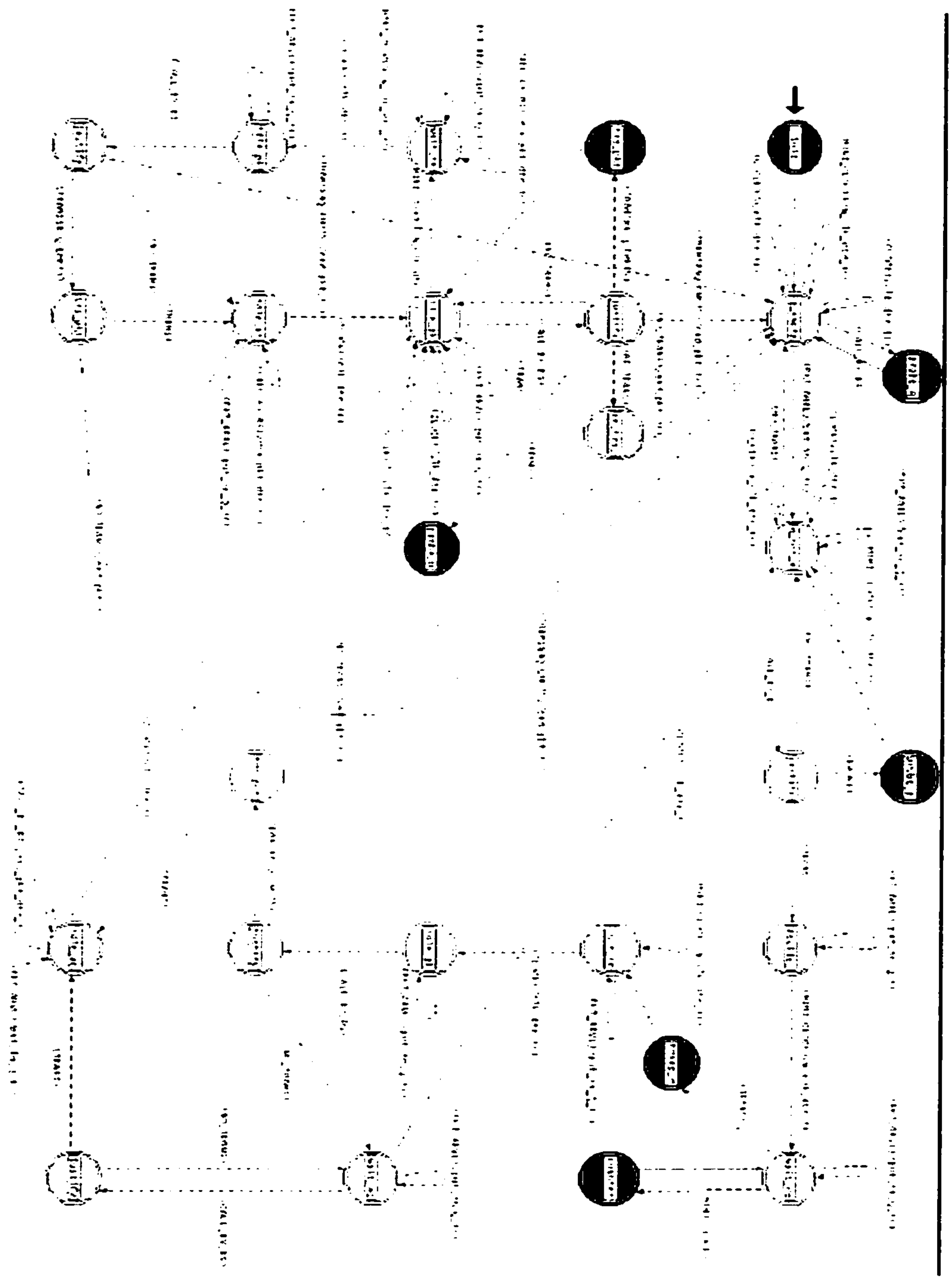


Figure B.3 Process model of base station

Process Model Comments

Process Model Interface Attributes

Attribute **begsim intrpt** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	enabled	N/A
Default Value:	disabled	YES
Data Type:	toggle	N/A
Attribute Description:	Private	N/A
Comments:		YES
	This attribute specifies whether a 'begin simulation interrupt' is generated for a processor module's root process at the start of the simulation.	
Symbol Map:	NONE	YES

Attribute **doc file** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	nd_module	N/A
Default Value:	nd_module	YES
Data Type:	string	N/A
Attribute Description:	Private	N/A
Comments:		YES
	This attribute defines the name of the online help file which will be displayed when the user invokes help for this object.	
Symbol Map:	NONE	YES

Attribute **endsim intrpt** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	disabled	N/A
Default Value:	disabled	YES
Data Type:	toggle	N/A
Attribute Description:	Private	N/A
Comments:		YES
	This attribute specifies whether an end simulation interrupt' is generated for a processor module's root process at the end of the simulation.	
Symbol Map:	NONE	YES

Attribute **failure intrpts** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	disabled	N/A
Default Value:	disabled	YES
Data Type:	enumerated	N/A

Attribute Description:	Private	N/A
Comments:		YES
	This attribute specifies whether failure interrupts are generated for a processor module's root process upon failure of nodes or links in the network model.	
Symbol Map:	NONE	YES

Attribute **intrpt interval** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	disabled	N/A
Default Value:	disabled	YES
Data Type:	toggle double	N/A
Attribute Description:	Private	N/A
Units:	sec.	YES
Comments:		YES
	This attribute specifies how often regular interrupts are scheduled for the root process of a processor module.	
Symbol Map:	NONE	YES

Attribute **priority** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	0	N/A
Default Value:	0	YES
Data Type:	integer	N/A
Attribute Description:	Private	N/A
Low Range:	-32767 inclusive	YES
High Range:	32767 inclusive	YES
Comments:		YES
	This attribute is used to determine the execution order of events that are scheduled to occur at the same simulation time	
Symbol Map:	NONE	YES

Attribute **recovery intrpts** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	disabled	N/A
Default Value:	disabled	YES
Data Type:	enumerated	N/A
Attribute Description:	Private	N/A
Comments:		YES
	This attribute specifies whether recovery interrupts are scheduled for the processor module's root process upon recovery of nodes or links in the network model.	
Symbol Map:	NONE	YES

Attribute **subqueue** properties

Property	*Value*	*Inherit*

Assign Status:	set	
Initial Value:	(...)	N/A
Default Value:		YES
Data Type:	compound	N/A
Attribute Description:	Private	N/A
Comments:		YES
	This operation attribute permits the addition and deletion of subqueues within the queue module.	
Symbol Map:	NONE	YES

Attribute **super priority** properties

Property	*Value*	*Inherit*
Assign Status:	set	
Initial Value:	disabled	N/A
Default Value:	disabled	YES
Data Type:	toggle	N/A
Attribute Description:	Private	N/A
Comments:		YES
	This attribute is used to determine the execution order of events that are scheduled to occur at the same simulation time	
Symbol Map:	NONE	YES

Header Block

```
#define ON_END (op_intrpt_type()==OPC_INTRPT_SELF && op_intrpt_code()==1)
#define OFF_END (op_intrpt_type()==OPC_INTRPT_SELF && op_intrpt_code()==2)
#define PKT_GEN (op_intrpt_type()==OPC_INTRPT_SELF && op_intrpt_code()==0)
#define LEN_SLOT 0.005

#define END (op_intrpt_type()==OPC_INTRPT_ENDSIM)
```

Temporary Variable Block

```
Packet * pk_ptr;
```

Function Block

```
off_beg()
    {
/*  double off_rand_time; */
    FIN (off_beg())

    op_intrpt_clear_self().
/* calculate the random time for generating the next pkt
    off_rand_time = op_dist_outcome (off_dist_ptr);

schedule intrpt for the delivery of the next pkt
    op_intrpt_schedule_self (op_sim_time () + off_rand_time, 1); */

    op_intrpt_schedule_self (op_sim_time () + op_dist_exponential(1.35), 3);
```

```
    FOUT

    ;

on_beg()
    ;
/*  double on_rand_time;*/
    FIN (on_beg())

    op_intrpt_clear_self();
/* calculate the random time for generating the next pkt
    on_rand_time = op_dist_outcome (on_dist_ptr);

 schedule intrpt for the delivery of the next pkt
    op_intrpt_schedule_self (op_sim_time () + on_rand_time, 3);*/
    op_intrpt_schedule_self (op_sim_time(), 0);
    op_intrpt_schedule_self (op_sim_time () + op_dist_exponential(1.0), 3);

    FOUT

    ;
```

forced state init			
attribute	*value*	*type*	*default value*
name	init	string	st
Enter Execs	(empty)	textlist	(empty)
Exit Execs	(empty)	textlist	(empty)
status	forced	toggle	unforced
doc file	pr_state	string	pr_state

transition init -> off			
attribute	*value*	*type*	*default value*
name	tr_33	string	tr
condition		string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition

forced state on			
attribute	*value*	*type*	*default value*
name	on	string	st
Enter Execs	(See below.)	textlist	
Exit Execs	(empty)	textlist	(empty)
status	forced	toggle	unforced
doc file	pr_state	string	pr_state

Enter Execs on

```
op_intrpt_schedule_self (op_sim_time () + op_dist_exponential(1.0), 1);
op_intrpt_schedule_self (op_sim_time(), 0);
```

transition on -> pkt_gen

attribute	*value*	*type*	*default value*
name	tr_35	string	tr
condition		string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition

unforced state off

attribute	*value*	*type*	*default value*
name	off	string	st
Enter Execs	(See below.)	textlist	
Exit Execs	(empty)	textlist	(empty)
status	unforced	toggle	unforced
doc file	pr_state	string	pr_state

Enter Execs off

```
op_intrpt_clear_self();
op_intrpt_schedule_self (op_sim_time () + op_dist_exponential(1.35), 2);
```

transition off -> on

attribute	*value*	*type*	*default value*
name	tr_41	string	tr
condition	OFF_END	string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition

transition off -> off

attribute	*value*	*type*	*default value*
name	tr_53	string	tr
condition	END	string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition

unforced state **pkt_gen**			
attribute	*value*	*type*	*default value*
name	pkt_gen	string	st
Enter Execs	(empty)	textlist	(empty)
Exit Execs	(See below.)	textlist	
status	unforced	toggle	unforced
doc file	pr_state	string	pr_state

Exit Execs **pkt_gen**

```
op_intrpt_schedule_self(op_sim_time()+LEN_SLOT,0);
```

transition **pkt_gen -> st_13**			
attribute	*value*	*type*	*default value*
name	tr_37	string	tr
condition	PKT_GEN	string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition

transition **pkt_gen -> off**			
attribute	*value*	*type*	*default value*
name	tr_39	string	tr
condition	ON_END	string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition

transition **pkt_gen -> pkt_gen**			
attribute	*value*	*type*	*default value*
name	tr_51	string	tr
condition	END	string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition

forced state **st_13**

attribute	*value*	*type*	*default value*
name	st_13	string	st
Enter Execs	(See below.)	textlist	
Exit Execs	(empty)	textlist	(empty)
status	forced	toggle	unforced
doc file	pr_state	string	pr_state

Enter Execs **st_13**

```
pk_ptr=op_pk_create_fmt("wangq_packet");
op_pk_send (pk_ptr, 0);
```

transition **st_13 -> pkt_gen**

attribute	*value*	*type*	*default value*
name	tr_47	string	tr
condition		string	
executive		string	
color	RGB000	color	RGB333
drawing style	spline	toggle	spline
doc file	pr_transition	string	pr_transition