

# **Scheduling and Call Admission Control (CAC) in IEEE 802.16 Mesh Networks**

**M.Tech. Project First Stage Report**

Submitted in partial fulfillment of the requirements  
for the degree of

**Master of Technology**

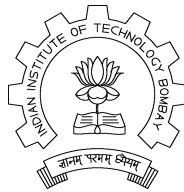
by

**Jeevan B. Chalke**

**Roll No: 06329011**

under the guidance of

**Prof. Anirudha Sahoo**



Department of Computer Science and Engineering  
Indian Institute of Technology, Bombay  
Mumbai

# Acknowledgements

I would like to thank my guide **Prof. Anirudha Sahoo**, for his invaluable help and guidance during the course of my MTP I stage. I also thank **Dr. Ajay Bakre** for his support. I am highly indebted to him for constantly encouraging me by giving his criticisms on my work.

# Declaration

This report is based on the input from sources that I have found on the Internet. I acknowledge the use of these sources carefully and diligently. The figures used in the report have been taken from these sources. The source for each figure has been mentioned along with the figure. A list of the sources that were referred for the creation of the report appears in the bibliography.

Jeevan B. Chalke

July, 2007

CSE, IIT Bombay

# Abstract

The IEEE 802.16 - WiMax provides a mechanism for deploying broadband wireless networks in geographically large areas. It supports three modes: Point to Point (P2P), Point to Multi-Point (PMP) and Mesh. The MAC and physical layer of WiMax is carefully designed to support several real time applications with quality of service (QoS) guarantees. In absence of proper scheduling and admission control the system cannot provide the promised QoS guarantees. Unfortunately the IEEE 802.16 standard does not specify any scheduling or admission control mechanism in PMP and Mesh mode.

In the mesh mode, the network topology is a tree rooted at base station and the problem is to determine the routing, link scheduling and call admission control for the tree. This report gives the overview of routing, scheduling and call admission control in mesh mode to provide certain QoS guarantees. The report also describes some work done in area of call admission control on WiMax chip in PMP mode.

**Keywords:** IEEE 802.16 networks, Mesh Networks, Quality of Service (QoS), Scheduling, Call Admission Control (CAC), Routing algorithms

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Literature Survey</b>	<b>4</b>
2.1	Overview of 802.16 mesh mode . . . . .	4
2.1.1	Mesh Frame Structure . . . . .	5
2.1.2	Scheduling in mesh topology . . . . .	5
2.2	Routing Algorithms . . . . .	8
2.3	Scheduling Algorithms . . . . .	10
2.4	Call Admission Control . . . . .	13
<b>3</b>	<b>Work Done/Implementation</b>	<b>14</b>
3.1	Intel Pro/Wireless 5116 Broadband Interface . . . . .	14
3.2	Implementation . . . . .	15
<b>4</b>	<b>Problem Definition</b>	<b>18</b>
4.1	Designing Scheduling and/or Call Admission Control . . . . .	18
4.2	Proposed Plan . . . . .	18
<b>5</b>	<b>Conclusion and Future Work</b>	<b>19</b>
	<b>Bibliography</b>	<b>20</b>
	<b>Appendices</b>	<b>22</b>
<b>A</b>	<b>Call Admission Control Prototypes</b>	<b>22</b>
<b>B</b>	<b>Dynamic Service Related Prototypes</b>	<b>25</b>

# Chapter 1

## Introduction

The IEEE 802.16 protocol for wireless metropolitan area networks (WMAN) has been recently standardized to meet the needs of wireless broadband access. The IEEE 802.16d [1] supports point-to-point, point-to-multipoint and mesh topology and provides a scalable solution for extension of a fiber optic backbone. A WiMax base station can offer greater coverage area around five miles with line-of-sight (LOS) transmission within bandwidth of up to 70 Mbps. WiMax is suitable for many neighborhoods that are too remote to receive Internet access via cable or DSL, and for anyplace where the cost of laying or upgrading landline to broadband access is expensive. When backhaul based WiMax is deployed in mesh mode, it not only increases the wireless coverage but also provides features like lower backhaul deployment cost, rapid building, easy deployment, robustness and re-configurability.

The 802.16 mesh networks are mainly used to provide cost effective Internet access to sparsely populated areas. Thus the network topology is a tree rooted at base station and the problem is to determine the routing, link scheduling and call admission control for the tree, either jointly or separately. In community wireless networks, most of the nodes are either stationary or minimally mobile. Hence the focus of routing algorithm is on improving the network capacity of the performance of the individual transfers, instead of coping with mobility or minimizing power usage.

Scheduling and call admission control plays an important role in increasing the spatial reuse, achieving high throughput and providing fair access to the subscriber stations. The call admission control analyzes the request and determines the acceptance or rejection of request, and if accepted, its QoS attributes are registered in service flow database. Scheduling then uses this service flow database and routing algorithm to schedule the connection in a frame.

The remainder of this report is organized as follows; chapter 2 will present the literature survey in the area of IEEE 802.16 mesh networks. Where as chapter 3 will describe the work done in Intel on WiMax board in 802.16 PMP mode. Chapter 4 presents the problem definition and proposed plan followed by conclusion and future work in chapter 5.

# Chapter 2

## Literature Survey

### 2.1 Overview of 802.16 mesh mode

The IEEE 803.16 provides both PMP and mesh topologies. The main difference between PMP and mesh mode is that in PMP mode, traffic only occurs between base station (BS) and subscriber stations (SSs) where as in mesh mode; traffic can be routed through other SSs as well.

As shown in Figure 2.1, the overall area is divided into meshes and managed by a single node called as mesh base station (MBS). It serves as the interface for WiMax base mesh to the external network. A transmission can occur between two SSs within a mesh or within two meshes. A transmission within a single mesh may or may not involve the MBS where as transmission between SSs in two different meshes must route through the MBS. In other case, SS routes its packets to its MBS via several SSs in between, MBS then route it to BS which forwards the packet to the destination SS's MBS. Finally the MBS of destined SS routes the packet to the SS via several SSs in the route. For example, in Figure 2.1 packet from SS3 to SS4 will take path as: SS3 - SS2 - MBS (Mesh 1) - BS - MBS (Mesh 2) - SS4. During this MTP I am going to focus on single mesh.

The IEEE 802.16 mesh mode MAC supports both centralized and distributed scheduling. Centralized scheme is widely used to establish the high speed broadband mesh connection, in which MBS coordinates the radio resource allocation within the mesh network. We will see scheduling in mesh mode in next section before that we will see IEEE 802.16 frame structure.

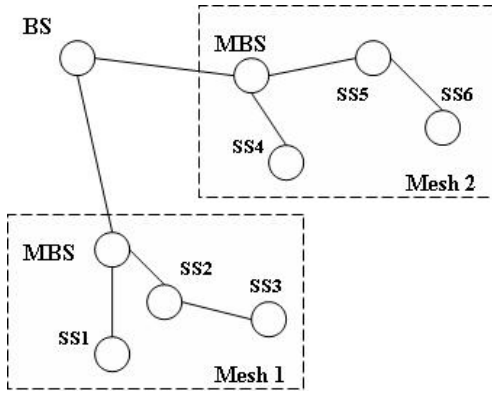


Figure 2.1: A typical mesh network

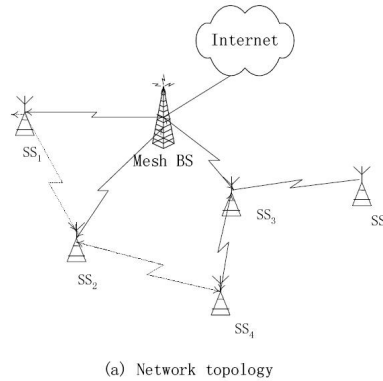
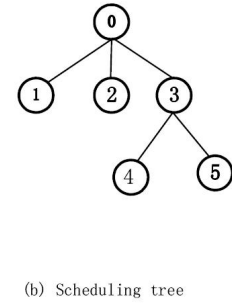


Figure 2.2: An example of an IEEE 802.16 based mesh network (source: [4])



### 2.1.1 Mesh Frame Structure

Contrary to PMP mode, there are no separate uplink and downlink subframes in the mesh mode. The mesh mode only supports the TDMA - Time Division Multiple Access for channel access among the MBS and SSs. A mesh frame consists of control and data subframe. The control subframe serves two functions: network control and schedule control. The data subframe is shared between centralized and distributed scheduling. Figure 2.3 shows the IEEE 802.16 mesh frame structure. In a network control subframe, mesh network configuration (MSH-NCFG) and mesh network entry (MSH-NENT) packets provide some basic level of communication for nodes to exchange network configuration information. In a schedule control subframe, the mesh centralized scheduling (MSHCSCS) and mesh centralized scheduling configuration (MSHCSCF) packets are used for transmission bursts corresponding to centralized messages, and rest is allocated to transmission bursts containing mesh distributed scheduling (MSH-DSCH) packets for distributed scheduling. The data subframe consists of minislots. Each minislot, except the last minislot, consists of  $\lceil (OFDM\ symbols/frame - MSH-CTRL-LEN \times 7)/256 \rceil$  symbols, where MSH-CTRL-LEN is the length of the 802.16 mesh control plane. A scheduled allocation consists of one or more minislots.

### 2.1.2 Scheduling in mesh topology

There will be certain nodes that provide the BS function of connecting the Mesh network to the backhaul links. When using Mesh centralized scheduling (described below), these BS nodes

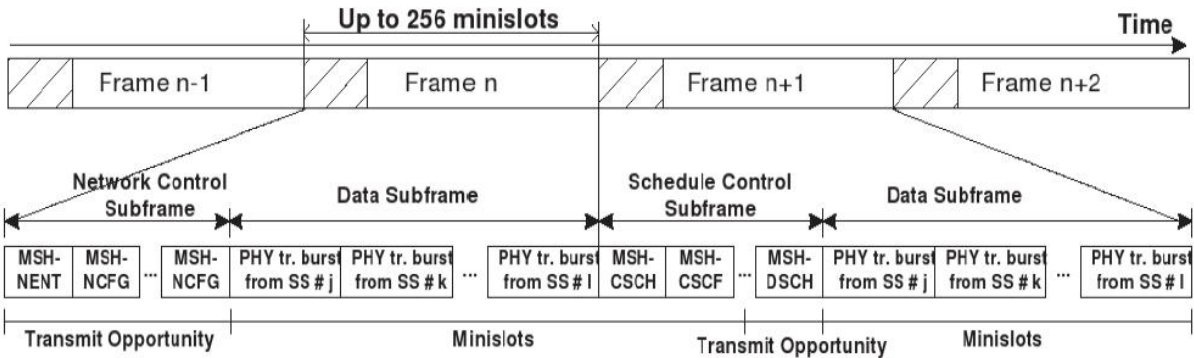


Figure 2.3: IEEE 802.16 mesh frame structure (source: [6])

perform much of the same basic functions as do the BS in PMP mode. Thus, the key difference is that in Mesh mode all the SSs may have a direct links with other SSs. Also, there is no need to have direct link from an SS to the BS of the Mesh network. This connection can be provided via other SSs. Communication in all these links shall be controlled by a centralized algorithm (either by the BS) or decentralized by all nodes periodically.

### Distributed Scheduling

The stations that have direct links are called neighbors and shall form a neighborhood. A node's neighbors are considered to be "one hop" away from the node. A two-hop extended neighborhood contains all the neighbors of the neighborhood. In the coordinated distributed scheduling mode, all the stations (BS and SSs) shall coordinate their transmissions in their extended two-hop neighborhood. The coordinated distributed scheduling mode uses control portion of each frame to regularly transmit its own schedule and proposed schedule changes on a PMP basis to all its neighbors. Within a given channel all neighbor stations receive the same schedule transmissions. All the stations in a network shall use this same channel to transmit schedule information in a format of specific resource requests and grants. Coordinated distributed scheduling ensures that transmissions are scheduled in a manner that does not rely on the operation of a BS, and that are not necessarily directed to or from the BS. Uncoordinated distributed scheduling can be used for fast, ad-hoc setup of schedules on a link-by-link basis. Uncoordinated distributed schedules are established by directed requests and grants between two nodes, and shall be scheduled to ensure that the resulting data transmissions (and the request and grant packets themselves) do

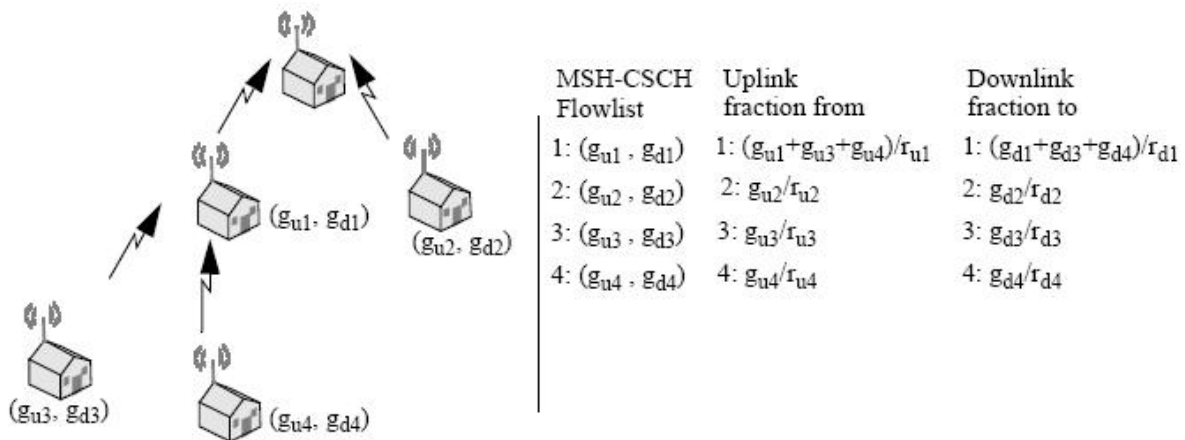


Figure 2.4: Centralized scheduling flow-mechanism (source: [1])

not cause collisions with the data and control traffic scheduled by the coordinated distributed nor the centralized scheduling methods.

The differences between coordinated and uncoordinated distributed scheduling is that in the coordinated case, the MSH-DSCH messages are scheduled in the control subframe in a collision free manner; whereas, in the uncoordinated case, MSH-DSCH messages may collide.

### Centralized Scheduling

In centralized scheduling the scheduled transmissions for the SSs is defined by the BS. The BS determines the flow assignments from the resource requests from the SSs. Then, the SSs determine the actual schedule from these flow assignments by using a common algorithm that divides the frame proportionally to the assignments. Thus, the BS acts just like the BS in a PMP network except that not all of the SSs have to be directly connected to the BS, and the assignments determined by the BS extends to those SSs not directly connected to the BS. The SS resource requests and the BS assignments are both transmitted during the control portion of the frame. A simple example of the use of the centralized scheduling flow-mechanism in MSH-CSCH is as shown in Figure 2.4.

The number of frames during which the CSCH schedule is valid is limited by the number of frames it takes to aggregate and distribute the next schedule. Each node uses the newly received schedule to compute the following:

- The time the node shall transmit this schedule (if eligible) for nodes further down the transmission tree
- The frame where the last node in the transmission tree will be receiving this schedule
- The original transmission time by the Mesh BS of this schedule

To compute this, the node uses the routing tree from the last MSH-CSCF messages as modified by the link updates of the last MSH-CSCH message (which dictates the size of MSH-CSCH messages) and the following steps:

1. The Mesh BS transmits first in a new frame
2. Then, the eligible children of the Mesh BS (i.e., nodes with a hop count equals 1), ordered by their appearance in the routing tree, transmit
3. Then, the eligible children of the nodes from Step 2) (i.e., nodes with a hop count that equals 2), also ordered by their appearance in the routing tree, transmit
4. The process continues until all eligible nodes in the routing tree have transmitted

## 2.2 Routing Algorithms

There are few fundamental challenges in routing over wireless mesh networks. Wireless routing has to ensure robustness against a wide spectrum of soft and hard failures, links with intermediate loss rates, from several channel disconnections, nodes under denial-of-service (DOS) attacks, and failing nodes. Challenges in routing are to address both these issues and at the same time it should be scalable enough to handle large node population. This section gives an overview of some routing algorithms which deals with some of the above issues.

### Interference Aware Routing

Multiple-access interference is a major limiting factor for wireless communication systems. Interference in wireless systems is one of the most significant factors that limit the network capacity and scalability. The motivation behind the interference aware routing proposed in [5] is to design an efficient multi-hop routing and scheduling scheme that is interference aware, and hence maximizes parallel transmission, providing high throughput and scalability.

The scheme proposed in [5] includes a novel interference aware route construction algorithm and an enhanced centralized mesh scheduling scheme, which consider both traffic load demand and interference conditions. This provides better spatial reuse and hence higher spectral efficiency. The scheme is based on tree based routing framework. The paper considers WiMax based mesh which is managed by Mesh BS. The metric considered for routing is blocking metric  $B(k)$ . The Blocking Metric  $B(k)$  of a multihop route indicates the number of blocked/interfered nodes by all the intermediate nodes along the route from the root node towards the destination node  $k$ . The paper defines blocking value  $b(\eta)$  of a node  $\eta$ , as the number of blocked/interfered nodes when  $\eta$  is transmitting. Thus blocking metric of a route is summation of the blocking values of nodes that transmits or forwards packets along the route.

## Routing for throughput maximization

In Interference-Aware routing, the authors considered only a blocking metric of a route. To overcome this limitation, [6] extended this idea, and also took the number of packets into account by defining the blocking metric of the node  $v$  to be the number of blocked nodes multiplied by the number of packets at the node  $v$ . Hence the modified blocking metric for a node  $v$  is:  $B(v) = (\text{Number of nodes blocked by } v) \times (\text{number of packets at } v)$ . Finally the path with the minimum blocking metric is selected as  $P_0 = \arg \min B(P)$ .

Authors considered problem of routing based on centralized scheme in which the MBS acts as a centralized scheduler for the entire network. The authors used following constraints for any transmission: 1) A node cannot send and receive simultaneously. 2) There may be only one transmitter in the neighborhood of a receiver. 3) There may be only one receiver in the neighborhood of a transmitter.

The problem can then be stated as follows. Given  $G = (V, E)$ , where set  $V$  consists of MBS  $v_0$  and SSs  $v_1, v_2, \dots, v_n$ , such that  $(v_i, v_j)$  belongs to  $E$  if and only if  $v_i$  and  $v_j$  are within the transmission range of each other. SS  $v_i$  needs to send  $w(v_i)$  packets to the base station. The objective is to find a feasible routing tree and a schedule for the packets such that the number of timeslots required is minimized. In proposed system model, the routing tree (scheduling tree) is constructed in two conditions. First, when a new node enters the network, the scheduling tree is updated according to broadcast messages MASH-NCFG and MASH-NENT from the new node. Then MBS recalculates the routing node and reconfigures the network by broadcasting the MASH-CSCH message to the SSs. Second, the MBS also periodically recomputes the rout-

ing tree by considering new updated throughput requirements, and changing the routing tree if required.

## Fixed Routing for Supporting QoS

Good routing algorithm should have the same route for all the traffic at a node irrespective of its node of origin and whether it is real-time or data traffic. Also the route should be fixed i.e. it should not vary even when the wireless channels are time varying. To provide QoS guarantee, we need to reserve resources along the path. This is possible only if we do not change the route of a connection unless it is absolutely necessary. Based on these, [2] proposed a fixed routing algorithm which will work well for both real as well as data traffic.

Let  $r_k(i, j)$  be the assigned transmission rate,  $X_k(i, j)$  the external arrivals and  $Y_k(i, j)$  the arrivals from other nodes to node  $i$  for output link  $(i, j)$  during the frame  $k$ . If the schedule is fixed then the link  $(i, j)$  is always assigned  $n_{i,j}$  slots in a frame, such that

$$\sum_{i=1}^M n_i = N \quad (2.1)$$

Where,  $N$  is the number of slots in a mesh frame. Also let  $\lambda_{i,j} = E[X_k(i, j)]$ . Then

$$Q_{k+1}(i, j) = \max(0, (Q_k(i, j) + Y_k(i, j) - n_{i,j}r_k(i, j))) + X_k(i, j) \quad (2.2)$$

And  $Q_k(i, j)$  be the queue length at node  $i$  for the output link  $(i, j)$  in the beginning of frame  $k$ . For the queue to be stable, following condition should be satisfied,

$$n_{i,j}E[r(i, j)]E[X(i, j) + Y(i, j)] = \lambda_{i,j} + E[Y(i, j)] \quad (2.3)$$

Where the expectation  $E[Y(i, j)]$  is under the stationary distribution.

There are various other routing algorithms such as Routing for Throughput Enhancement using Concurrent Transmissions as proposed in [7], ROMER [10] and Multi-Radio Link-Quality Source Routing (MR-LQSR) [11] designed for 802.16 mesh networks.

## 2.3 Scheduling Algorithms

To achieve high system throughput, to increase spatial reuse and to provide fair access for the subscriber stations, scheduling plays an important role. In this section we will see some of the scheduling algorithms.

```

1:  $t \leftarrow 1$ 
2: while exit any  $Y(j) > 0$  for any link  $j$  do
3:    $k \leftarrow \arg \max Y(j), \quad \forall j$  // select link  $k$ 
4:    $B \leftarrow \emptyset$  // set of blocked links in this iteration  $k$ 
5:    $A \leftarrow \emptyset$  // set of selected active links in this iteration  $k$ 
6:   while  $k \neq \emptyset$  do
7:     Add  $k$  to  $A$ 
8:     Add  $Blocked\_Neighbor(k)$  to  $B$ 
9:      $k \leftarrow \arg \max Y(j), \quad j \in A \cup B, Y(j) > 0$ 
10:  end while
11:   $ActiveLinks(t) \leftarrow A$ 
12:   $t \leftarrow (t + 1)$ 
13:   $Y(j) \leftarrow (Y(j) - 1), \quad \forall j \in A$ 
14: end while

```

Figure 2.5: Interference aware scheduling algorithm (source: [5])

## Interference-Aware Scheduling

On the basis of interference aware routing described in above section, [5] propose an interference aware scheduling. The design goal of interference aware scheduling is to exploit concurrent transmission to achieve high system throughput. Let  $D(k)$  denotes the capacity request of an SS node from  $k$ .  $D(k)$  can also be represented in terms of  $Y(j)$  for every link  $j$ . In each allocation iteration  $t$ , the scheduling algorithm determines set of active links. Then the link with highest unallocated traffic demand is selected for next allocation of unit traffic. The interfering links are excluded. The iterative allocation continues until there is no unallocated capacity request. The algorithm is shown in Figure 2.5.

The algorithm is compared to the basic scheduling described in the IEEE 802.16 standards. They have also compared it with the theoretical upper bound obtained by linear programming. In chain topology, the throughput achieved outperforms basic scheme and it approaches the upper bound. While in case of random mesh topology, the throughput is better than basic scheme but it is less than the upper bound obtained by linear programming. The algorithm leads to better spatial reuse and thus higher spectral efficiency.

## Centralized scheduling

[2] presents a scheduling algorithm which provides per flow QoS guarantees to real-time and interactive data applications by utilizing the network resources efficiently. Authors propose scheduling algorithm for real-time and non-real-time (data) traffic. Real-time application uses UDP where as data traffic uses TCP. UDP traffic is considered as high priority traffic over TCP traffic since it deal with real time applications such as VoIP, Video conferencing etc.

The total number of slots allotted for TCP and UDP traffic is calculated as follows: Let  $n_U(i)$  denote the total number of slots for node  $i$  to meet QoS requirements of all the UDP flows passing through node  $i$  respectively. Let  $\lambda_U(i)$  and  $\lambda_T(i)$  denote the average throughput required by all UDP (CBR and VBR) and TCP flows from node  $i$  respectively. Then the average throughput required is  $\lambda(i) = \lambda_U(i) + \lambda_T(i)$ . Now by using fixed allocation scheme<sup>1</sup>, calculate the number of slots  $n_T(i)$  required to provide a mean throughput of  $\lambda(i)$  to node  $i$ . Hence finally total number of slots required for node  $i$  per frame is calculated as:

$$n(i) = \max(n_U(i), n_T(i)) \quad (2.4)$$

Since UDP has higher priority this will ensure QoS to all UDP connections while TCP connection will get average throughput. Also the wastage of bandwidth is minimized. In channel adaptive fixed allocation scheme, in order to satisfy the requirement of the UDP flows,  $n_U(i)$  slots have to be allocated to the node  $i$  in every frame. Depending on the channel condition, the allocation of other  $(n(i) - n_U(i))$  slots can be defer.

## Fair and efficient uplink scheduling

In [2], the authors have proposed centralized scheduling and routing tree construction algorithms to provide per flow QoS in WiMax mesh networks. However, they assume that there is no spatial reuse, that is, only one of the links in the entire mesh network can be active in a minislot. However, in a multi-hop mesh network, allowing spatial reuse, i.e., allowing concurrent transmissions on links that are not interfering with each other is very important to achieve high spectral efficiency and system throughput.

In [4], authors have presented an efficient fair scheduling algorithm for IEEE 802.16 multi-hop mesh networks according to a new fairness model in which the bandwidth allocation is contingent on the actual traffic demands in such a way that the capacity region is not sacrificed

---

<sup>1</sup>Details of this scheme is presented in an actual paper

by imposing the fairness constraints. They formulated the scheduling problem to maximize the system throughput subject to the fairness notion. By exploiting the characteristics of WiMax mesh networks, they developed an efficient algorithm to find the optimal schedule corresponding to the optimum of the formulated problem<sup>2</sup>.

## 2.4 Call Admission Control

Call admission control is another important module to provide QoS guarantees. The main goal behind designing of CAC is to accept or reject the flow depending on its required QoS parameters. If all the required parameters are satisfied, CAC module admits the flow and adds an entry in a database. There are very less literatures available in area of Call admission control in mesh network. Here I present one call admission control proposed in [3].

The CAC presented in [3], first constructs appropriate tree-based topology connecting wireless to the wired gateway and then admits the best possible subsets of connection while respecting their required QoS guarantees. Then it tries to admit all the connections with certain degradation of their QoS requirements. To admit the best set of connections following two conditions are enforced for rate and delay respectively.

$$F(e) = \sum_{i,z} r_i^z * flow_i^z(e) * s_i \quad \forall v \quad \dots \text{Rate Constraint} \quad (2.5)$$

$$Delay \leq D(\sigma_i, \rho_i, K_i) \quad \forall i, z \quad \dots \text{Delay Constraint} \quad (2.6)$$

Where,

$v$  denotes the connection,

$F(e)$  is the total bit rate of all selected connections on link  $e$ ,

$r_i^z$  is the connection rate,

$flow_i^z(e)$  is a binary variable indicating whether or not  $e$  carries the traffic for connection  $i$  in direction  $z$ ,

$s_i$  is a binary variable indicating whether connection  $i$  is selected or not,

$D(\sigma_i, \rho_i, K_i)$  is a function of burst, rate and hope count respectively

---

<sup>2</sup>For detailed network and fairness model refer actual paper

# Chapter 3

## Work Done/Implementation

### 3.1 Intel Pro/Wireless 5116 Broadband Interface

The Intel PRO/Wireless 5116 is a highly integrated, IEEE 802.16d compliant system on chip (SoC) for both licensed and license-exempt radio frequencies. It delivers a solid foundation for the development of cost-effective customer premise equipment (CPE). Figure 3.1 and 3.2 shows the high-level block diagram and CPE system diagram respectively. This SoC is built around a high-performance OFDM modem. It supports wide range of applications and even we can change the channel bandwidths, datarate using programmable interface. It also supports both TDD and H/FDD duplexing modes and has various adaptive modulation techniques.

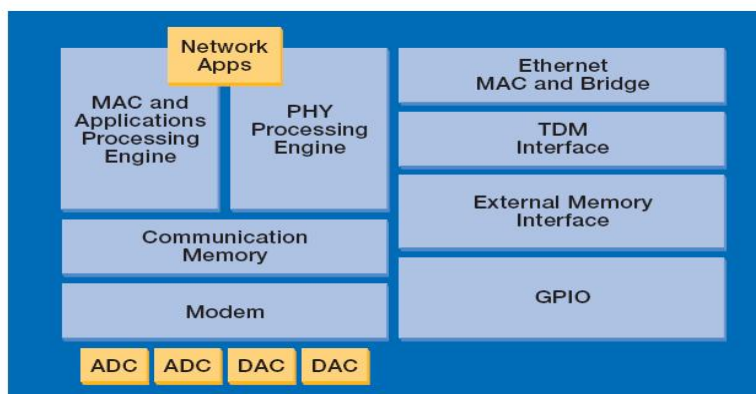


Figure 3.1: Intel Pro/Wireless 5116 high-level block diagram (source: [14])

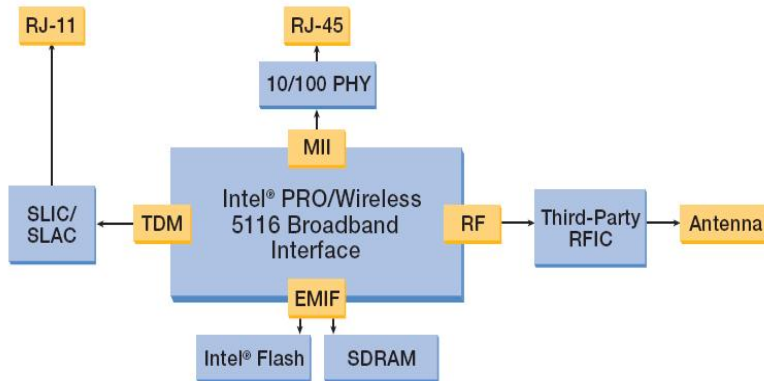


Figure 3.2: Intel Pro/Wireless 5116 CPE system diagram (source: [14])

## 3.2 Implementation

At the starting phase I went through whole IEEE 802.16-2004 spec and noted down all the possible activities or functions which MAC suppose to support. Then depending on the requirement, I classified all activities in “Must” and “Can ignore” categories after discussion. Here we put mesh and AAS functionality in optional part, since main goal is to introduce PMP and QoS on the WiMax board. I also prioritized the functions to be implemented and listed them according to priority. Then I started to work on the CAC and dynamic service functionalities which is explained in next section. Before dividing the work, I went through whole code base and marked all the functions which are already implemented or which need to be modified.

### Designing Call Admission Control

Call admission control is a vital module in supporting the QoS guarantees in WiMax. Any new connection of UGS/rtPS/nrtPS type has to go through the CAC module to see whether the required QoS guarantees will meet or not. Depending on the available bandwidth and constraints, CAC module accept or reject the incoming flow. Here I designed the basic bandwidth stealing CAC which assume constant delay for all the flows and accept or reject the flow depending on the available bandwidth (or slots). The CAC algorithm is as follows:

## Assumption

1. Each SS has been allotted a maximum number of slots for all its connection to provide fairness among the subscriber stations
2. Each flow within the SS has been assigned available (avl) number of slots (enforced some limit)
3. Each flow can steal bandwidth (or slots) to its maximum limited defined by  $U_{max}$  and  $R_{max}$  for UGS and rtPS respectively
4.  $U_{cur}$ ,  $R_{cur}$  stores currently assigned slots for UGS and rtPS respectively.
5. A flow starts stealing bandwidth (if required) from lowest flow to highest flow.
6. A flow cannot steal bandwidth/slots from its higher priority flow.

## Algorithm

Figure 3.3 shows the proposed bandwidth stealing call admission control procedure based on above mentioned assumptions.

In designing the CAC, we need to store and maintain all accepted flow's information so that to provide input to the scheduler. Scheduler then uses this database to generate the UL and DL maps. We also need to keep information of all flows and SS for taking decision of any new incoming flow. Data structures *tConnectionInfo* and *tCacData* will be used respectively for above purposes. *tConnectionInfo* is a linked list containing all accepted flows where as *tCacData* is a linked list containing all CAC data for the SS. Each node of *tCacData* is corresponds to a SS.

Various functions and data structures implemented for bandwidth stealing CAC are included in Appendix A. It also includes the flowchart depicting the CAC algorithm.

## Implementing Dynamic Services on WiMax board in PMP mode

Dynamic service includes Dynamic Service Addition (DSA), Dynamic Service Change (DSC) and Dynamic Service Deletion (DSD). To provide quality of service, we need to check whether the required QoS guarantees can be satisfy by BS or not. Hence whenever any new flow arrives,

```

1: Calculate the required number of slots for new flow
2: if SS limit exceed // If accepted then
3:   Reject and return
4: else
5:   if Is required slots available then
6:     Accept and return
7:   else
8:     if Within flow's MAX limit then
9:       if Is slots available in lower flows then
10:        Steal required slots from lower flow
11:        Accept and return
12:       else
13:        Check for next lower flow by repeating step 9, if flow's priority is less than itself
14:       end if
15:     else
16:       Reject and return
17:     end if
18:   end if
19: end if

```

Figure 3.3: Proposed Call Admission Control

want to change existing QoS parameters or want to delete the existing flow, we need to send the DSA/DSC/DSD request to the CAC module on BS. CAC analyzes the connection and sends DSA/DSC/DSD response back.

All above messages have their own format as specified in IEEE 802.16-2004 specification. I designed all these messages in their required form, parsed the request/response and prepare them to send in next frame. Here for simplicity we assume certain fixed QoS parameters hence no need to send in a DSA/C request. In DSA/C/D we only need to send priority (or flow type), to take the decision. In response, BS sends the data cid back to the SS to use in further transactions. All the request/response/ack transactions are identified by unique transaction ID generated by sender.

Various data structures used to store and queue the request, constants and functions prototype required to implement DSx functionality is included in the Appendix B.

# Chapter 4

## Problem Definition

### 4.1 Designing Scheduling and/or Call Admission Control

Scheduling and call admission control are important module in providing QoS guarantee. IEEE 802.16 standard does not specify any standard to design and implement these in mesh networks. There are lot of work done in the area of P2P and PMP but there are very few literatures available in mesh topology since it is a newer technology and also designing algorithms for multi hop network are considered to be complex.

Here I am planning to design efficient scheduling and/or call admission control algorithms in mesh networks. The main goal behind designing a scheduling algorithm is that it should accommodate more number of connections per frame. As routing is closely coupled with scheduling, for designing scheduling we need to have some routing mechanism as well.

### 4.2 Proposed Plan

1. Since routing, scheduling and call admission control are closely coupled with each other, during II stage of the MTP, I will design scheduling on top of the routing algorithm in mesh network. Here I am planning to come up with the design of the scheduling algorithm or some scheduling model.
2. In stage III, if hardware available, I will try to implement the proposed scheduling algorithm on actual hardware. I also try to validate the model using some simulation or analytic model. Then I will present the results of performance evaluation of the algorithm proposed.

# Chapter 5

## Conclusion and Future Work

Here I have presented various routing, link scheduling, and admission control algorithms proposed in various literatures. But as compare to point-to-multipoint there are very few work done in area of providing QoS guarantees in mesh networks. For providing required QoS guarantees we need to have efficient routing, scheduling and call admission control policy. Also designing algorithm for multi hop networks is considered to be complex as packets can route through various intermediate nodes.

It would be interesting and will be a novel idea to implement the routing, scheduling and call admission control which is able to provide required QoS guarantees and at the same time it will be efficient. Here efficient mean the algorithm is able to accommodate more number of connections per frame.

# Bibliography

- [1] IEEE. “*IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*”, IEEE standard, October 2004.
- [2] H. Shetiya and V. Sharma, “*Algorithms for Routing and Centralized Scheduling to Provide QoS in IEEE 802.16 Mesh Networks*”, in WMuNeP '05: Proceedings of the 1<sup>st</sup> ACM workshop on Wireless multimedia networking and performance modeling, New York, NY, USA, 2005, pp. 140-149, ACM Press.
- [3] Seungjoon Lee, Girija Narlikar, Martin Pal, Gordon Wilfong, Lisa Zhang., “*Admission Control for Multihop Wireless Backhaul Networks with QoS Support*”, IEEE WCNC 2006, Las Vegas NV, April 2006
- [4] M. Cao, V. Raghunathan, and P. R. Kumar., “*A tractable algorithm for fair and efficient uplink scheduling of multi-hop wimax mesh networks*”, In Proceedings of 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh 2006), September 2006.
- [5] H. Wei, S. Ganguly, A. Izmailov, and Z. Haas, “*Interference-Aware IEEE 802.16 WiMax Mesh Networks*”, in Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61<sup>st</sup>, Vol. 5; 3102-3106.
- [6] F. Jin, A. Arora, J. Hwang, and H.-A. Choi, “*Routing and Packet Scheduling for Throughput Maximization in IEEE 802.16 Mesh Networks*”, submitted for publication.
- [7] J. Tao, F. Liu, Z. Zeng, and Z. Lin, “*Throughput Enhancement in WiMax Mesh Networks Using Concurrent Transmission*”, in International Conference on Wireless Communications, Networking and Mobile Computing, Sep 2005, pp. 871-874.
- [8] IEEE P802.16-REVd/D5-2004. “*Air Interface for Fixed Broadband Wireless Access Systems*”

- [9] I. F. Akyildiz, X. Wang, and W. Wang, “*Wireless Mesh Network: A Survey*”, January 2005.
- [10] Y. Yuan, H. Yang, S. Wong, S. Lu, W. Arbaugh, “*ROMER: Resilient Opportunistic Mesh Routing for Wireless Mesh Networks*”, in Proceeding of IEEE Workshop on Wireless Mesh Networks (WiMesh), 2005.
- [11] R. Draves, J. Padhye, and B. Zill, “*Routing in Multi-radio, Multihop Wireless Mesh Networks*”, in ACM MobiCom, Philadelphia, PA, September 2004.
- [12] Chandra and A. Sahoo. “*An Efficient Call Admission Control for IEEE 802.16 Networks*”, Technical Report, Available: <http://www.it.iitb.ac.in/research/techreport/reports/30.pdf>
- [13] Wang H., Li W. and Agrawal D.P. “*Dynamic admission control and QoS for 802.16 wireless MAN*”, Wireless Telecommunications Symposium, 2005, vol. no.pp. 60-66, 6-7 April 2005.
- [14] Intel Corporation. “*Intel Pro/Wireless 5116 Broadband Interface*”, Available: <http://www.intel.com/network/connectivity/products/wireless/307327.pdf>

# Appendix A

## Call Admission Control Prototypes

### Constants

```
#define MAX_UGS_SLOTS      45
#define MAX_RTPS_SLOTS    40
#define MAX_BE_SLOTS      40
#define AVAIL_UGS_SLOTS   35
#define AVAIL_RTPS_SLOTS  30
#define AVAIL_BE_SLOTS    35
```

### Data Structures

```
typedef struct sCacData {
    struct sCacData *next;
    UINT8 ssid;
    UINT8 SSmax; /* Maximum number of slots permitted to the SS */
    UINT8 Uavail, Ucur, Umax, Umin;
    UINT8 Ravail, Rcur, Rmax, Rmin;
    UINT8 Bavail, Bcur, Bmax, Bmin;
} tCacData;
tCacData *cacDataChain, *cacDataLast;

typedef struct sConnectionInfo {
    struct sConnectionInfo *next;
    UINT16 cid;
    UINT8 slots;
    UINT8 status;
} tConnectionInfo;
```

```
tConnectionInfo *connectionChain,  
*connectionLast;
```

```
typedef struct sConnAlive {  
    UINT8 UGS;  
    UINT8 rtPS;  
    UINT8 BE;  
} tConnAlive;  
tConnAlive isConnAlive;
```

## Functions

```
static int uCpsAddConnection(UINT16 cid, UINT8 slots, UINT8 status);  
static int uCpsChangeConnection(UINT16 cid, UINT8 slots, UINT8 status);  
static int uCpsDeleteConnection(UINT16 cid);  
static int uCpsSetConnectionStatus(UINT16 cid, UINT8 status);  
static int uCpsCacInit(void);  
static int uCpsCallAdmissionControl(UINT8 ssid, eCidType flow, UINT8 req);  
static int uCpsModifyCacData(UINT8 ssid, eCidType flow, UINT8 del);  
static UINT8 uCpsGetConnectionStatus(UINT16 cid);  
static void uCpsSetCacData(tCacData *cacData, UINT8 ssid, float SSmax, UINT8 Uavail,  
UINT8 Umax, UINT8 Ravail, UINT8 Rmax, UINT8 Bavail, UINT8 Bmax);  
static tCacData* uCpsGetCacData(UINT8 ssid);  
static tConnectionInfo* uCpsSearchConnection(UINT16 cid);
```

## Call Admission Control Flow Chart

Figure A.1 shows the flow chart of proposed bandwidth stealing call admission control.

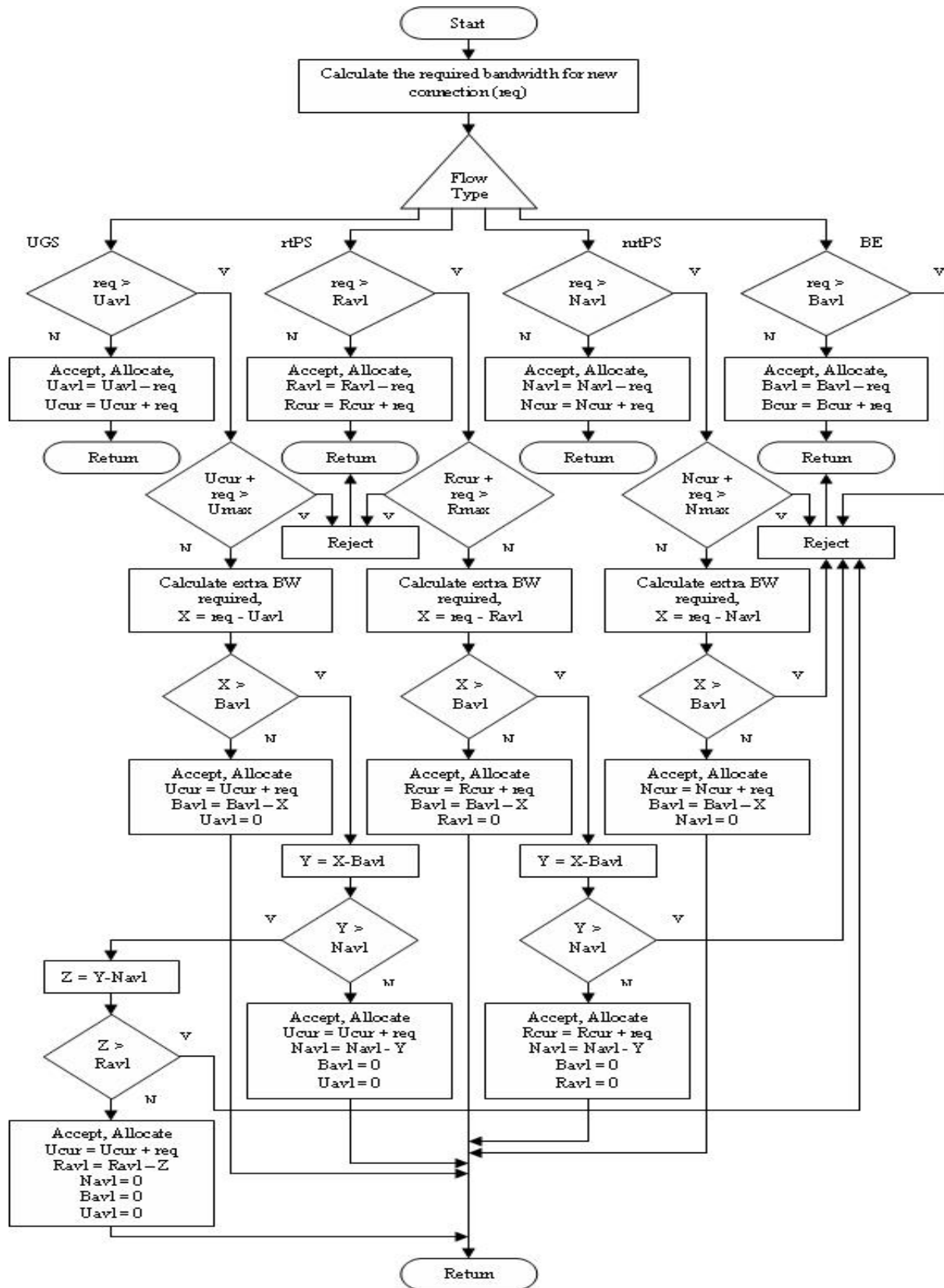


Figure A.1: Proposed Call Admission Control flow chart

# Appendix B

## Dynamic Service Related Prototypes

### Constants and Variables

```
#define DSA_REQ_MSG_TYPE      (11)
#define DSA_RSP_MSG_TYPE     (12)
#define DSA_ACK_MSG_TYPE     (13)
#define DSC_REQ_MSG_TYPE     (14)
#define DSC_RSP_MSG_TYPE     (15)
#define DSC_ACK_MSG_TYPE     (16)
#define DSD_REQ_MSG_TYPE     (17)
#define DSD_RSP_MSG_TYPE     (18)

#define DSX_MSG_LENGTH        (1)
#define DSX_TRANSACTION_ID_LENGTH (2)
#define DSX_CONFIRM_CODE_LENGTH (1)
#define SF_CID_TYPE           (2)
#define SF_CID_LENGTH         (2)
#define SF_PRIORITY_TYPE      (6)
#define SF_PRIORITY_LENGTH    (1)

#define DSX_REQ_TLV_SIZE_COMMON (SF_PRIORITY_LENGTH + 2)
#define DSX_RSP_TLV_SIZE_COMMON (SF_CID_LENGTH + 2)
#define DSX_REQ_TLV_SIZE DSX_REQ_TLV_SIZE_COMMON
#define DSX_RSP_TLV_SIZE DSX_RSP_TLV_SIZE_COMMON
#define DSX_REQ_FIXED_IN_DATA (DSX_MSG_LENGTH + \
                               DSX_TRANSACTION_ID_LENGTH)
#define DSX_RSP_FIXED_IN_DATA (DSX_MSG_LENGTH + \
                               DSX_TRANSACTION_ID_LENGTH + \
```

```

DSX_CONFIRM_CODE_LENGTH)
#define DSX_REQ_SIZE (DSX_REQ_FIXED_IN_DATA + DSX_REQ_TLV_SIZE)
#define DSX_RSP_SIZE (DSX_RSP_FIXED_IN_DATA + DSX_RSP_TLV_SIZE)
#define DSX_ACK_SIZE (DSX_RSP_FIXED_IN_DATA)

```

```

UINT16 transactionId;

```

## Data Structures

```

typedef struct _tDsxReq {
    char msgType;
    UINT16 transactionId;
    char tlvCode[1];
} tDsxReq;

```

```

typedef struct _tDsxRsp {
    char msgType;
    UINT16 transactionId;
    char confirmationCode;
    char tlvCode[1];
} tDsxRsp;

```

```

typedef struct _tDsxAck {
    char msgType;
    UINT16 transactionId;
    char confirmationCode;
} tDsxAck;

```

```

typedef struct sDsxReqData {
    struct sDsxReqData *next;
    char msgType;
    UINT16 transactionId;
    char priority;
} tDsxReqData;

```

```

typedef struct sDsxAckData {
    struct sDsxAckData *next;
    char msgType;
    UINT16 transactionId;
    char confirmationCode;
} tDsxAckData;

```

```

} tDsxAckData;

static tDsxReqData *dsxReqChain, *dsxReqLast, *dsxReqWaiting;
static tDsxAckData *dsxAckChain, *dsxAckLast;
tMpiDataSegDescr *uCpsDsxGetRequest(void);
tMpiDataSegDescr *uCpsDsxGetAck(void);

typedef struct sDsxRspData {
    struct sDsxRspData *next;
    char msgType;
    UINT16 transactionId;
    char confirmationCode;
    UINT16 cid;
} tDsxRspData;

static tDsxRspData *dsxRspChain, *dsxRspLast, *dsxRspWaiting;
typedef enum confirmationCode    eConfirmationCode;

```

## Functions

```

UINT16 uCpsGenerateNewTransactionId(void);
static int uCpsHandleDsxMgmtPdu(tMpiDataSegDescr* dseg, UINT16 cid);

```