Bandwidth Allocation for Quality of Service Provision in IEEE 802.16 Systems

Tze Wei Tang

Thesis submitted for the degree of

Doctor of Philosophy

in

Electrical and Electronic Engineering

at

The University of Adelaide

(Faculty of Engineering, Computer and Mathematical Sciences)

School of Electrical and Electronic Engineering

March 3, 2009

Chapter 6

Priority-Based Dual-Queue Scheduler

In this chapter, we propose the Priority-Based Dual-Queue (PBDQ) scheduler, which is made up of multiple DQs differentiated by connections' priority class. That is, each priority class is handled in a separate DQ. The PBDQ ensures connections that belong to a higher priority class are served ahead of connections that belong to lower priority classes at all times, in systems where connections may change their priority class.

6.1 Motivation

In Chapter 3, we discussed different objectives that can be applied to an 802.16 network, based on system parameters, such as throughput or revenue. In Chapter 5, we considered the primary objective of the network to be maximisation of the number of connections that experience "good service". In this chapter, we contend, however, that the proposal of Chapter 5 can not deliver other objectives of a network effectively, such as maximisation of throughput and revenue.

Let us consider an example where the primary objective of an 802.16 network is to maximise throughput. We assume each connection in the network has the same offered load and the network is congested. In order to maximise the throughput of the network, the DQ scheduler must ensure that no connections at the β-queue have higher PHY mode than any connections at the α-queue. However, the DQ scheduler proposed in Chapter 5 would fail to achieve this condition if the PHY mode of connections at the α-queue and the β-queue is allowed to change during operation, which is the case in a real 802.16 network. As a result, the objective of maximising throughput of the network may be violated. Therefore, we propose a priority-based DQ structure to handle this problem.

The system parameters considered in the objectives of an 802.16 network can be used to identify the priority class of connections in the network. In order to achieve the objectives of the network with different priority classes of connections at all times, we extend the DQ scheduler from a single DQ to a multiple DQ structure, which we call the Priority-Based DQ scheduler. The proposed PBDQ scheduler is able to achieve the objectives of the network even when there are changes in the priority class of any connections in the network during operation.

For the example above, a connection's priority class can be based on its PHY mode because a higher throughput is achieved by the network when connections in the network transmit at a higher PHY mode. Hence, a connection's priority class changes whenever its PHY mode changes. Without a PBDQ scheduler, the objective of a DQ scheduler may be totally violated in the worst-case scenario where all the connections in the α-queue change to a lower PHY mode, whilst connections at the β-queue change to a higher PHY mode. Therefore, it may be important for a service provider to be able to choose to implement a PBDQ scheduler, even though it may be more complex.

6.2 The PBDQ structure

Based on the priority class of connections, we allocate a separate DQ structure for each priority class. The PBDQ scheduler is an extension of the DQ proposed in

Chapter 5, giving each priority class its own complete DQ, including α-queue and β-queue, its own QoS Violation Detection and QoS Recovery Detection, and its own connection selection criteria. Figure 6.2.1 shows a PBDQ structure with three priority classes (high, medium and low).

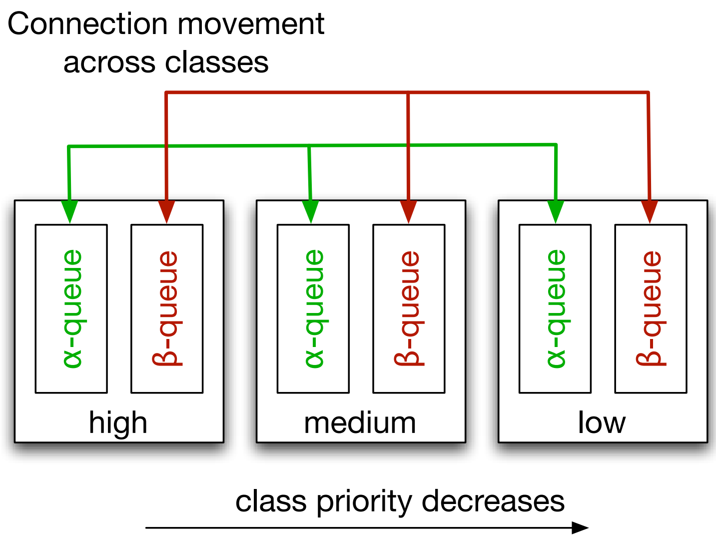

Figure 6.2.1: The structure of a Priority-Based DQ Scheduler.

When a packet arrives at the network, it is sent to one of the DQs based on its priority class. Within a priority class, the packet is sent to either the α-queue or the β-queue. In order to respect the priority classes of the system, a higher priority class is always served ahead of the lower priority classes. If there is no available bandwidth left after serving a higher priority class, the lower priority classes will not be served. Within a priority class, the α-queue is served ahead of the β-queue, which is an inherent property of the DQ scheduler.

6.3 The PBDQ mechanism

Under uncongested conditions, only the α-queue of each of the priority classes is occupied and served by the PBDQ scheduler. When a QoS Violation event is detected by the QoS Violation Detection at a priority class, a connection at its α-queue will be sent to its β-queue. Due to the service sequence of the scheduler, a QoS Violation event is expected to be detected earlier at the lowest priority class. Similarly, when the QoS Recovery Detection of a priority class indicates its α-queue is able to accommodate another connection, a connection from its β-queue will be redirected back to its α-queue.

Let us consider an example with two priority classes: a low priority class and a high priority class. Initially, the network is not congested and hence some connections are being served at the α-queue of the low and high priority classes respectively. Now, assume that the available bandwidth is only sufficient to serve the high priority class. Hence, the low priority class will not be served. As a result, a QoS Violation event will be detected at the low priority class and a single connection at the α-queue will be moved to the β-queue at a time. Eventually, all connections of the low priority class at the α-queue will be moved to the β-queue.

Within the high priority class, if the available bandwidth is not sufficient to serve all connections at the α-queue, a QoS Violation event will be detected and one single connection will be moved to the β-queue. When more bandwidth becomes available in the network, the β-queue of the high priority class will be served ahead of the β-queue of the low priority class.

Due to the priority-based structure of the PBDQ scheduler, a connection can be easily moved to a different priority class whenever there is a priority class change. The next problem is to decide to which queue we move the connection that has just changed priority class. To solve this problem, let us consider all the possible priority class changes for a connection in the network.

- A connection at the α-queue changes from a high to a low priority class: This connection was very likely being fully served at the α-queue before it changes its priority class as it belongs to the α-queue of the high priority class. Hence, we choose to send it to the α-queue of its new lower priority class because it should be treated the same as those connections in the α-queue of its new priority class.

- A connection at the β-queue changes from a high to a low priority class: This connection may be partially or not served at the β-queue before it changes its priority class, which indicates that the lower priority classes are not served at all. Hence, this connection can be either sent to the α-queue or the β-queue of its new priority class. If we chose to send this connection to the α-queue of its new priority class, we would expect it to be moved to the β-queue eventually. This is because connections in the β-queue of the high priority class will be served ahead of those in the α-queue of the low priority class. Therefore, we choose to send this connection to the β-queue of its new priority class.

- A connection at the α-queue changes from a low to a high priority class: In general, if there are any connections at the α-queue of a low priority class, it indicates that there are no connections at the β-queue of the high priority class. Hence, this connection can be either sent to the α-queue or the β-queue of its new priority class. We choose to send this connection to the α-queue because this connection was very likely being served before it changes its priority class, and so, it should be treated the same as those connections in the α-queue of its new priority class.

- A connection at the β-queue changes from a low to a high priority class: This connection is very unlikely to be served at the β-queue of the low priority class. Hence, this connection should be served only if there is left over bandwidth after serving the connections at the α-queue of the high priority class. Therefore, we choose to send this connection to the β-queue of its new priority class.

Based on the above, we contend that a connection at the β-queue should be moved to the β-queue of its new priority class in order to preserve the inherent priority for connections at the α-queue of the new priority class. Similarly, for a connection at the α-queue that has changed it is proposed that it be moved to the α-queue of its new priority class. Therefore, we apply an α-to-α or a β-to-β connection movement whenever there are any changes in a connection's priority class.

In fact, there are only two out of the four events described above that may impact the objective of maximising MAC throughput achieved in a network: a connection at the α-queue changes its PHY mode to a lower PHY rate and a connection at the β-queue changes its PHY mode to a higher PHY rate.

6.4 Experiment for the PBDQ scheduler

We examine two schemes in this experiment, which aim to maximise MAC throughput achieved in a network.

- Scheme 1 (Benchmark): We carry out an experiment, which is based on the DQ scheduler described in Chapter 5. This scheme uses the same parameters presented in Table 5.5.2 of Chapter 5. Note that this scheme is used as a benchmark to compare the MAC throughput achieved between this scheme and the scheme employing a PBDQ scheduler.

- Scheme 2 (PBDQ): The objectives of this system are described in Table 6.4.1. Primarily, we seek to maximise the MAC throughput of the 802.16 system using a PBDQ scheduler. We assign a different priority class for each PHY mode supported in the network, where the highest PHY mode corresponds to the highest priority class, and similarly for all other PHY modes. Hence, a higher PHY mode connection will be served ahead of the lower PHY mode connections. Whenever there is a change in PHY mode of a connection, this connection will be moved to the priority class that corresponds to the new PHY mode. Further, an α-to-α and a β-to-β rule applies to all connection movements between the priority classes.

Table 6.4.1: The objectives defined for the system and the connection prioritisation mechanism for the PBDQ experiment.

Objectives	1. Maximising MAC throughput. 2. Maximising the number of connections with good service.
Connection Prioritisation	First, based on PHY mode, where the highest PHY mode corresponds to the highest priority. Then, based on index, where the highest index corresponds to the highest priority.

We consider a network with four SSs connected to a BS. Initially, all SSs have the same PHY mode (PHY rate = 15 $Mbps$). Each SS carries a single DL rtPS connection generated by a CBR traffic source at a rate of 2.3 $Mbps$. The packet size of the CBR traffic source is 100 B (including 20 B of IP header).

We consider the two events that may impact on the MAC throughput achieved in schemes 1 and 2: a connection at the α-queue changes its PHY mode to a lower PHY rate and a connection at the β-queue changes its PHY mode to a higher PHY rate. In order to differentiate between overlapping lines, the PHY mode of each connection is plotted with a different resolution in Figure 6.4.1 (a).

6.4.1 Experimental Regions

Similar to the experiments described in Chapter 5, this experiment is carried out for 20 seconds. For discussion purposes, we divide the experiment into 6 regions, where regions 3, 4, 5 and 6 are designed to show the differences between the two schemes under observation.

- Region 1: $0 \leq t < 5$, where t is time in seconds. This region is reserved for the initialisation process. At $t = 3$, all connections are started with the same offered load of 0.1 $Mbps$.

- Region 2: $5 \leq t < 6$. At $t = 5$, the offered load of each connection is increased to 2.3 $Mbps$.

- Region 3: $6 \leq t < 10$. At $t = 6$, connection 3 is in the α-queue and changes its PHY mode from 15 to 10 $Mbps$.

- Region 4: $10 \leq t < 12$. At $t = 10$, connection 3 changes its PHY mode back to its initial PHY rate of 15 $Mbps$.

- Region 5: $12 \leq t < 18$. At $t = 12$, connection 0 is in the β-queue and changes its PHY mode from 15 to 20 $Mbps$.

- Region 6: $18 \leq t < 20$. At $t = 18$, connection 0 changes its PHY mode back to its initial PHY rate of 15 $Mbps$

6.4.2 Experimental Results

In the following discussion, we interleave the discussion of schemes 1 and 2 in order to provide a direct comparison of their performance to the reader.

Region 1 $(0 \leq t < 5)$ – Schemes 1 and 2

All the delay, MAC throughput and QoS features in this region for schemes 1 and 2 have been discussed in the previous experiments described in Chapter 5, and the results are shown in the appropriate regions of Figures 6.4.1 to 6.4.4.

Region 2 $(5 \leq t < 6)$ – Schemes 1 and 2

Due to the limited number of slots to serve all connections at the α-queue, QoS Violation events are detected, which causes both connections 0 and 1 to be sent to the β-queue. Hence, in both schemes, both connections 2 and 3 experience good service and both connections 0 and 1 experience degraded service.

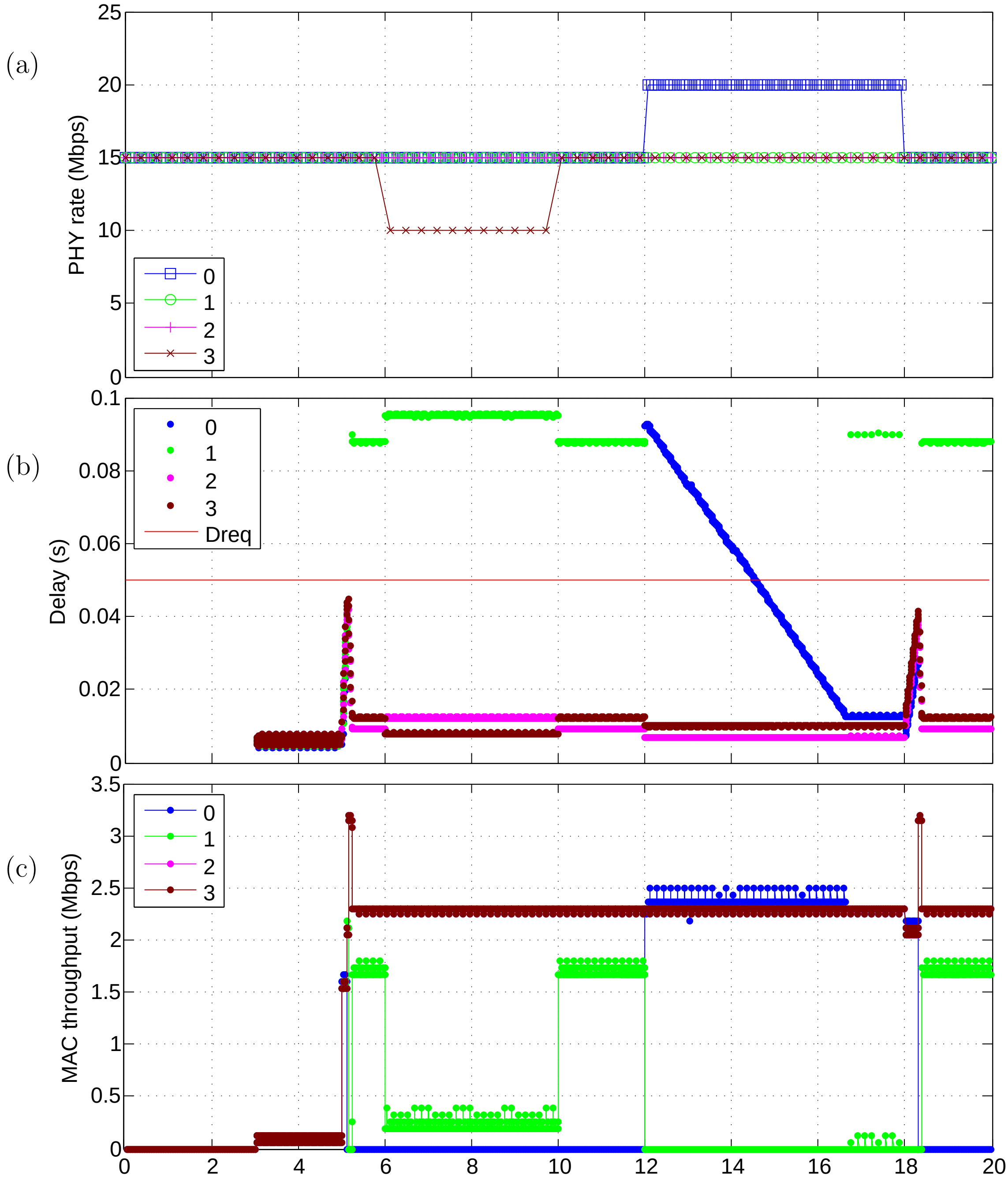

Figure 6.4.1: Scheme 1 (Benchmark): (a) PHY rate, (b) packet delay, and (c) MAC throughput.

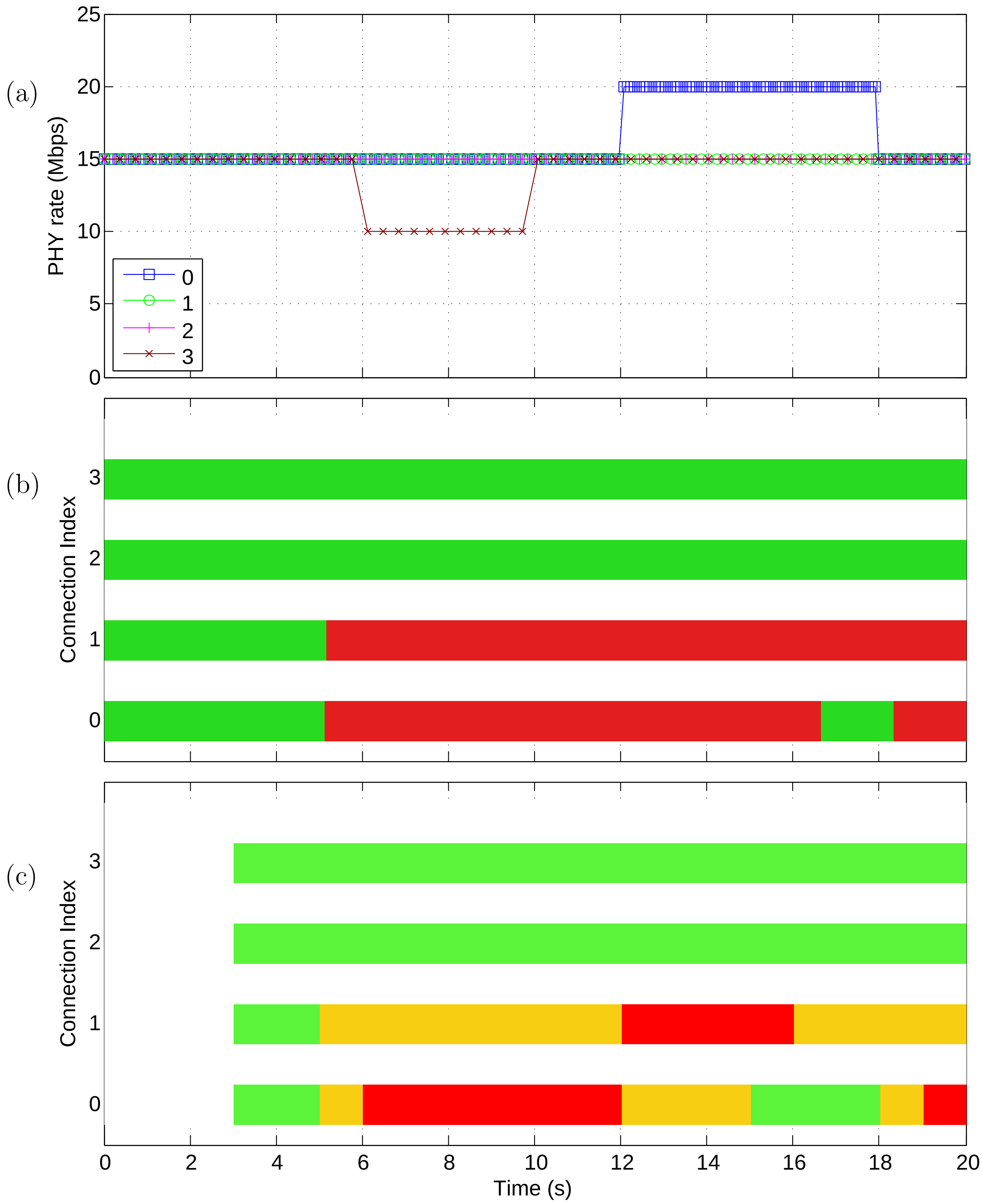

Figure 6.4.2: Scheme 1 (Benchmark): (a) PHY rate, (b) connections at each queue; α-queue (green), β-queue (red), and (c) QoS received; good service (green), degraded service (amber) and no service (red).

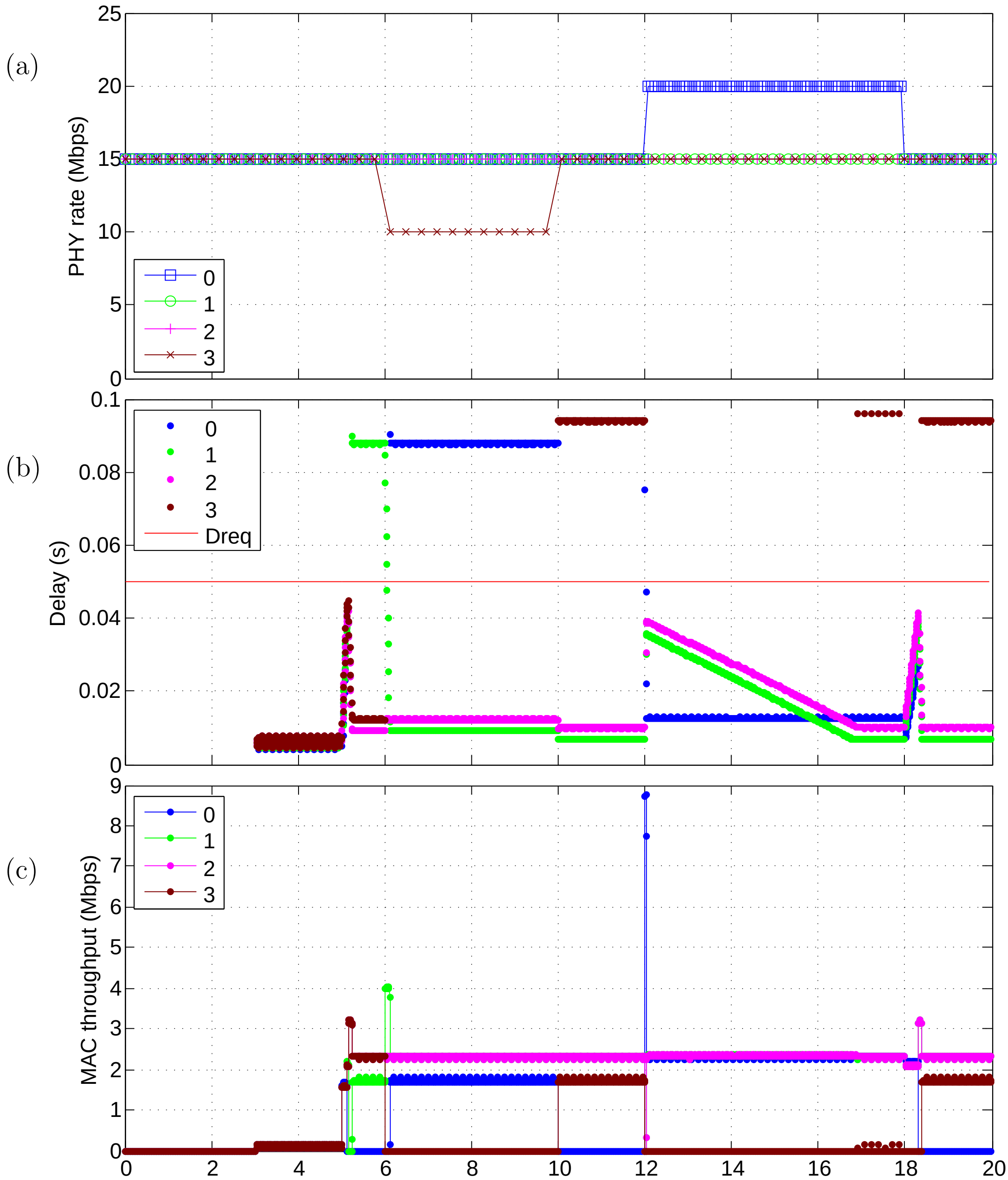

Figure 6.4.3: Scheme 2 (PBDQ): (a) PHY rate, (b) packet delay, and (c) MAC throughput.

(a)
(b)
(c)

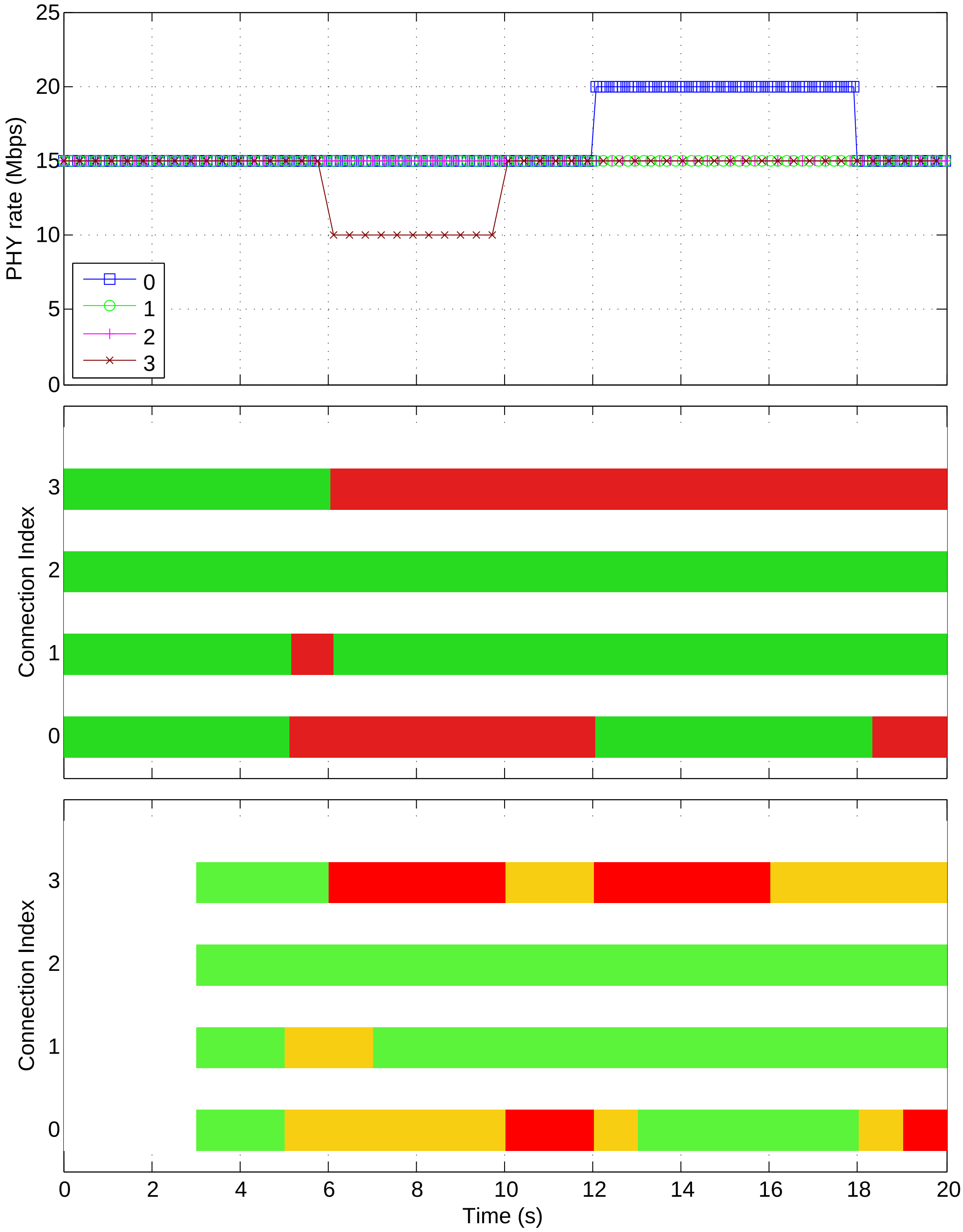

Figure 6.4.4: Scheme 2 (PBDQ): (a) PHY rate, (b) connections at each queue; α-queue (green), β-queue (red), and (c) QoS received; good service (green), degraded service (amber) and no service (red).

Connection 0 achieves no MAC throughput soon after $t = 5$ in both schemes, as can be seen in Figures 6.4.1 (c) and 6.4.3 (c), but it is considered as receiving degraded service because its QoS is determined in every second and it has transmitted some packets at the start of $t = 5$. Note that in scheme 2, all connections belong to the same priority class, since they have the same PHY mode.

Region 3 $(6 \leq t < 10)$ **– Scheme 1**

When connection 3 at the α-queue changes its PHY mode to a lower PHY mode in scheme 1, it causes a drop in the MAC throughput of connection 1, which is a connection being served at the β-queue. This is because connection 3 requires more slots and so there are fewer slots left after serving the α-queue.

From Figure 6.4.5 (b), we observe that the total MAC throughput of all connections in this region is less than the one achieved in region 2 because connection 3 is being served with a lower PHY mode. In other words, connection 3 requires more slots to achieve the same MAC throughput as connection 2.

Region 3 $(6 \leq t < 10)$ **– Scheme 2**

Under scheme 2, connection 3 is moved from the α-queue of its current priority class to the α-queue of a lower priority class because it has changed to a lower PHY mode. From the design of the PBDQ scheduler, connections at the highest priority class will be served ahead of connections at lower priority classes. Hence, connection 3 will only be served after connections at higher priority classes are served.

Due to the movement of connection 3 to a lower priority class, connection 1 at the β-queue of the priority class corresponding to the PHY rate of 15 $Mbps$ is given more slots to transmit its packets at its queue. Hence, it is redirected back to the α-queue. Then, connection 0 at the β-queue is chosen to be served using the left over slots after serving the α-queue. Therefore, connection 3 is not served at all.

From Figure 6.4.5 (b), we observe that the total MAC throughput of the network in this region is same as that achieved in region 2, since all connections being

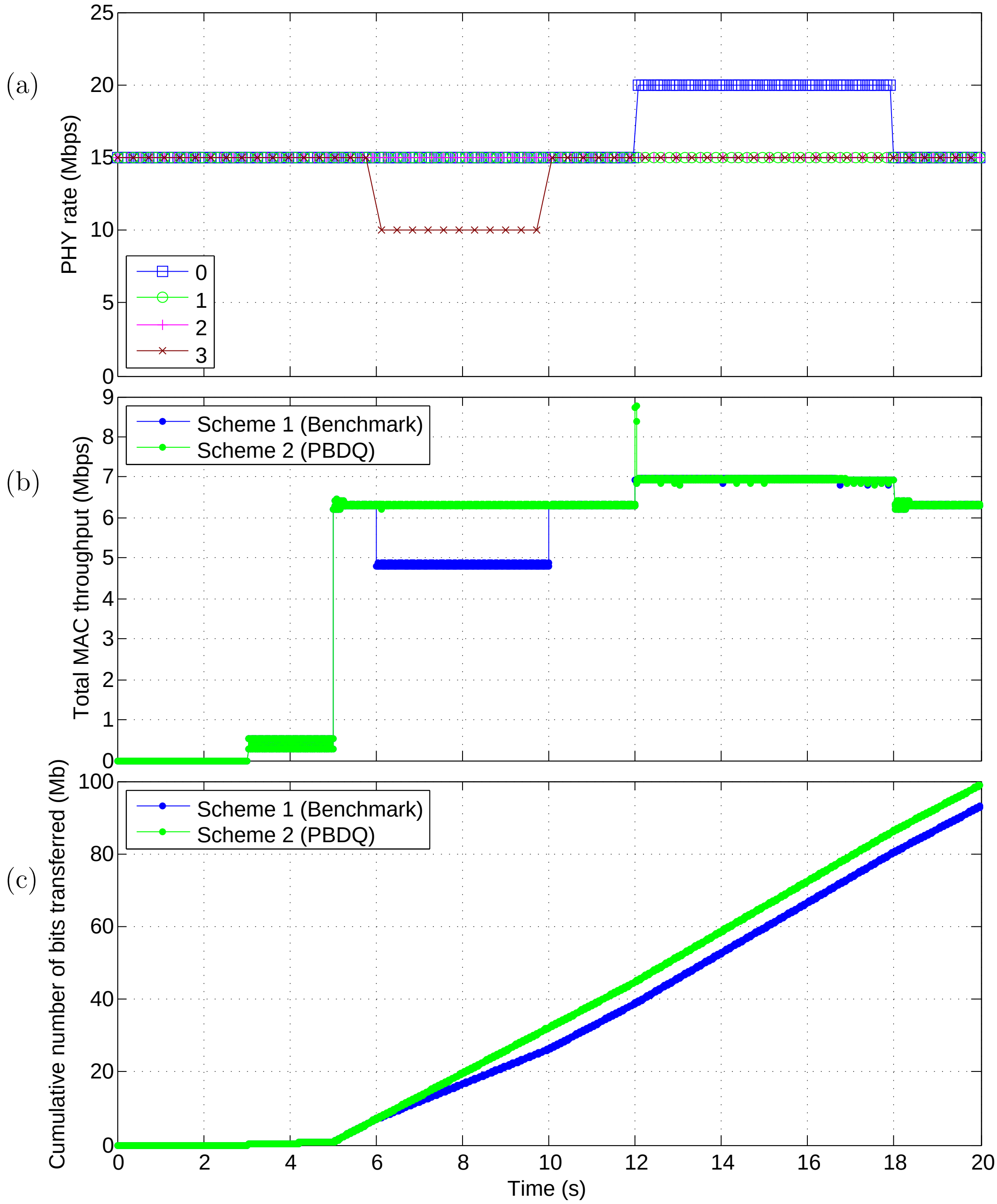

Figure 6.4.5: (a) PHY rate, (b) total MAC throughput, and (c) cumulative number of bits transferred.

served belong to the same priority class. From Figure 6.4.5 (c), we observe that the cumulative number of bits transferred for scheme 1 starts to diverge from the cumulative number of bits transferred for scheme 2, from $t = 6$ until $t = 10$.

Region 4 $(10 \leq t < 12)$ − **Scheme 1**

At $t = 10$, connection 3 changes its PHY mode back to its initial PHY rate. Hence, connection 1 is given more slots to transmit its packets and its MAC throughput increases to 1.7 $Mbps$.

Region 4 $(10 \leq t < 12)$ − **Scheme 2**

When connection 3 changes its PHY mode back to its initial PHY rate, it is moved back to the β-queue of priority class that corresponds to a PHY rate of 15 $Mbps$. At the β-queue, connection 3 is chosen to be served instead of connection 0 because it wins the tie breaker based on each connection's index.

Region 5 $(12 \leq t < 18)$ − **Scheme 1**

Before $t = 12$, all connections have the same PHY mode. Connections 2 and 3 are served at the α-queue and connections 0 and 1 are served at the β-queue. At the β-queue, connection 1 is partially served and it achieves a MAC throughput of 1.7 $Mbps$, while connection 0 is not served at all.

At $t = 12$, connection 0 changes its PHY mode to a higher rate, which allows it to be chosen and served at the β-queue since it has a higher PHY mode than connection 1. Note that connection 1 is served using only the left over bandwidth after serving connections 2 and 3. This is shown in Figure 6.4.1 (c). Due to its higher PHY mode, connection 0 is expected to achieve a MAC throughput of more than 1.7 $Mbps$, which is the MAC throughput achieved by connection 1 with a lower PHY mode. Therefore, the total MAC throughput achieved in this region is the highest among the other regions, as shown in Figure 6.4.5 (b). In order to see the total MAC throughput achieved by this scheme clearly, we plot the total MAC throughput achieved by each scheme from $t = 12$ to $t = 18$ in Figure 6.4.6.

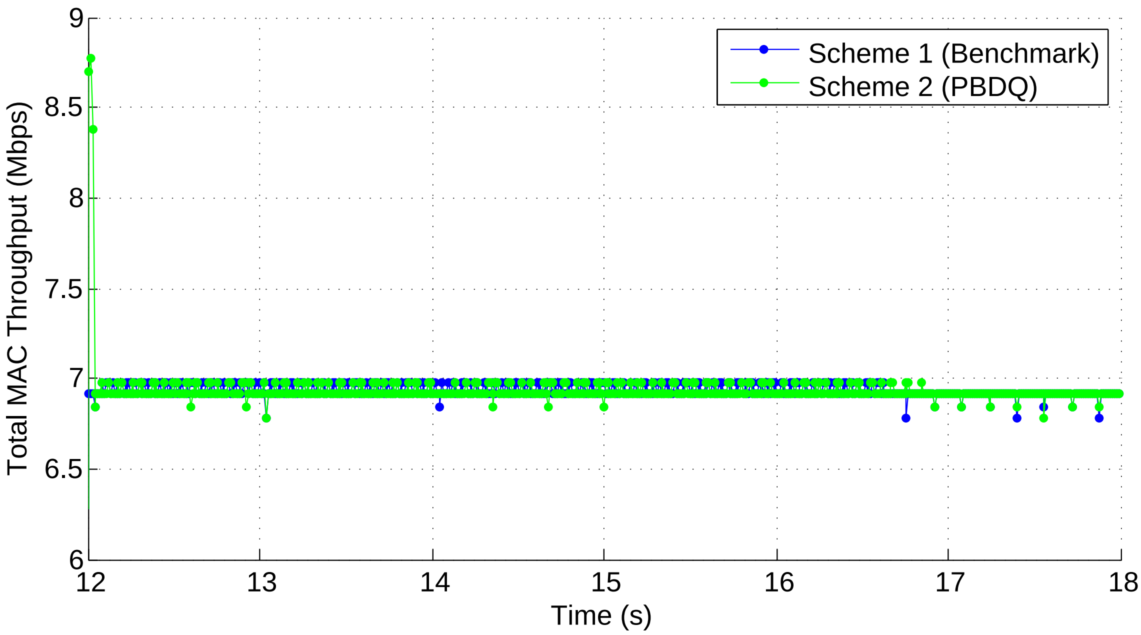

Figure 6.4.6: The total MAC throughput for both schemes during the time interval between $t = 12$ and $t = 18$.

Notice that the MAC throughput of connection 0 is slightly above its offered load, as can be seen in Figure 6.4.1 (b), due to the queued packets of connection 0 at the β-queue. It takes over 4 seconds for this backlog of packets to be cleared. While this backlog of packets for connection 0 is being cleared, the number of queued packets at the β-queue starts to decrease, and hence the delay experienced by connection 0 also starts to decrease, as shown in Figure 6.4.1 (b). These events show that there is enough bandwidth to accommodate connection 0 at the α-queue. When connection 0 experiences a delay less than D_{req} at $t = 14.4$, connection 0 is regarded as receiving good service at $t = 15$, as shown in Figure 6.4.2 (c). At $t = 16.4$, the queued packets of connection 0 have all been transmitted, and so it is redirected back to the α-queue, as shown in Figure 6.4.2 (b).

Note that the bandwidth left to serve connection 1 is very close to zero after

$t = 16.4$. Thus, it only receives service between $t = 16.4$ and $t = 18$ in a few MAC frames.

Region 5 ($12 \leq t < 18$) – Scheme 2

Under scheme 2, connection 0 is moved from the β-queue of its current priority class to the β-queue of a higher priority class because it has changed to a higher PHY mode. Unlike scheme 1 where connection 0 is served using only the left over bandwidth after serving connections 2 and 3 at the α-queue, connection 0 is served ahead of the other connections at the lower priority classes, namely connections 1, 2 and 3, because it belongs to a higher priority class. Therefore, connection 0 is given all the available slots and achieves a very high MAC throughput at $t = 12$, due to the transmission of its queued packets.

Until all queued packets of connection 0 are transmitted, both connections 1 and 2 are not served even though they are at the α-queue of priority class that corresponds to PHY rate of 15 $Mbps$. Hence, both connections 1 and 2 are not allocated any transmission slots for a few consecutive MAC frames when connection 0 is allocated all available transmission slots. When all the queued packets of connection 0 have been transmitted, as indicated by QoS Recovery Detection, it is redirected back to the α-queue of priority class that correspond to PHY rate of 20 $Mbps$.

Due to the temporary transmission starvation experienced by both connections 1 and 2, they experience an increase in delay. It so happens that the increase in delay is still below D_{req}, hence no QoS Violation is detected. As soon as connections 1 and 2 are served, the delay experienced by both connections starts to decrease and they achieve a MAC throughput that is slightly above their offered load, due to the transmission of queued packets. Until all queued packets of connections 1 and 2 are transmitted at $t = 16.4$, connection 3 is not served, as shown in Figure 6.4.3 (c) where connection 3 only receives service between $t = 16.4$ and $t = 18$ in a few MAC frames.

From Figure 6.4.6 (b), the total MAC throughput of the network has a peak at

$t = 12$, due to the transmission of the queued packets of connection 0 at its high PHY mode. Note that the total MAC throughput achieved in this scheme after all the queued packets of connection 0 are cleared, is slightly lower than the total MAC throughput achieved in scheme 1. This is due to the slightly higher MAC throughput achieved by connection 0 in scheme 1 when its queued packets is slowly cleared at the β-queue from time soon after $t = 12$ until $t = 16.4$, as shown in Figure 6.4.1 (c). In fact, we expect the average total MAC throughput achieved in schemes 1 and 2 between $t = 12$ and $t = 16.4$ to be the same. The only difference between the schemes is that scheme 2 achieves a very high total MAC throughput at $t = 12$ for a very short time interval, while scheme 1 achieves a slightly higher total MAC throughput for a longer time interval. From $t = 16.4$ onwards, the total MAC throughput for schemes 1 and 2 is identical, except there are a few very small mismatched dips.

Hence, there is no impact on the cumulative number of bits transferred by all connections between the schemes when a connection at the β-queue changes its PHY mode to a higher PHY rate in our example. However, if the number of slots available to serve this connection is not sufficient to support the offered load of this connection after changing its PHY mode to a higher rate, scheme 2 would again demonstrate a higher MAC throughput than scheme 1.

In order to support our argument above, we extend our investigation on this issue by carrying out a set of experiments with different offered loads, ranging from 2.1 to 3.1 *Mbps*. This range of values is chosen in order to have two connections in the α-queue and the β-queue respectively before one of the connections in the β-queue changes its PHY mode to a higher PHY mode. The improved cumulative number of bits transferred under scheme 2 (that is the cumulative number of bits transferred under scheme 2 minus the cumulative number of bits transferred under scheme 1 in this region) is plotted in Figure 6.4.7.

With a connection's offered load of 2.4 *Mbps*, we start to see an increase in total MAC throughput achieved under scheme 2 in this region, where a change in PHY

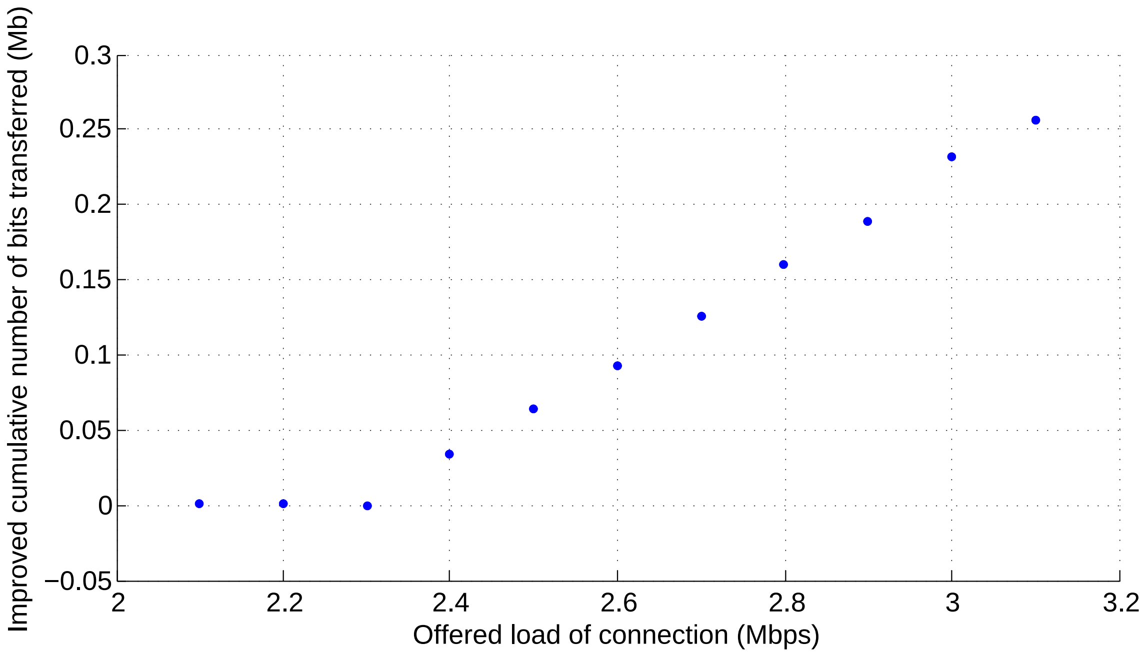

Figure 6.4.7: The improved cumulative number of bits transferred under scheme 2 during the time interval between $t = 12$ and $t = 18$.

mode of connection 0 from a lower to a higher PHY rate is not sufficient to release enough bandwidth to allow it to be redirected back to the α-queue under scheme 1. Note that the improved cumulative number of bits transferred under scheme 2 is expected to be zero between connections' offered loads of 2.1 and 2.3 $Mbps$, but the values obtained from the experiment range from -6.4×10^{-5} to 3.84×10^{-4} Mb. This is due to the variations in packet packing and fragmentation.

Region 6 $(18 \leq t < 20)$ – Scheme 1

At $t = 18$, connections 0, 2 and 3 are being served at the α-queue, leaving only a few slots to serve the β-queue. Hence, when connection 0 changes its PHY mode back to its initial PHY rate, congestion occurs in the α-queue. As a result, an increase in delay experienced by connections at the α-queue is observed and a QoS Violation event is detected, which sends connection 0 to the β-queue. At the β-queue, connection 1 is being served because it wins the tie breaker based on each connection's index.

Region 6 $(18 \leq t < 20)$ **– Scheme 2**

At $t = 18$, connection 0 is being served at the α-queue of the priority class that corresponds to the PHY rate of 20 *Mbps* and both connections 1 and 2 are being served at the α-queue of the priority class that corresponds to the PHY rate of 15 *Mbps*. This leaves only a few slots to serve connection 3 at the β-queue of the priority class that corresponds to a PHY rate of 15 *Mbps*. When connection 0 changes its PHY mode back to its initial rate, it is moved to the α-queue of priority class that corresponds to the PHY rate of 15 *Mbps*.Hence, congestion occurs in the α-queue of priority class that corresponds to the PHY rate of 15 *Mbps*.

As a result, an increase in delay experienced by connections at the α-queue is observed and a QoS Violation event is detected, which sends connection 0 to the β-queue. At the β-queue, connection 3 is being served because it wins the tie breaker based on each connection's index.

6.4.3 Discussion

Due to the service sequence, from the highest to the lowest priority class, of the PBDQ scheduler, the number of high PHY mode connections served is maximised despite the changes in PHY mode of the connections. Therefore, the cumulative number of bits transferred of a network being scheduled with a PBDQ scheduler is higher than that proposed in Chapter 5. This affirms our argument that the proposal of Chapter 5 can not deliver other objectives effectively.

However, the QoS experienced by three of the connections in the network with the PBDQ scheduler is affected, due to the moving of connections between priority classes. If we consider the QoS of each connection throughout the experiment, we would obtain a single connection that experiences good service throughout the experiment with the PBDQ scheduler, instead of two, which is achieved under scheme

1. This is the trade off for trying to maximise the throughput of the network at all times.

In our experiment, we showed two possible events that may affect the ability of a network to maximise its throughput; a connection at the α-queue changes its PHY mode to a lower PHY rate and a connection at the β-queue changes its PHY mode to a higher PHY rate. In the worst-case scenario, these events may significantly reduce the throughput of the network since it is possible for these events to cause connections with a lower PHY mode to stay in the α-queue, while connections with a higher PHY mode stay in the β-queue.

6.5 Other Applications of the PBDQ scheduler

Instead of using PHY mode to determine the priority class of a connection, parameters such as the revenue of a connection or the Gold/Silver/Bronze priorities used in SS differentiation, can be employed. Hence, the PBDQ scheduler provides differentiation between connections that belong to the same CoS based on any criteria specified by the network. Note that the PBDQ scheduler is very flexible and it can be made to have no effect by moving all connections to a single priority class.

One of the other possible applications of the PBDQ scheduler is to schedule different CoSs in an 802.16 system. That is, assigning a priority class to each CoS of the 802.16 system. For example, we could assign the highest priority class to the UGS traffic, followed by the ertPS, rtPS, nrtPS and BE traffic. In this case, a strict priority rule can be employed to allow the highest priority class to use up all the available slots in the system if required. On the other hand, a different threshold can also be set for each CoS to determine the proportion of throughput that each CoS can transmit in the system during congestion. Note that connection movement between priority classes is not needed in this application.

6.6 Summary

In this chapter, we have proposed the Priority-Based Dual-Queue scheduler, which is made up of multiple DQs differentiated by connections' priority classes. The connections classified with the highest priority are given all the transmission opportunities they can use before the other classes receive any transmission opportunities. When there are changes in the priority class of some connections, these connections will be moved to their new priority class. Hence, the PBDQ scheduler ensures that connections with a higher priority class are served ahead of connections with lower priority classes.

Given the objectives defined for a network, each connection in the network can be assigned a priority, which can be used by the PBDQ scheduler to achieve the network's objectives. We have showed that the primary objective of a network to maximise throughput can be achieved by using the PBDQ scheduler. We have also discussed other applications of the PBDQ scheduler in this chapter.

Chapter 7

Joint DL and UL Dual-Queue Scheduling

In this chapter, we investigate the benefits of carrying out the DLDQ and ULDQ scheduling jointly. We first investigate a scenario where an 802.16 network consists of only one-directional rtPS connections. We then extend our investigation to a scenario where the network consists of bi-directional rtPS connections.

7.1 Introduction

In Chapter 5, we investigated the DLDQ and ULDQ schedulers separately, in order to show the different DQ mechanisms on the DL and the UL respectively. However, we argued that the DL and the UL scheduling processes should be carried out jointly, in Chapter 3, in order to provide QoS support for the different CoSs defined in an 802.16 system. In that chapter, we also discussed the importance of dynamic subframe partitioning for CoS differentiation. In this chapter, we affirm our argument by investigating the benefits of scheduling the DL and UL jointly. We define a connection as receiving "good service" when it experiences a delay of less than the delay requirement of the network (D_{req}) and it has no packet loss.

By considering both DL and UL scheduling together, resources in an 802.16

network can be allocated flexibly, in order to maximise the number of connections that experience good service. We contend that this objective may not be achieved without a joint DL and UL scheduler.

The terms used frequently throughout this chapter are listed below:

S: total number of slots allocated to real-time services after any ARQ allocations.

S_t: total number of slots allocated to real-time services for $t =$ DL, UL.

$S_{t,\gamma}$: number of slots allocated to the γ-queue for $t =$ DL, UL, where $\gamma = \alpha, \beta$.

X_t: total number of slots required for real-time services for $t =$ DL, UL.

$X_{t,\gamma}$: number of slots required for the γ-queue for $t =$ DL, UL, where $\gamma = \alpha, \beta$.

From these quantities, we can state that the network is overloaded when $X_{DL} + X_{UL} = X_{DL,\alpha} + X_{DL,\beta} + X_{UL,\alpha} + X_{UL,\beta} > S$.

7.2 Handling One-Directional Connections

In this section, we investigate a joint DL and UL scheduler, which has as its primary objective to maximise the number of one-directional rtPS connections in the network that experience good service. In order to achieve this objective, dynamic sub-frame partitioning is a prerequisite. We also select maximising throughput as the scheduler's secondary objective.

Since maximising throughput is selected as the secondary objective, the connection prioritisation mechanism employed in this section is first based on PHY mode. Then, a tie breaker based on connection indices is employed: for the DL, the highest priority is assigned to the connection with the lowest index; while for the UL, the highest priority is assigned to the connection with the highest index. Note that we have deliberately chosen different tie breakers for the DL and UL. The reason for this is explained in Section 7.5.1.

We first describe a scheme known as the "no coordination" scheme to create a benchmark, where the DLDQ and the ULDQ scheduling processes are carried out separately. That is, the DLDQ and ULDQ scheduling processes are carried out independently for detecting QoS Violation events in the network. We then describe a joint DL and UL scheduling scheme known as the "basic coordination" scheme, where the joint scheduler is responsible for scheduling the DL and the UL connections in an 802.16 network together.

7.2.1 No Coordination (Benchmark for Comparison)

This scheme has two schedulers: the DLDQ and ULDQ schedulers. These schedulers are independent and they schedule transmissions on the DL and the UL respectively. This scheme acts as a benchmark to compare to our joint DL and UL scheduler in terms of maximising the number of connections in the network that experience good service.

For every MAC frame, the DL and UL are allocated with S_{DL} and S_{UL} slots respectively by the main scheduler. The algorithm used to allocate S_{DL} and S_{UL} is shown below. When the network is overloaded ($X_{DL} + X_{UL} > S$), we apply the allocation rule described in Chapter 5, which is known as the "DL and UL allocation rule". Reiterating this rule, we have

$$S_{DL} = \begin{cases} r_{dl} \times S, & \text{if } X_{DL} \geq r_{dl} \times S \text{ and } X_{UL} \geq r_{ul} \times S, \\ X_{DL}, & \text{if } X_{DL} \leq r_{dl} \times S, \\ S - X_{UL}, & \text{if } X_{UL} \leq r_{ul} \times S, \end{cases} \tag{7.2.1}$$

and

$$S_{UL} = S - S_{DL}, \tag{7.2.2}$$

where $r_{dl} = r_{ul} = 0.5$.

From the above, we notice that there is no differentiation between the requests from the α-queue and the β-queue, since the main scheduler considers aggregate requests from the α-queue and the β-queue respectively for dynamic partitioning between the DL and the UL sub-frames.

Given S_{DL} and S_{UL}, the DLDQ and the ULDQ schedulers allocate slots to the α-queue and the β-queue on the DL and the UL respectively. Hence, differentiation between the requests from the α-queue and the β-queue is carried out by the relevant scheduler:

$$S_{t,\alpha} = \min(S_t, X_{t,\alpha}) \text{ and } S_{t,\beta} = S_t - S_{t,\alpha}, \qquad \text{for } t = \text{DL, UL.}$$

Therefore, $S_{DL,\beta}$ and $S_{UL,\beta}$ can be non-zero at the same time, which implies that two connections can be served at the β-queue at the same time.

7.2.2 Basic Coordination

The idea behind the basic coordination scheme is to schedule both the DL and the UL connections together by a joint DL and UL scheduler. Hence, a main scheduler that distributes slots to the DL and the UL is not required. However, an extra tie breaker between the DL and UL connections that have the same PHY mode and index is needed for the connection prioritisation mechanism. We propose a tie breaker that is based on the location of the most recent QoS Violation event being detected; that is, if the QoS Violation event was detected on the DL, a DL connection would be chosen to be sent to the β-queue, and this connection would also be chosen to be served using the left over bandwidth after serving the α-queue.

The joint DL and UL scheduler serves the α-queue on both the DL and the UL ahead of the β-queue on both the DL and the UL. Between the connections in the α-queue, they are served based on a rule similar to the DL and the UL allocation rule considering only slot requests from the α-queue. When there is left over bandwidth from serving the α-queue, only one connection from the β-queue will be served. This is different from the approach taken in the "no coordination" scheme, where it is very likely that two connections at the β-queue will be served at the same time (one connection from the DL and another connection from the UL). This is the main feature in our proposed joint DL and UL scheduler that helps to maximise the number of connections in the network that experience good service.

In other words, when a QoS Violation event is detected on either the DL or the UL, a connection with the lowest priority is chosen from the pool of DL and UL connections at the α-queue to be sent to the β-queue. Similarly, a connection with the highest priority will be chosen from the pool of DL and UL connections at the β-queue to be served when there are slots left after serving the α-queue.

7.3 Experiment with One-Directional Connections

In the following, we consider an experiment to compare the performance of the two schemes discussed above. We consider a simple network that consists of two rtPS connections on the DL (DL1 and DL2) and the UL (UL1 and UL2) respectively. Similar to our previous experiments in Chapters 5 and 6, we use CBR traffic sources to demonstrate the DQ scheduling features, as this gives us a predictable and readily demonstrable environment.

Each of these connections has an offered load of 2 $Mbps$, where this offered load is maintained constant throughout the remainder of this experiment. Further, the same network parameters presented in Table 5.5.1 from Chapter 5 are used. Initially, all connections have the same PHY mode (PHY rate = 22.5 $Mbps$).

This experiment is carried out for 20 seconds, where the first 5 seconds is reserved for the initialisation process. To create a changing environment, we also introduce a variable rate UGS traffic source with packet size of 100 B (including 20 B of IP header). The UGS source is scheduled at the highest priority and is a DL connection. It is also possible for the UGS traffic to take up all the slots available, if all slots are required for transmission.

7.3.1 Experimental Regions

In order to compare the schemes, there are 3 congestion regions in this experiment. For both of the schemes, we carry out the same experiment with identical offered traffic and compare the performance between the schemes for each of the congestion regions defined below:

- Region 1 ($6 \leq t \leq 8$): Congestion occurs due to an increase in UGS traffic. At $t = 6$, the UGS traffic is increased to 3 $Mbps$. It is further increased to 5 $Mbps$ at $t = 7$. Therefore, the increase in UGS traffic makes both the DL and the UL appear congested in this interval.

- Region 2 ($10 \leq t \leq 12$): Congestion occurs due to a change in PHY mode of both UL connections. At $t = 10$, both the UL connections change PHY rate from 22.5 to 10 $Mbps$. Therefore, this event makes the UL appear congested, but not the DL. At $t = 12$, both UL connections change their PHY rate back to 22.5 $Mbps$.

- Region 3 ($14 \leq t \leq 18$): This region can be divided into 2 parts. First, at $t = 14$, congestion occurs due to a change in PHY mode of connection DL1 from 22.5 to 10 $Mbps$. This makes the DL appear congested. Second, at $t = 15$, congestion occurs due to a change in PHY mode of connection UL1 from 22.5 to 10 $Mbps$. This makes the UL appear congested.

7.3.2 Experimental Results

In the following, we discuss the performance of each scheme based on the congestion regions defined in the previous section.

Region 1 ($6 \leq t \leq 8$) – No coordination

Under the "no coordination" scheme, congestion due to an increase in the UGS traffic impacts both the DL and UL, since the number of slots available to serve the

DL and the UL has decreased. Given the total number of slots available to serve the DL and the UL, slots are allocated to the DL and the UL based on the DL and UL allocation rule.

At the start of this region ($t = 6$), we observe that a QoS Violation event is detected earlier on the UL, which is a feature that has been discussed in Chapter 5. This is due to the fact that less time is taken for the delay of the connections being served at the α-queue on the UL to exceed D_{req} since the average delay of the UL connections is D_{mac} higher than the average delay of the DL connections (see Figure 7.3.1 (b)). This is also due to the more conservative approach taken in the UL QoS Violation Detection. After the detection of a QoS Violation event on the UL, which sends connection UL1 to the β-queue, another QoS Violation event is detected on the DL because both the DL and UL are congested, which in turn sends connection DL2 to the β-queue.

At the β-queue, both connections UL1 and DL2 achieve a MAC throughput of 1.5 *Mbps* during the time interval between $t = 6.2$ and $t = 7$, as shown in Figure 7.3.1 (c). Since the offered load of each connection in the network is 2 *Mbps*, we know that the bandwidth allocated to these connections could be distributed in a way that allows one of them to be redirected back to the α-queue. This is, however, not possible as the DLDQ and ULDQ schedulers are independent. Therefore, the number of connections in the network that experience good service is not maximised.

At $t = 7$, these connections achieve a lower MAC throughput of 0.5 *Mbps*, due to the further decrease in available slots to serve the rtPS traffic. Note that both connections achieve the same MAC throughput due to the DL and UL allocation rule and they both experience degraded service.

At the end of this region, the rate of the UGS traffic is decreased to a value close to zero, which releases enough slots to accommodate all connections in the experiment. Therefore, all connections are redirected back to the α-queue. These events are also repeated at the end of regions 2 and 3. From Figure 7.3.2 (c), we notice that all connections are considered as receiving degraded service even though

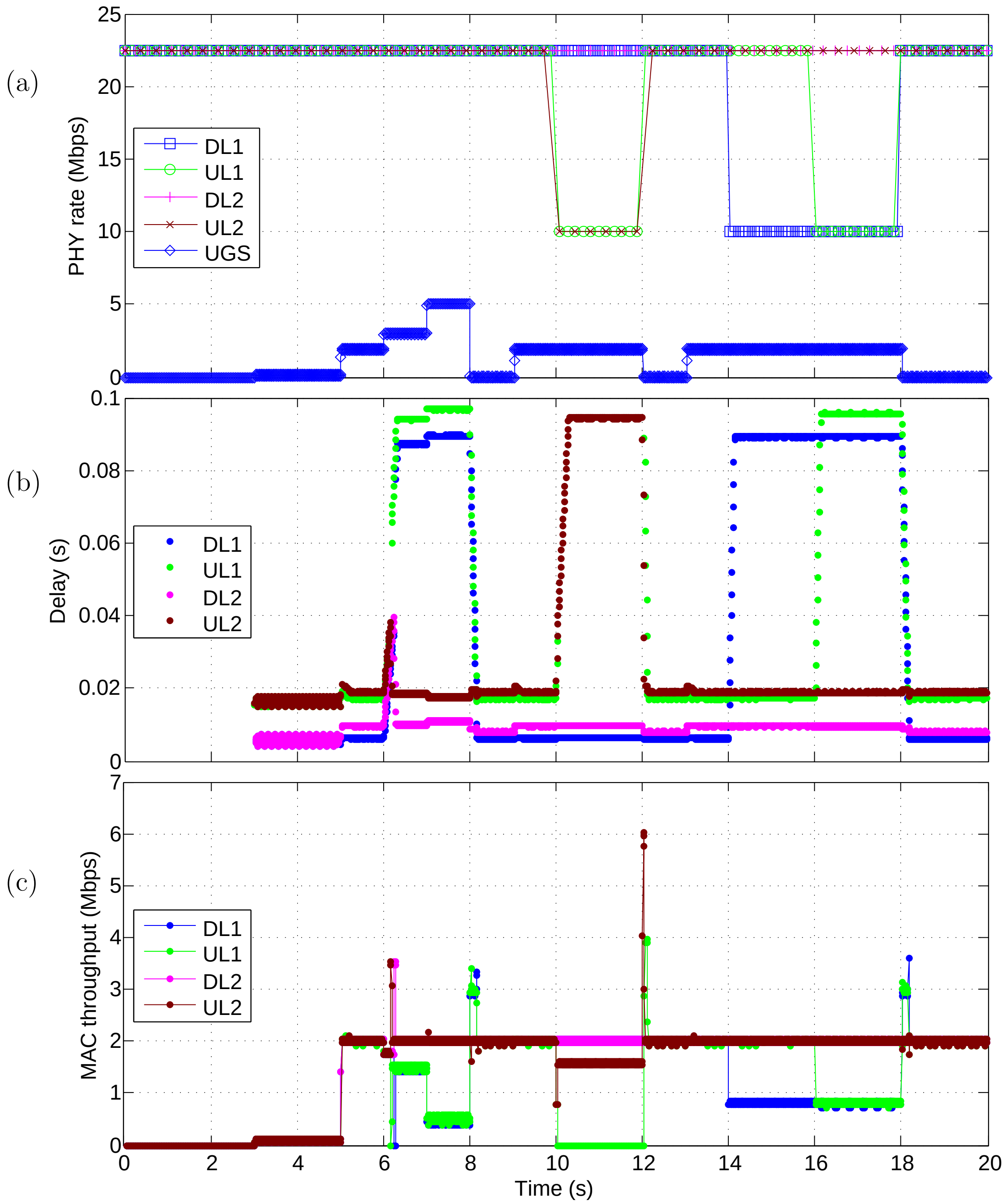

Figure 7.3.1: No coordination: (a) PHY rate, (b) packet delay, and (c) MAC throughput.

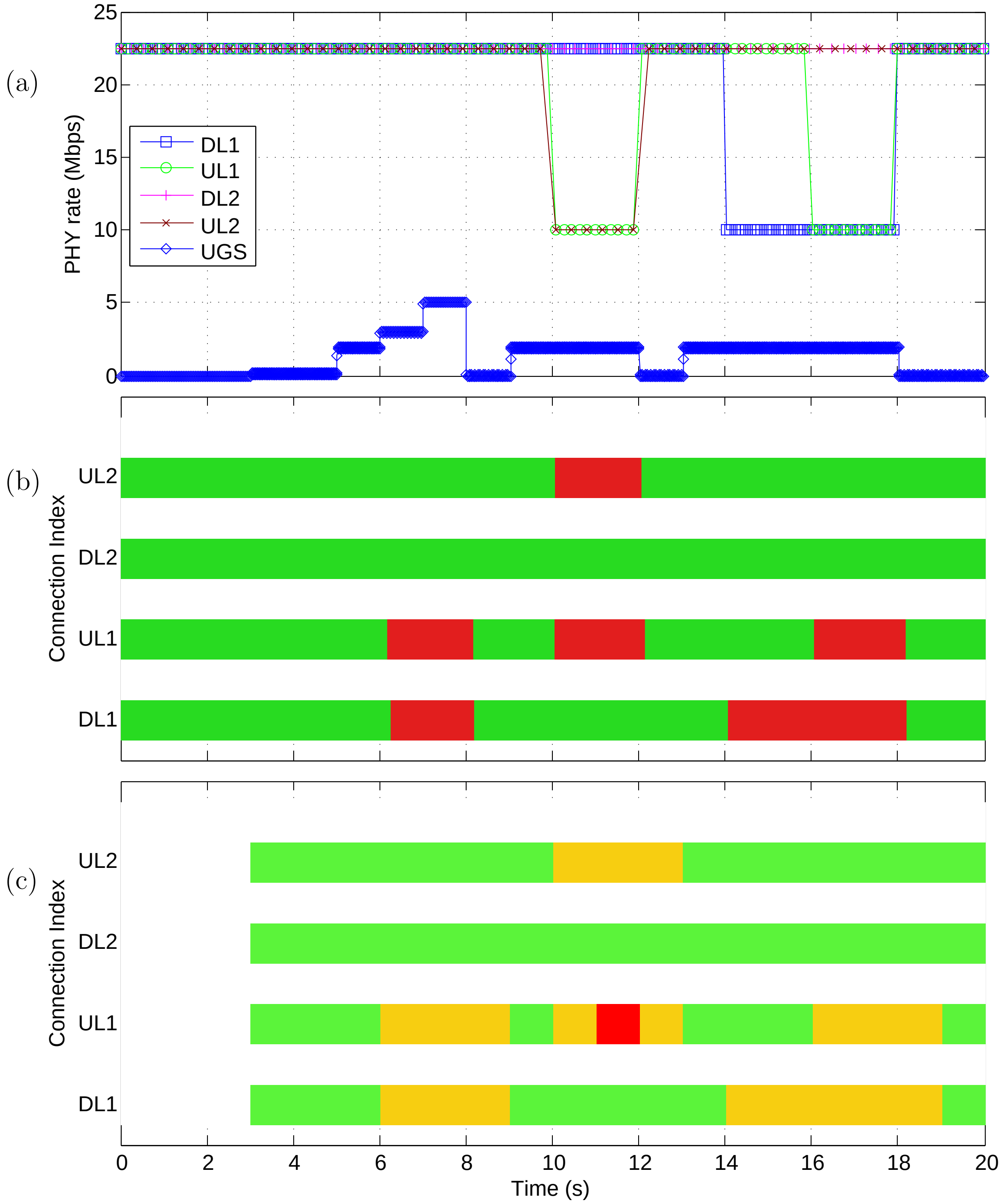

Figure 7.3.2: No coordination: (a) PHY rate, (b) connections at each queue; α-queue (green), β-queue (red), and (c) QoS received; good service (green), degraded service (amber) and no service (red).

they have been redirected back to the α-queue soon after $t = 8$. This is because their QoS is evaluated every second and they receive degraded service at the beginning of the interval starting at $t = 8$.

Region 1 ($6 \leq t \leq 8$) – Basic coordination

Similar to the no coordination scheme, a QoS Violation event is detected earlier for the UL. That is, at $t = 6$, an increase in the delay of all connections is observed, which eventually results in connection UL1 being sent to the β-queue. Note that connection UL1 is chosen because it wins the tie breaker between connections UL1 and DL2.

However, some slots are released as soon as connection UL1 is sent to the β-queue, which become available to serve the remaining connections at the α-queues of both the DL and UL. Hence, this brings down the average delay experienced by the remaining DL and UL connections at the α-queue, and so, no further QoS Violation events are detected. At the β-queue, connection UL1 is served using the left over slots after the α-queue is served and it achieves a MAC throughput of 1 *Mbps* (see Figure 7.3.3 (c)). Therefore, only a single UL connection is sent to the β-queue soon after $t = 6$, and the number of connections in the network that experience good service is maximised.

At $t = 7$, the rate of the UGS traffic increases further, which causes another QoS event to be detected on the DL. As a result, connection DL2 is sent to the β-queue, since it has the lowest priority among the remaining connections at the α-queues. Now, both connections UL1 and DL2 are at their respective β-queues, but connection DL2 wins the tie breaker and hence it is chosen to be served. This is seen from Figure 7.3.3 (c), where the MAC throughput of connection UL1 drops to zero as soon as connection DL2 is sent to the β-queue, and connection DL2 achieves a MAC throughput of 1 *Mbps*.

Under this scheme, the number of connections that experience good service is maximised at 3 from $t = 6$ to $t = 7$, as seen in Figure 7.3.4 (c), because the joint

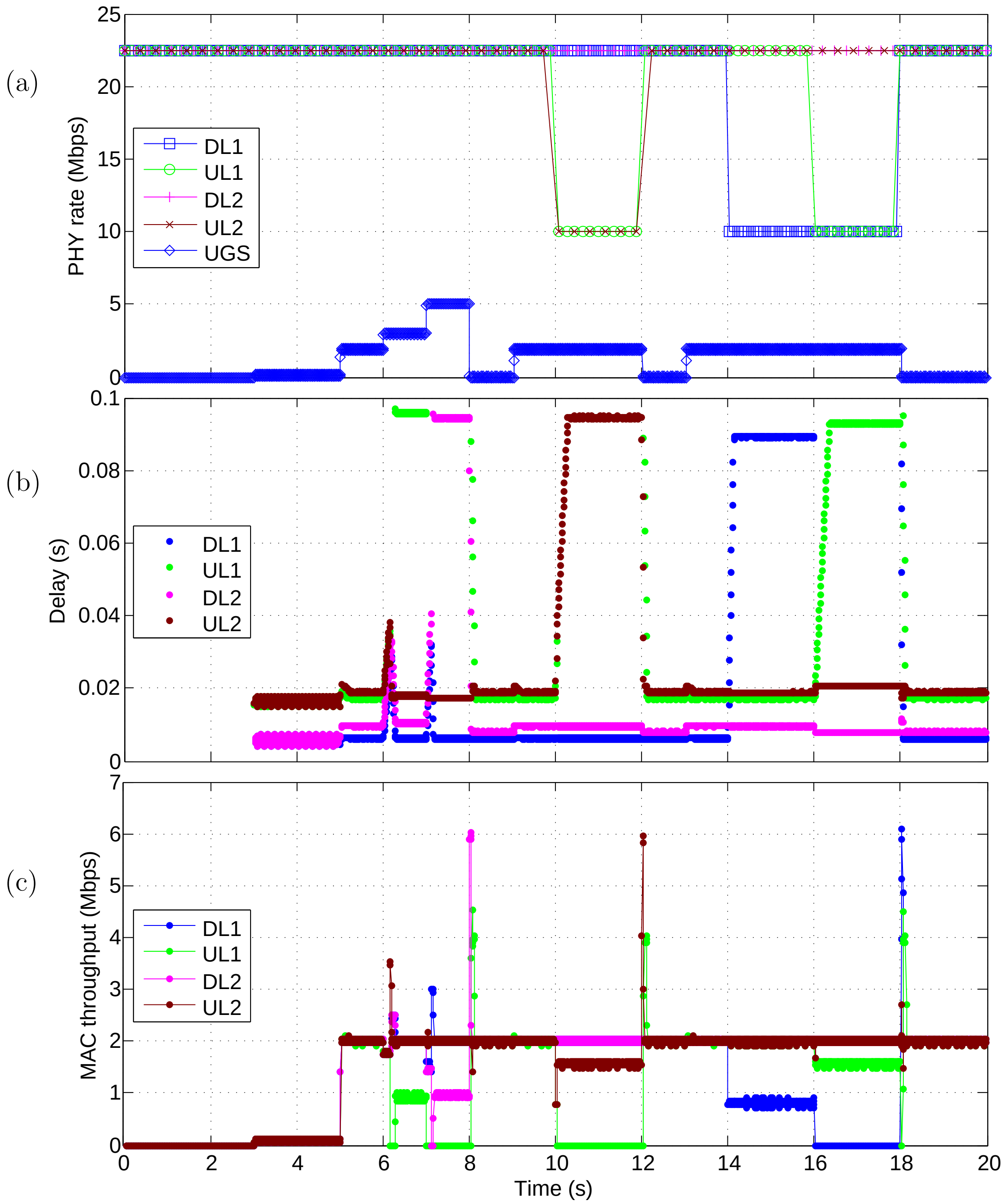

Figure 7.3.3: Basic coordination: (a) PHY rate, (b) packet delay, and (c) MAC throughput.

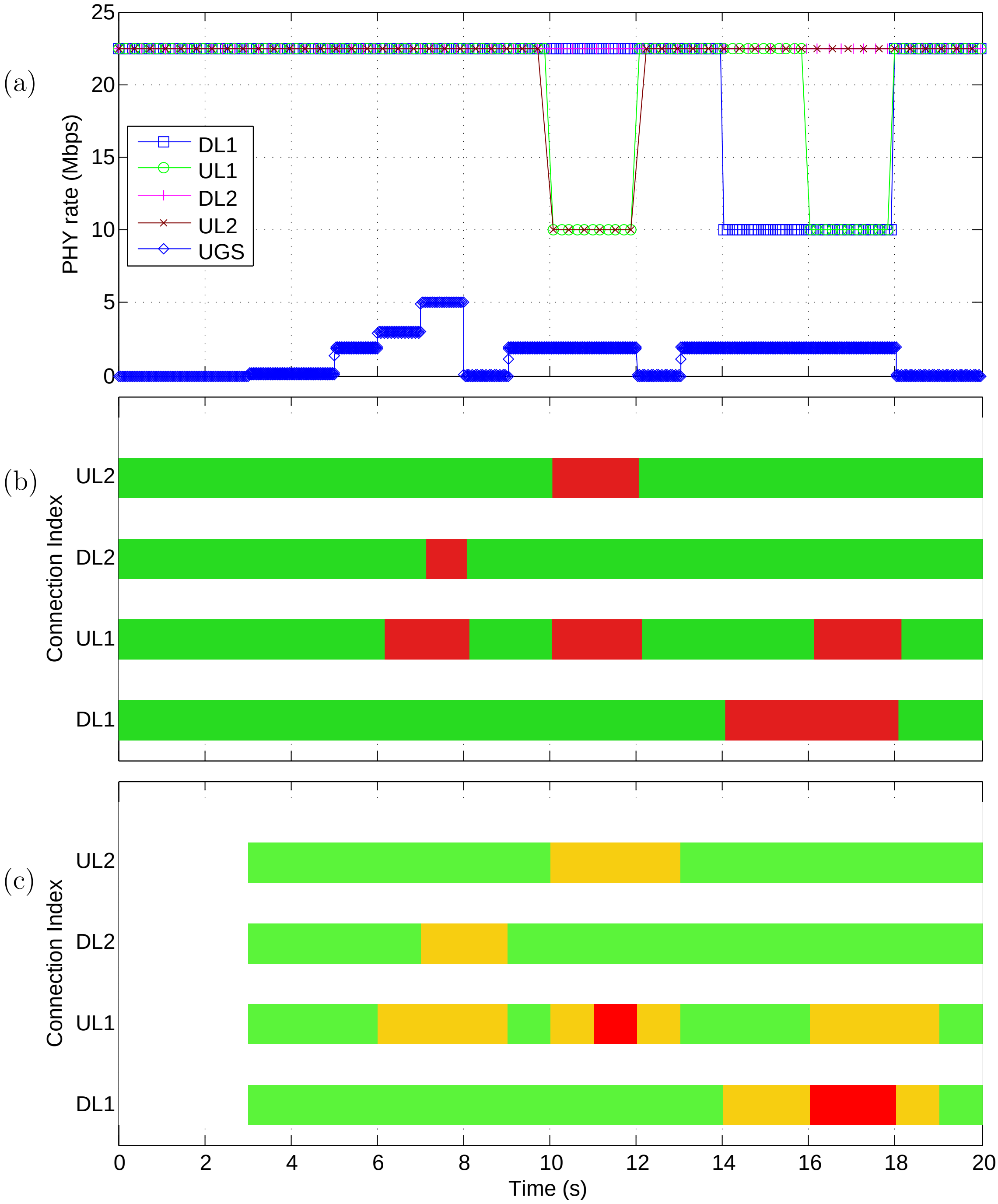

Figure 7.3.4: Basic coordination: (a) PHY rate, (b) connections at each queue; α-queue (green), β-queue (red), and (c) QoS received: good service (green), degraded service (amber) and no service (red).

DQ scheduler handles the DL and the UL connections together. Further, when there are connections in the β-queue on both the DL and the UL, only one of these connections is served under this scheme.

Region 2 ($10 \leq t \leq 12$) − No coordination

Since the congestion in this region is due to the increase in the number of slots required for each UL connection, we observe that QoS Violation events are detected only on the UL, which causes both connections UL1 and UL2 to be sent to the β-queue. From Figure 7.3.2 (c), we notice that the average delay of the DL connections is not affected by the congestion on the UL.

At the β-queue, connection UL2 is chosen to be served because it has the highest priority and it achieves a MAC throughput of 1.5 $Mbps$, as shown in Figure 7.3.1 (c), and hence, it experiences degraded service. Since connection UL1 is not being served at all, it experiences no service.

Region 2 ($10 \leq t \leq 12$) − Basic coordination

In this region, the UL connections have a lower priority than the DL connections because they have changed to a lower PHY mode. Hence, at $t = 10$, QoS Violation events are detected in the network, which sends both connections UL1 and UL2 to the β-queue, one after another.

At the β-queue, connection UL2 has a higher priority than connection UL1, and so, it is chosen to be served and it achieves a MAC throughput of 1.6 $Mbps$. From Figure 7.3.4 (c), we observe that connection UL2 experiences degraded service as it is being partially served, while connection UL1 experiences no service from $t = 11$ to $t = 12$.

Region 3 ($14 \leq t \leq 18$) **– No coordination**

In this region, we observe that a QoS Violation event is detected on the DL when the DL is congested at $t = 14$. Therefore, connection DL1 is sent to the β-queue because it has the lowest PHY mode among the DL connections at the α-queue. Notice that the average delay of the UL connections is not affected in this region.

At $t = 16$, the UL is congested due to the change in PHY mode of connection UL1 to a lower PHY mode, which results in a QoS Violation event to be detected on the UL. Hence, connection UL1 is sent to the β-queue because it has the lowest PHY mode among the UL connections at the α-queue. Notice that the MAC throughput of connection DL1 is not affected by the congestion that stems from the UL. At the β-queue, both connections DL1 and UL1 achieve a MAC throughput of 0.8 *Mbps*. Since connections DL2 and UL2 are never sent to the β-queues, they both experience good service in this region.

Region 3 ($14 \leq t \leq 18$) **– Basic coordination**

At $t = 14$, connection DL1 changes its PHY mode to a lower PHY rate, which causes congestion in the network. As a result, a QoS Violation event is detected, which sends connection DL1 to the β-queue as it has a lower PHY mode compared to all the other DL or UL connections at the α-queues. At the β-queue, connection DL1 achieves a MAC throughput of 0.8 *Mbps*.

Similarly at $t = 15$, connection UL1 changes its PHY mode to a lower PHY mode, which causes another QoS Violation event to be detected and connection UL1 is sent to the β-queue as it has a lower PHY mode compared to all other connections at the α-queues. Between connections UL1 and DL1 at the β-queues, connection UL1 is chosen to be served as it has the highest priority.

From Figure 7.3.4 (c), we observe that connection DL1 experiences degraded service from $t = 14$ to $t = 16$, and experiences no service from $t = 16$ to $t = 18$, after connection UL1 is sent to the β-queue.

7.3.3 Summary

In Table 7.3.1, we summarise our experimental results, where a tick indicates that the number of one-directional connections in the network that experience good service is maximised.

Table 7.3.1: Summary of the joint DL and UL scheme in terms of maximising the number of one-directional connections in the network that experience good service.

Schemes / Experimental Regions	Region 1	Region 2	Region 3
No coordination	×	✓	✓
Basic coordination	✓	✓	✓

Under the "no coordination" scheme, we observe that the network's objective to maximise the number of connections in the network that experience good service is not achieved in region 1, where congestion in the network is due to a decrease in the number of slots to serve the rtPS traffic. However, in regions 2 and 3, the network's objective is achieved. This is because the congestion in these regions is due to a change to a lower PHY mode by one or more connections. When a QoS Violation event is detected, these connections that cause congestion will be sent to the β-queue, which removes the source of congestion. Further, it is possible for the network to serve two connections at the β-queue at the same time under this scheme, and so a connection is not redirected back to the α-queue at the earliest opportunity.

Under the "basic coordination" scheme, the DQ scheduler of the network handles the DL and the UL connections together. Hence, when QoS Violation events are detected in the network on either the DL or the UL, the scheduler chooses the lowest priority connection at the α-queue from either the DL or the UL to be sent to the β-queue. Further, the bandwidth left after serving the α-queue is allocated to one connection at the β-queue on either the DL or UL. Due to these features, the number of one-directional connections that experience good service in the network is maximised in all 3 regions of congestion.

7.4 Handling Bi-Directional Connections

In this section, we extend our investigation to bi-directional sessions, such as Voice over IP and video conferencing. Each of these sessions consists of bi-directional connections between the BS and the SSs. We regard such a session as receiving good service only if the connections in both directions are receiving good service. We refer to a connection as the partner connection of another connection if they both belong to the same session.

Assuming that the BS has sufficient information to identify the two connections involved in such a session, we contend that the DLDQ and ULDQ scheduling processes must be carried out jointly in order to maximise the number of bi-directional sessions that experience good service.

We investigate the benefits of carrying out the DLDQ and the ULDQ scheduling processes jointly through two schemes: “partial coordination” and “full coordination”. Both aim to maximise the number of bi-directional sessions that experience good service. When the network is not congested, there is no difference between the two schemes because all DL and UL connections can be fully served.

7.4.1 Partial Coordination

The “partial coordination” scheme is an extension of the “basic coordination” scheme described in Section 7.2.2. That is, a joint DL and UL scheduler is responsible for the scheduling of both the DL and UL connections. However, the first factor considered in the connection prioritisation mechanism is whether a connection has its partner connection in the β-queue or the α-queue. If a connection has its partner connection in the α-queue, it is assigned a higher priority than others that have not. In contrast, if a connection has its partner connection in the β-queue, it is assigned a lower priority than others that have not. After this factor is considered, other factors such as the PHY mode and index are considered.

Note that it is only possible to have at most one session that has one of its connections at the α-queue in one direction and the other connection at the β-queue in the other direction. This is because a connection that has its partner connection in the β-queue, is the next connection to be sent to the β-queue, when the scheduler needs to send another connection to the β-queue.

Similar to the "basic coordination" scheme, the joint scheduler ensures that the α-queues on the DL and the UL are served ahead of the β-queues. Between the α-queue of the DL and the UL, slots are allocated based on a rule similar to the DL and the UL allocation rule described by equations 7.2.1 and 7.2.2, considering only slot requests from the α-queue.

7.4.2 Full Coordination

This scheme is a further extension of both the "basic coordination" and "partial coordination" schemes, and it aims to penalise both connections of a bi-directional session together instead of a single connection. In other words, the joint DL and UL scheduler handles connections in the network in terms of sessions. Therefore, we need to define a session prioritisation mechanism. This mechanism may be more complicated than that described in the "partial coordination" scheme, because two connections are involved in a session. For instance, if we would like to maximise throughput, the PHY mode of both connections of a session has to be considered together to work out the priority of the session.

The major differences between this scheme and the "partial coordination" scheme are the decision on which, and how many, connections are to be sent to the β-queue, when a QoS Violation event is detected on either the DL or the UL, and the way that the number of slots are allocated in the β-queue. Under this scheme, the joint DQ scheduler sends the whole bi-directional session with the lowest priority into the

β-queue when a QoS Violation event is detected, rather than sending just one direction of the session into the β-queue. Similarly, the whole session with the highest priority must be redirected back to the α-queue when there is bandwidth available to accommodate the whole session, rather than redirecting just one direction of the session back to the α-queue.

In order to ensure that a whole session can be redirected back to the α-queue together, the number of slots left to serve the β-queue is allocated to both connections of the session with the highest priority. However, there are many factors that may affect the way that slots are allocated to both connections of a session, such as the PHY mode and the offered load of each connection. We choose to allocate the number of slots left to serve both connections of the session at the β-queue evenly, that is, this approach maintains air-time fairness between these connections.

7.5 Experiment with Bi-Directional Connections

The experiment to be carried out in this section is predominantly identical to that described in Section 7.3, except now connection DL1 is a partner connection of connection UL1 from the same bi-directional session (V1) and similarly connection DL2 is a partner connection of connection UL2 from the same bi-directional session (V2). We carry out an experiment for the "partial coordination" and the "full coordination" schemes. Note that we select a simple session prioritisation mechanism based on index for the "full coordination" scheme, that is session V2 has a higher priority than session V1. We also carry out a benchmarking experiment with the "no coodination" scheme described in Section 7.2, in order to compare the two schemes properly. In fact, the result for this benchmarking experiment is identical to that shown in Section 7.2, by considering connections DL1 and UL1 to be from the same bi-directional session, and similarly for connections DL2 and UL2. Therefore, we refer to the results described in Section 7.2 for this benchmarking experiment.

7.5.1 Experimental Results

In the following, we discuss the performance of the partial and full coordination schemes based on the congestion regions defined in Section 7.3. We also revisit the results for the "no coordination" scheme in terms of maximising the number of bi-directional sessions that experience good service.

Region 1 ($6 \leq t \leq 8$) – No coordination

From Figure 7.3.2 (c), we observe that both connections UL1 and DL2 experience degraded service in this region. Hence, both sessions V1 and V2 experience degraded service. Note that we deliberately choose different tie breakers for the DL and UL, in order to obtain this result. If the tie breakers for the DL and UL were the same, we would have a single session experiencing degraded service, instead of two.

Region 1 ($6 \leq t \leq 8$) – Partial coordination

Under the partial coordination scheme, congestion due to the increase in the UGS traffic also causes an increase in the average delay experienced by the DL and UL connections at the α-queues, as shown in Figure 7.5.1 (b). A QoS Violation event is also detected earlier on the UL as explained before, which results in the moving of connection UL1 to the β-queue.

At $t = 7$, the UGS traffic is further increased to a higher rate, which causes another congestion at the α-queue. When this happens, we observe that the MAC throughput of connection UL1 drops to zero, since it is being served at the β-queue. At this stage, there are two DL connections and one single UL connection at the α-queue, competing for transmission slots. These connections at the α-queues will be allocated slots based on a rule similar to the DL and UL allocation rule. Therefore, connection UL2 receives all the slots it needs and so its delay is not affected. In contrast, a QoS Violation event is detected on the DL, which causes connection DL1 to be sent to the β-queue because it is the partner connection of connection UL1

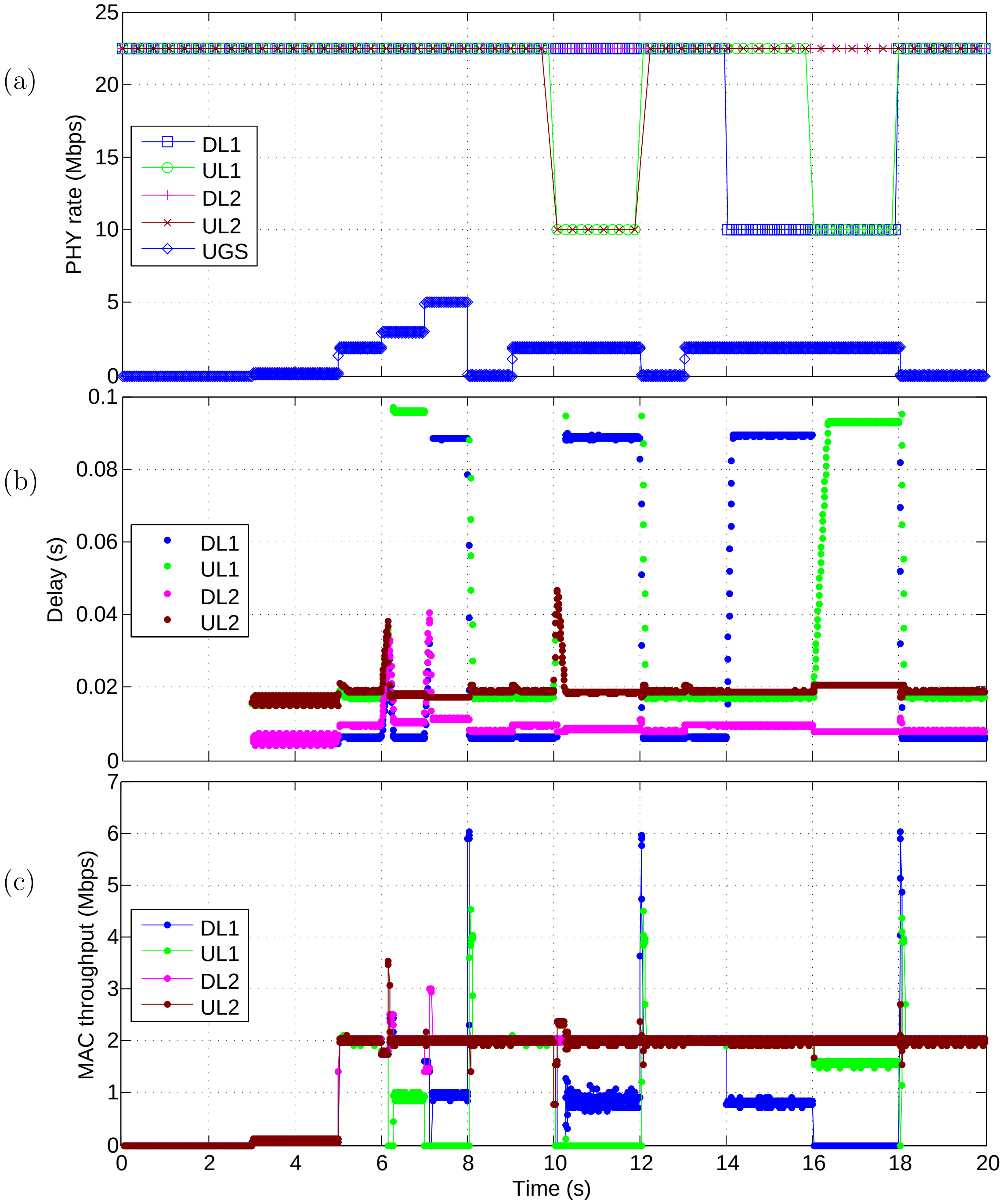

Figure 7.5.1: Partial coordination: (a) PHY rate, (b) packet delay, and (c) MAC throughput.

since they both belong to session V1.

At the β-queue, the scheduler allocates all left over slots, after serving the α-queue, to serve one connection from either the DL or the UL at the β-queue. In this case, connection DL1 is chosen to be served as it has the highest priority, and hence it experiences degraded service and connection UL1 experiences no service. Under this scheme, the number of bi-directional sessions that experience good service is maximised at 1 because both connections of session V2 experience good service, as shown in Figure 7.5.2 (c).

Region 1 ($6 \leq t \leq 8$) **– Full coordination**

Just after $t = 6$, a QoS Violation event is detected on the UL, which causes session V1, that is both connections UL1 and DL1, to be sent to their respective β-queues. As soon as this happens, there is an increase in the number of slots available to serve the DL and the UL connections at the α-queue. Hence, the MAC throughput of these connections at the α-queue remains at 2 $Mbps$, as shown in Figure 7.5.3 (c). At the β-queues, connections DL1 and UL1 are allocated slots evenly and hence, they achieve the same MAC throughput of 1.5 $Mbps$.

At $t = 7$, the number of slots available to serve the rtPS traffic decreases, which results in a decrease in the MAC throughput achieved by the connections at the β-queues. That is, the MAC throughput of both connections DL1 and UL1 decreases to 0.5 $Mbps$.

Since connections DL1 and UL1 are sent to their respective β-queues, they experience degraded service. Therefore, the number of bi-directional sessions that experience good service is maximised at 1 because both connections of session V2 experience good service, as shown in Figure 7.5.4 (c).

Region 2 ($10 \leq t \leq 12$) **– No coordination**

In this region, both sessions V1 and V2 are considered to be receiving degraded service since their UL connections experience either degraded service or no service.

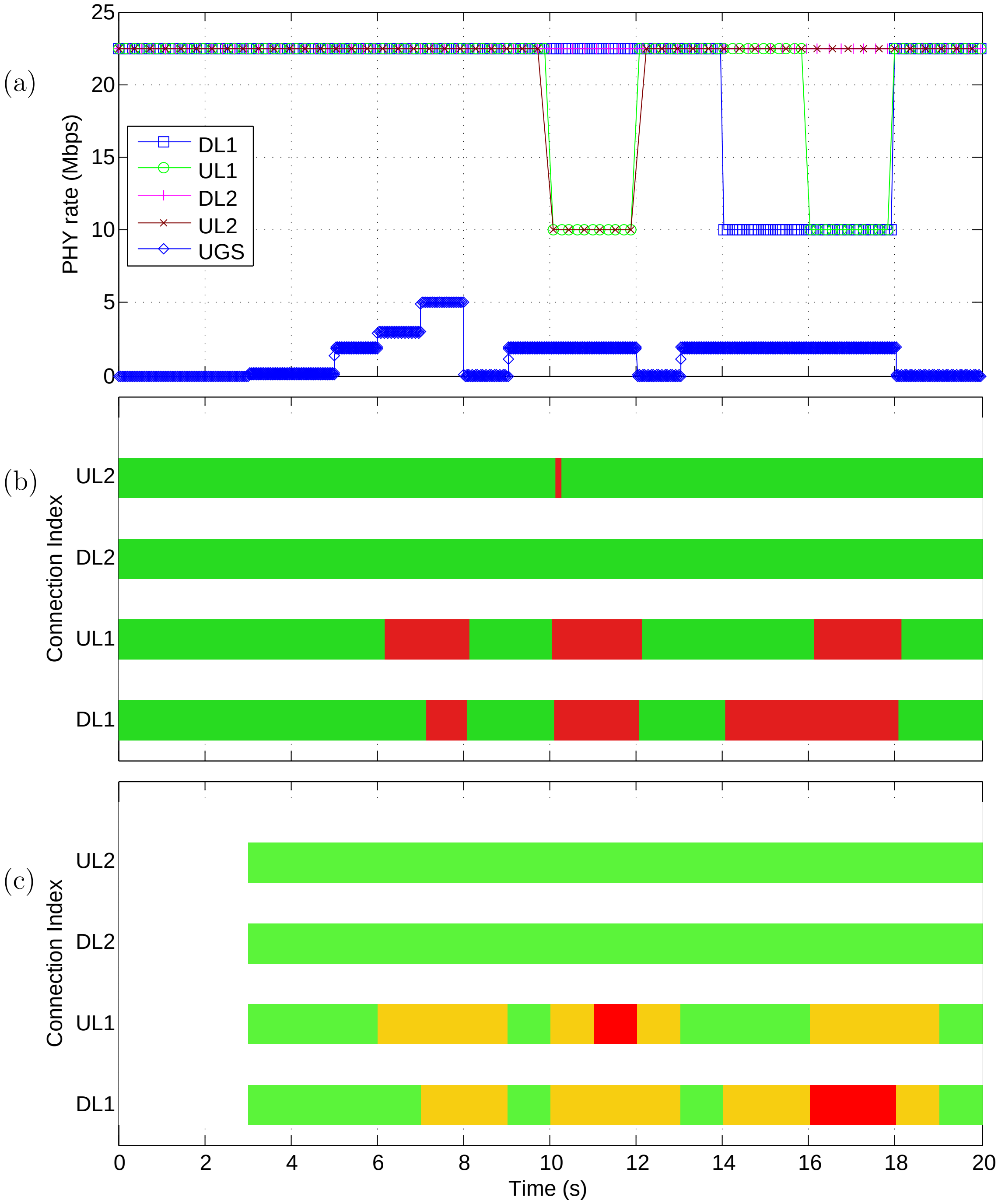

Figure 7.5.2: Partial coordination: (a) PHY rate, (b) connections at each queue; α-queue (green), β-queue (red), and (c) QoS received: good service (green), degraded service (amber) and no service (red).

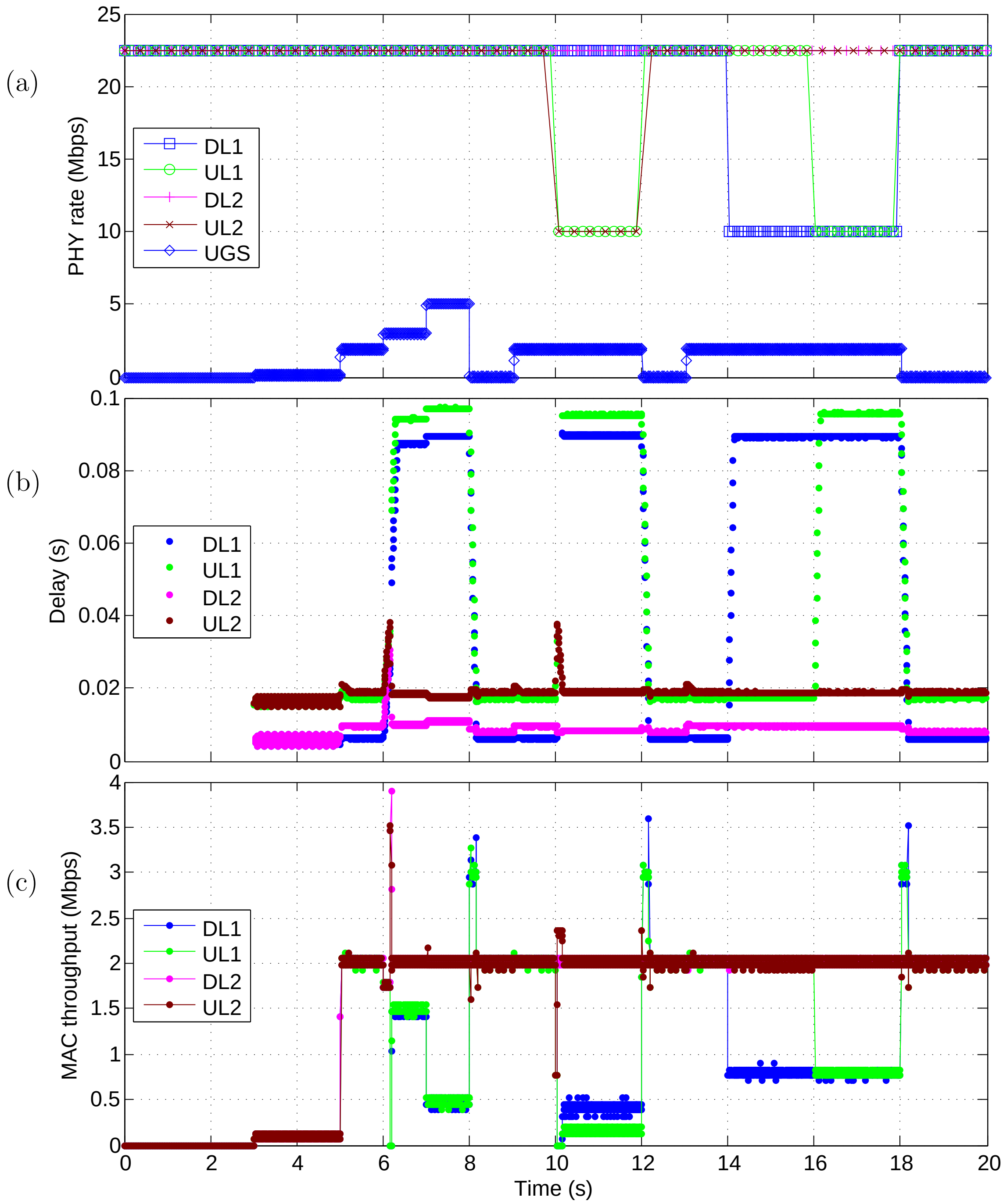

Figure 7.5.3: Full coordination: (a) PHY rate, (b) packet delay, and (c) MAC throughput.

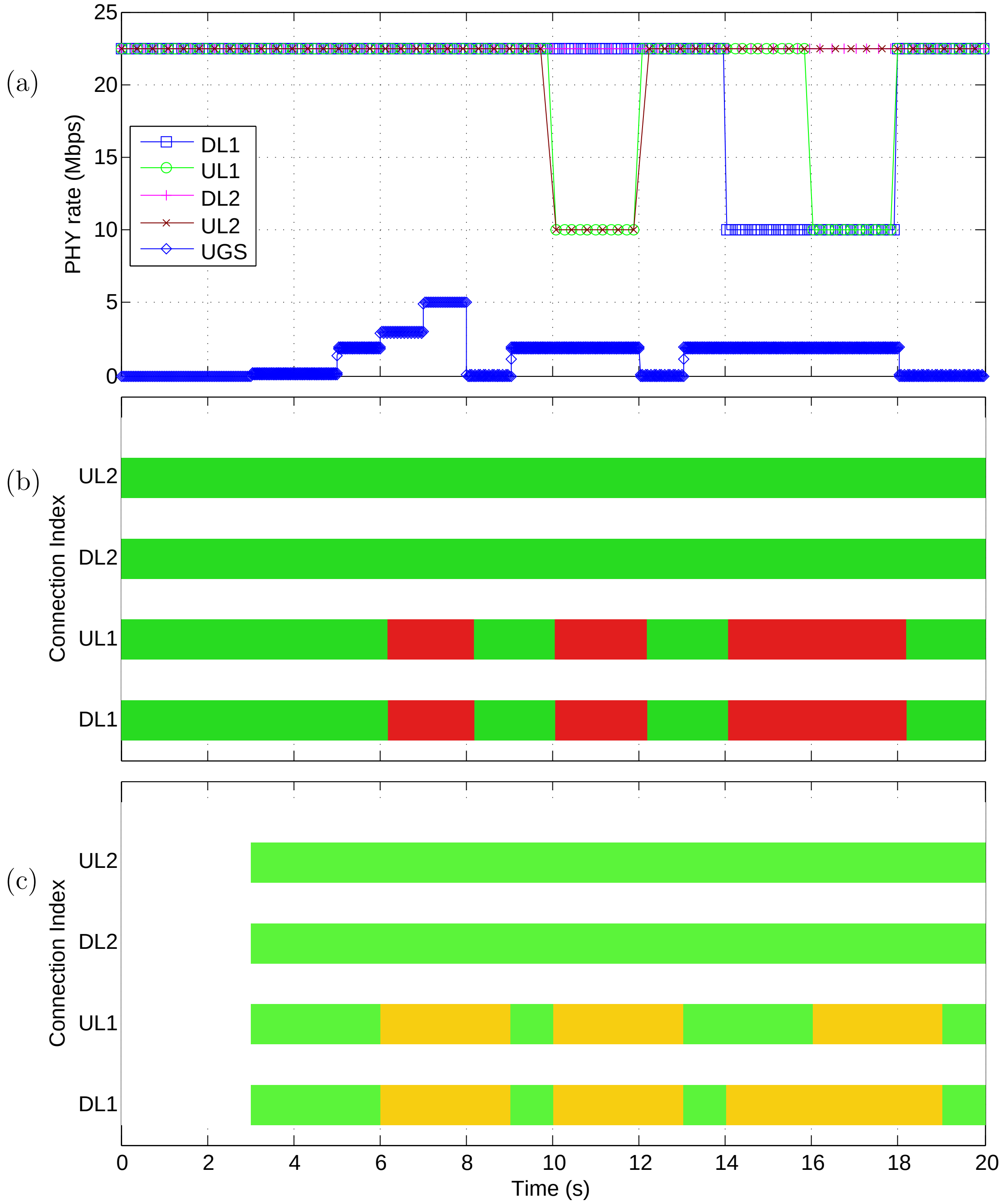

Figure 7.5.4: Full coordination: (a) PHY rate, (b) connections at each queue; α-queue (green), β-queue (red), and (c) QoS received: good service (green), degraded service (amber) and no service (red).

Region 2 $(10 \leq t \leq 12)$ **– Partial coordination**

From the actual data used for Figure 7.5.2 (b), we observe that the congestion in this region has caused connections UL1, DL1 and UL2 to be sent to their respective β-queues sequentially. Connection DL1 is sent to the β-queue due to a QoS Violation event that was detected on the UL, since there was no increase in delay experienced by the DL connections. Due to the more conservative approach taken in the QoS Violation Detection on the UL, a false QoS Violation event is detected on the UL, which causes connection UL2 to be sent to the β-queue for a few MAC frames before it is redirected back to the α-queue.

When connections UL1, DL1 and UL2 are at the β-queue, we observe that connection UL2 is chosen to be served, despite being a lower PHY mode connection, because it has the highest priority since it has its partner connection at the α-queue. From Figure 7.5.2 (c), we observe that connection UL2 experiences good service in this region even though it has been sent to the β-queue for a few MAC frames. This is because the increased delay experienced by connection UL2 is still below D_{req} when it is being served at the β-queue.

When there are only connections UL1 and DL1 in the β-queue, connection DL1 is served and hence it experiences degraded service and connection UL1 experiences no service. Therefore, the number of bi-directional sessions that experience good service is maximised at 1 because both connections of session V2 experience good service.

Region 2 $(10 \leq t \leq 12)$ **– Full coordination**

Under this scheme, congestion on the UL causes a QoS Violation event to be detected on the UL, which causes session V1, that is both connections DL1 and UL1, to be sent to the β-queue. At the β-queues, connections DL1 and UL1 are allocated slots evenly. Given the same number of slots between these connections, connection DL1 achieves a higher MAC throughput because it has a higher PHY mode. Since connections DL2 and UL2 experience good service, session V2 is considered to be receiving good service.

Region 3 ($14 \leq t \leq 18$) – **No coordination**

In this region, connections DL2 and UL2 experience good service, hence session V2 is considered to be receiving good service. Note that the number of bi-directional sessions receiving good service is maximised at 1 in this region due to the objective of maximising MAC throughput, because both connections of session V1 happen to have the lowest PHY mode.

Region 3 ($14 \leq t \leq 18$) – **Partial coordination**

Similar to the "no coordination" scheme, connection DL1 is sent to the β-queue when a QoS Violation event is detected on the DL at $t = 14$. There is also no impact on the average delay experienced by the UL connections due to congestion on the DL. At $t = 16$, a QoS Violation event is detected on the UL, which causes connection UL1 to be sent to the β-queue.

At the β-queue, connection UL1 is chosen to be served and hence it experiences degraded service and connection DL1 experiences no service. Therefore, the number of bi-directional sessions that experience good service is maximised at 1 because both connections of session V2 experience good service.

Region 3 ($14 \leq t \leq 18$) – **Full coordination**

Similar to the other regions, at $t = 14$, a QoS Violation event is detected on the DL due to the change in PHY mode of connection DL1, which causes session V1 to be sent to the β-queue. Notice that connection UL1 is receiving good service even though it is served at the β-queue. This is because the number of slots allocated to serve this connection at the β-queue is enough to maintain its offered load of 2 $Mbps$. However, connection DL1 is not able to achieve its offered load of 2 $Mbps$ because it has a lower PHY mode. At $t = 16$, connection UL1 changes to the same lower PHY mode as connection DL1. Hence, both connections DL1 and UL1 achieve the same MAC throughput of 0.7 $Mbps$, as shown in Figure 7.5.3 (c), and both experience degraded service.

7.5.2 Summary

In Table 7.5.1, we summarise our experimental results, where a tick indicates that the number of bi-directional sessions that experience "good service" is maximised. Recall, we defined a bi-directional session as receiving good service when both its DL and UL connections receive good service.

Table 7.5.1: Summary of the joint DL and UL schemes in terms of maximising the number of bi-directional sessions that experience good service.

Schemes / Experimental Regions	Region 1	Region 2	Region 3
No coordination	×	×	✓
Partial coordination	✓	✓	✓
Full coordination	✓	✓	✓

In our experiment, we have shown that the number of bi-directional sessions that experience good service may not be maximised with the "no coordination" scheme. Hence, this scheme is not desirable for handling bi-directional sessions, such as video conferencing and VoIP, where the QoS of a session is dependent on the QoS of both connections of the session. Note that the number of bi-directional sessions that experience good service is maximised with the "no coordination" scheme only in region 3, because both connections of session V1 happen to have the lowest PHY mode.

Under the "partial coordination" scheme, the number of bi-directional sessions that experience good service is maximised. When a QoS Violation event is detected on either the DL or UL, the connection with the lowest priority from either the DL or UL will be selected to be sent to the β-queue. At the β-queue, only a single connection with the highest priority on either the DL or the UL will be served. With this scheme, we observe an alternating sequence between the DL and the UL in sending connections to the β-queue and in redirecting connections back to the α-queue. This is due to the connection prioritisation mechanism that is first based on whether a connection has its partner connection in either the α-queue or the β-queue.

Under the "full coordination" scheme, the number of bi-directional sessions that experience good service is also maximised. Unlike the "partial coordination" scheme, both connections of a selected bi-directional session will be sent to the β-queue at the same time when a QoS Violation event is detected on either the DL or the UL. In order to redirect both connections of a bi-directional session on the DL and the UL at the β-queue back to the α-queue together, the number of slots left after serving the α-queue will be allocated equally to both connections of a bi-directional session to be served at the β-queue.

The objective to maximise the number of bi-directional sessions that experience good service is achieved in both the partial and full coordination schemes. Therefore, the choice between the partial and full coordination schemes depends on the interpretation of the performance requirement of bi-directional sessions by the network provider. In other words, is one direction receiving good service and the other direction receiving reduced service, better or worse than both directions receiving the same moderate level of service? This judgement may well depend on the actual application and the network provider's customers. Note that these coordination schemes are only possible because the BS has control of the DL-MAP and the UL-MAP, and the fact that we have been able to develop a ULDQ scheduler that is able to remotely manage the UL queues effectively.

7.6 Summary

In this chapter, we have showed that a joint DL and UL DQ scheduler is necessary to improve the number of one-directional connections that experience "good service" in an 802.16 network, which we called the "basic coordination" scheduler. This scheduler handles the DL and UL connections together and assigns different priority to these connections.

Further, we have extended our investigation to bi-directional connections, such as VoIP and video conferencing sessions. We have showed that the DLDQ and ULDQ schedulers must be coordinated to some extent in order to achieve the objective of maximising the number of bi-directional sessions that experience good service in both directions. We have proposed two possible schemes based on the basic coordination scheme, "partial coordination" and "full coordination", which are able to achieve this objective.

The main differences between the partial and full coordination schemes are the way connections are moved between the α-queues and the β-queues, and the way slots are allocated for connections at the β-queues. We contend that the choice of which scheme to implement depends on the interpretation of the performance requirement of the bi-directional sessions by the network provider.

Chapter 8

Conclusion and Future Work

In this thesis, we have presented our work on bandwidth allocation in IEEE 802.16 systems, from proposing a framework for MAC layer scheduling design, through to deploying the DQ scheduling discipline to provide enhanced QoS, in terms of maximised number of satisfied customers. This chapter summarises our findings and presents a discussion of future work.

8.1 Summary

In Chapter 2, we described the bandwidth allocation process by the MAC scheduler in IEEE 802.16 based systems. It is the responsibility of the MAC scheduler to provide QoS to the systems. We highlighted the key components of the PHY and MAC layers that affect the MAC layer scheduling process. We also discussed the different aspects of the MAC layer scheduling process, such as the concept of Connection ID, Bandwidth Request/Grant mechanisms and the creation of the maps (DL-MAP and the UL-MAP) for describing the bandwidth allocation and usage of the system.

Based on the MAC and PHY features, we described the MAC scheduling process as a process to allocate slots to all the connections in an 802.16 system. Further, we have discussed the 5 CoSs defined in the 802.16 standard. Each of these CoSs has different QoS requirements. However, the actual scheduling algorithms for providing

QoS to these CoSs are not specified in the 802.16 standard.

There are altogether four criteria that we identified as important for an 802.16 MAC scheduler: CoS differentiation, dynamic sub-frame partitioning, SS differentiation and support for AMC. We examined related work according to their support for these criteria and found that the focus of the related work was on providing CoS differentiation. To the best of our knowledge, none of the related work has considered SS differentiation, which we addressed in conjunction with dynamic sub-frame partitioning and support for AMC.

In Chapter 3, we described the 802.16 MAC layer scheduling process as an optimisation process. While past work has focussed heavily on the scheduling question as it pertains to CoS differentiation, we approached the problem from a general perspective, in which the goal of the scheduling process is defined in terms of a set of objectives.

We proposed a framework that considers the set of objectives selected in the MAC scheduling design. The framework also meets the four criteria that we identified in Chapter 2. The set of potential objectives that we illustrated in that chapter is only a subset of all possible objectives. If there are multiple objectives involved, these objectives are considered in a hierarchical order.

Our scheduler design framework simultaneously considers any requirements from both the network provider and the customers that affect the scheduling process. Further, we included SS differentiation into our scheduler design framework. This feature allows the network provider to target a certain group of SSs or customers. We used a simple SS differentiation scheme as an example, by assigning gold or silver class to SSs, and demonstrated the importance of SS differentiation through this example. We also demonstrated the flexibility of our framework through examples that highlight how a service provider might achieve different throughput and revenue goals based on different objectives. Finally, we discussed the issue of achieving customer satisfaction in the network that leads us to an interesting objective, which is the objective of maximising the number of satisfied customers or SSs.

In Chapter 4, we presented a discussion about approaches for achieving customer satisfaction. We began by providing an introduction to customer satisfaction. Customer satisfaction is a highly subjective and abstract concept. It is also complex to define, and difficult to compare customer satisfaction across different individuals and different products. In some studies, the benefits of achieving customer satisfaction found include: increased customer loyalty, increased customer retention, improved market share and increased profitability; it is therefore a major goal of network providers to achieve customer satisfaction.

There is strong evidence showing a firm causal linkage between customer satisfaction and service performance. Hence, we contended that the network provider should provide good service performance in order to achieve customer satisfaction in telecommunication services. Service performance in the telecommunications field is affected by customer service and network performance. Therefore, we further investigated improving network performance during periods of congestion, in order to achieve better customer satisfaction. We then proposed to employ the DQ scheduling discipline to maximise the number of satisfied customers.

Chapter 4 also presented a detailed analysis of the DQ scheduling discipline in a wired environment. The DQ discipline aims at maximising the number of satisfied customers, as opposed to providing fairness to the customers. We generalised and identified 6 core DQ mechanisms of the DQ discipline; they are connection prioritisation, QoS Violation Detection, response to QoS Violation Detection, QoS Recovery Detection, response to QoS Recovery Detection and Explicit Packet Dropping. We discussed each of these core DQ mechanisms in detail. Towards the end of Chapter 4, we explained in detail the reasons why we could not directly deploy the original DQ algorithms from Hayes *et al.* into the 802.16 environment.

Chapter 5 presented the core of our work on bandwidth allocation using the DQ scheduling discipline. We proposed a DQ implementation in 802.16 systems for handling real-time services in both the DL and UL. We showed the required framework for the implementation, which addressed issues that include the structure of the α-queues and the β-queues, handling Automatic Repeat Request traffic and the service scheme deployed by the DQ scheduler.

We considered the DLDQ and ULDQ implementations separately. We first proposed the core DQ mechanisms for the DL. We then conducted experiments to show that our DL proposal is able to cope with the challenges associated with the changing bandwidth available to serve the DQ traffic. These challenges stem from the unpredictability of the slots allocation to the DQ traffic and due to the support for AMC. Similar to the DL, we proposed the core DQ mechanisms for the UL, which we claimed to be more complicated. This is because the BS needs to acquire Bandwidth Request information from the SSs for the UL traffic, rather than having the information available locally. We also conducted experiments and showed that our UL proposal has similar performance to the DL. In fact, we contend that a QoS Violation event can be detected earlier on the UL. This is due to two factors: a higher average delay is experienced by the UL connections and a conservative approach is deployed in the UL QoS Violation Detection algorithm.

Further, we compared the performance of our entire DQ scheduler to a WFQ scheduler. We also conducted a comparison between our DQ scheduler and an EPD enhanced WFQ scheduler. In addition, we included a discussion of the impact of bit errors and the use of ARQ. From the experimental results, we showed that our proposed DQ has a much better QoS performance in terms of maximising the number of connections that experience "good service" at the expense of sacrificing the QoS of lower priority connections. However, we showed that the degraded QoS experienced by the lower priority connections is small, compared to the improved QoS experienced by the higher priority connections. Therefore, the DQ scheduler is able to provide good service to many more connections simultaneously than the WFQ scheduling discipline.

On the other hand, we showed the importance of the EPD mechanism in the DQ and the EPD enhanced WFQ schedulers. With the EPD mechanism, a scheduler can avoid wasting transmission slots on packets that have exceeded their delay requirements, in order to give more transmission opportunities to packets that have the potential to meet their delay requirements. This results in a lower system utilisation being achieved in the DQ and the EPD enhanced schemes. This is a desired outcome as unused bandwidth can be allocated to other services.

Lastly in Chapter 5, we discussed the impact of deploying the DQ scheduler on the total MAC throughput. We found that the highest total MAC throughput is not achieved by the DQ scheduler because the system is not fully utilised at all times. We contended that a better MAC throughput measure is the MAC throughput that meets the QoS requirements. Hence, we compared the total MAC throughput that represents packets experiencing "good service" and showed that the highest total "good service" MAC throughput is achieved by the DQ scheme. Furthermore, we showed that our DQ system can handle mixed traffic profiles, such as MPEG-4 video and G.729B VoIP. We also showed that our DQ system is able to work well under different non-zero BER conditions.

In Chapter 6, we investigated alternative objectives that the DQ scheduler may try to achieve, as opposed to primarily maximising the number of connections that experience "good service". We proposed the PBDQ scheduler in order to support this feature. With the PBDQ scheduler, connections that belong to the same priority class are handled in separate DQs. Furthermore, connections that belong to a higher priority class are served ahead of connections that belong to lower priority classes at all times, in systems where connections may change their priority classes. We showed an example where the throughput of a network is maximised at all times by using the PBDQ scheduler.

In Chapter 7, we investigated a joint DL and UL Dual-Queue scheduler that is able to maximise the number of one-directional connections that experience "good service" in an 802.16 network, which we called the "basic coordination" scheduler. This scheduler handles DL and UL connections together and assigns different priorities to these connections.

We then extended our investigation to bi-directional traffic sessions, such as

VoIP and video conferencing, which involve both DL and UL connections. We showed through an example that the DLDQ and the ULDQ schedulers must be coordinated to some extent in order to achieve the objective of maximising the number of bi-directional sessions that experience "good service". Based on the basic coordination scheme, we proposed two possible schemes, "partial coordination" and "full coordination", which are able to achieve this objective.

The main differences between the two schemes are the way connections are moved between the α-queues and the β-queues, and the way bandwidth is allocated for connections at the β-queues. We contended that the choice of which scheme to implement depends on the interpretation of the performance requirement of the bi-directional sessions by the network provider.

8.2 Potential Research Extensions

8.2.1 Bandwidth Request Polling Interval

In our research, we focused on real-time polling service (rtPS) traffic as the DQ traffic. On the UL, we assumed the Bandwidth Request (BR) mechanism for this traffic is carried out on a per MAC frame basis, which corresponds to a polling interval of a MAC frame duration. In real 802.16 systems, the BR mechanism may not be carried out in a periodic fashion, in order to save transmission bandwidth. Furthermore, we have seen the impact of the polling interval on the delay of the UL traffic. Therefore, further research on the impact of a variable BR polling interval on the delay of the UL traffic being maintained by the ULDQ scheduler would be beneficial. Hence, further research work on extending the UL QoS Violation Detection algorithm with variable BR polling interval, or deriving a totally new UL QoS Violation algorithm, could be undertaken.

8.2.2 Connection Prioritisation Schemes

With homogenous traffic connections, which refer to connections that have the same data rate and PHY mode, maximising the number of connections with "good service" is an inherent objective from the deployment of the DQ discipline. When a QoS Violation event is detected, an arbitrary connection can be chosen from the α-queue to be sent to the β-queue without violating the objective to maximise the number of connections at the α-queue at all times since all the connections are treated equally. However, with inhomogeneous traffic connections, that is, connections may have different data rate or PHY mode, the data rate and PHY mode information for each connection has to be available to the DQ scheduler in order to maximise the number of connections at the α-queue at all times. For example, let us assume all the connections have the same PHY mode. If there are two low demand connections with a total data rate that equals the data rate of a single high demand connection, the DQ scheduler should prefer to have two low demand connections than one single high demand connection in order to maximise the number of connections at the α-queue. That is, low demand connections are given a higher priority than high demand connections.

Next, let us consider a more complicated case where connections may have different data rates and PHY modes. For instance, a low demand and low PHY mode connection is not necessarily less preferred over a high demand and high PHY mode connection. Hence, a tie breaker may be based on the average number of slots required for each connection, which can be influenced by both the data rate and PHY mode of the connection in real-time.

This leads us to another interesting research field, which is prediction of the average bandwidth required for each connection in the network. Based on this bandwidth information, each connection could be given a weight or priority, which is used in the scheduling process. This provides a new dimension for the connection prioritisation mechanism that helps to maximise the number of connections with "good service" at all times. Hence, further research on incorporating differentiation of connections' bandwidth demand as a criterion for connection prioritisation would be beneficial. Investigation following [54] that proposed a method for estimating the bandwidth of rtPS connections could be a good starting point.

Further, the objective to maximise the number of connections that experience "good service" considered in this thesis is based on information obtained from the current MAC frame. Therefore, the DQ scheduler only ensures that the number of connections that experience 'good service" is maximised in every MAC frame. However, it is not necessary that the same set of connections are selected to experience "good service" in every MAC frame. This is because connections may become active or inactive, and their priority may also change dynamically. For example, the DQ scheduler can be configured to maximise the number of connections which only receive "good service" in a 10 minute interval. During this 10 minute time interval, assume that there are 10 active connections in a network and the number of connections that can be accommodated at the α-queue varies between 5 and 10. The DQ scheduler must achieve a total of 5 connections with "good service" throughout the entire 10 minute interval in order to maximise the number of connections which only receive "good service" in the 10 minute interval. If this criterion is not important, the DQ scheduler might achieve the objective of maximising the number of connections at all times such that there are less than 5 connections experiencing "good service" for the entire 10 minute interval. Hence, further work can be carried out to investigate the connection selection criteria to maximise the number of connections only receiving "good service" in a pre-defined time interval. One of the possible solutions is based on the historical scheme found in Hayes work [2].