Chapter 5

RPS: A Real-time Packet Scheduling Scheme for TDMA MAC

5.1 Introduction

Link level transmission opportunities provided to nodes may not always converge to QoS provisioning. Having a high volume of traffic to transmit, a node must utilize its transmission opportunity judiciously in order to provision QoS for certain traffic flows. To address this issue, A packet level scheduling scheme is proposed in this chapter. The proposed scheme provisions a fine-tuned QoS for different types of real-time applications.

Link level scheduling schemes usually provide mechanisms for scheduling access to shared medium efficiently in order to improve overall network performance. In such schemes, QoS can be provisioned to real-time applications through prioritizing their access to the medium. However, this can only assure a prioritized channel access to the shared medium but not the transmission of packets belonging to a prioritized flow. When traffic load exceeds beyond the channel capacity, packets do not get chance for transmission and are buffered for later transmission opportunity. Hence, link level scheduling do not always guarantee QoS for real-time traffic in high load situations. Thus, QoS cannot be guaranteed using link level scheduling alone. In order to meet the QoS requirements of different real-time packets, transmission opportunities assigned to a node need to be utilized according to the different QoS requirements of real-time traffic. Therefore, a packet level scheduling can greatly improvise QoS issues in capacity-constrained WiLD networks by utilizing the link level transmission opportunities provided.

Providing a finer level of QoS to real-time voice and video based applications in WiLD networks warrants a timely and accurate reservation of transmission slot for their packets. Classifying traffic into few priority categories based on the QoS requirements, many protocols [?, ?, ?] propose to provide QoS to higher priority traffic. Admission control mechanisms [?, ?] are also common in QoS research for maintaining quality of the ongoing flows. But, providing a fine-tuned QoS guarantee for real-time traffic through precise TDMA scheduling is still a challenging issue.

In this chapter, we have proposed a *Real-time Packet Scheduler* or *RPS* in short for TDMA-based MAC protocols. It is an integrated approach combining a localized admission control and a flow based anticipatory packet scheduling scheme. Classifying the real-time traffic into three specific categories according to their delay and bandwidth requirements, the scheduling scheme ensures a fine grained QoS guarantee for the flows over TDMA-based MAC protocol in WiLD networks. RPS schedules the real-time packets anticipating their arrivals based on their periodicity. In order to sustain the QoS guarantees of the active flows, the protocol relies on a localized call admission process which can locally take decision about admission of a new flow rather than involving all the nodes in the routing path.

The rest of this chapter is organized as follows. Section ?? takes a look on the related works. A comparison of different packet level scheduling approaches is presented in Section ??. The classification framework used for categorizing different traffic according to their QoS requirement is explained in Section ??. Section ?? describes the proposed protocol. Delay analysis of the proposed packet scheduling scheme is discussed in Section ??. In Section ??, performance evaluation of the proposed protocol and comparison with 2C protocol are presented. Finally, Section ?? provides the conclusion to this chapter.

5.2 Related Works

Provisioning QoS in a packet switched network demands proper use of traffic scheduling. The main function of QoS scheduling is to schedule the packets meeting the QoS demands of various applications. The responsibility of traffic scheduling algorithm is to decide which traffic flow should be served and which should be discarded. Thus, it is possible to ensure QoS by using prioritized traffic scheduling approach. In TDMA-based MAC protocols, the main goal of any packet scheduling algorithm is to maximize the slot utilization and serve request for different classes of traffic coming from different applications and users. To achieve this, we need to have a proper classification of traffic according to their QoS requirements and a scheduling algorithm which can schedule different classes of traffic in the allotted TDMA slots maintaining the required priority. Admission control mechanisms are often used to provide QoS guarantees to the active flows in a system.

The traditional packet scheduling schemes use First-Come-First-Served (FCFS) policy which processes packets according to their arrival time. This kind of scheduling does not offer any priority to any packet. Hence, these schemes are not efficient in supporting throughput and delay-bound real-time applications. Weighted Fair Queuing (WFQ) [**?**] is a flow-based queuing algorithm that schedules low-volume traffic first, while letting high-volume traffic share the remaining bandwidth. This is handled by assigning a weight to each flow, where lower weights are the first to be serviced. Many deadline-based packet scheduling mechanisms such as Earliest Deadline First (EDF) [**?**] are also proposed in the literature. In this approach, each arriving packet is assigned a deadline. The packets are scheduled in the order of deadline.

IETF has developed two of the main QoS models for the Internet. Integrated Services (IntServ) [**?**] can provide fine-grained per-flow guarantees to the real-time applications. However, as a reservation-based approach, it puts high load on packet classifiers and schedulers. For large number of flows running in a system this overhead cannot be neglected. Differentiated Services (DiffServ) [**?**] provides differential treatment to packets of different traffic classes but it requires aggregation of packets at the destination [**?**].

A significant amount of prior work done in packet level scheduling to provide QoS in terms of delay, throughput, and jitter in TDMA-based MAC protocols are found in the literature. Normally, to meet QoS demands of different applications, traffic are classified into a few categories based on their QoS requirements. In JaldiMAC [**?**], traffic is classified into two categories; L class (Latency Sensitive Traffic) and B class (Bulk Traffic). The L class traffic are given higher priority over B class in order to meet their QoS demands. Zhao et al. [**?**] classifies traffic as high, normal and best-effort. The high priority traffic such as interactive real-time audio/video applications are strictly delay sensitive in nature. The normal priority traffic such as stored audio/video has relaxed delay bounds. The best-effort traffic has no specific delay bound. [?] proposes a 3-level priority packet scheduling algorithm for WSN. The authors considered three different priority classes- (i) real-time (highest priority), (ii) non-real-time remote packets, i.e., packets that arrive remotely located sensor nodes (medium priority), and (iii) non real-time local packets, i.e., the packets that are generated by the current sensor node (lowest priority). Riggio et al. [?] proposes such a scheme to improve QoS by combining service differentiation and packet aggregation in IEEE 802.11-based wireless networks. This scheme classifies the traffic into four broad categories: Best-effort, Low, Medium and High. [?] considers two types of traffic: delay sensitive and delay insensitive. While scheduling the packets, the packets whose delay is already higher than the acceptable delay of $150ms$ are dropped, and those packets whose delay is well within the acceptable delay limits are transmitted according to their descending experienced delay.

Real-time Flow Scheduling (RFS) [?] proposed an admission control scheme where the source node determines whether the new flow can be admitted without missing deadlines of any existing flows. A cross layer admission control proposed in [?] uses a route discovery process from source to destinations and each intermediate interface takes part in the decision making whether a new call admission is possible or not. DelayCheck [?] proposes an online centralized scheduling which schedules constant-bit-rate voice traffic in TDMA-based mesh networks. It targeted maximizing the number of voice call support in wireless mesh networks. This protocol further incorporated an admission control mechanism with DelayCheck.

Different real-time applications have varying QoS requirements in terms of bandwidth, delay, jitter, packet-loss, etc. Further, the factors such as packet arrival rate, deadline of packet, packet size, etc., are also not same for all real-time applications. The scheduling mechanisms discussed above bundles a number of applications with varying QoS requirements into a few priority classes. In this context, a fine grained QoS can be achieved by a more specific classification of real-time traffic. In our proposed scheme, we consider four classes of traffic based on their QoS requirements. It provides a very precise (slot-level) TDMA schedule for the packets belonging to real-time flows. A localized admission control scheme is augmented to the proposed scheme which can independently admit flows without involving the nodes of the entire end-to-end path in the decision making process. In doing so, it alleviates the QoS performance issues of high priority flows in the network.

5.3 Comparison of Existing Packet Scheduling Schemes

The main principle behind packet scheduling is to categorize the traffic into different classes and assign priorities to these based on their QoS requirements. A relatively large number of packet scheduling schemes are available in the literature. JaldiMAC [?] employs packet differentiation technique by dividing traffic into delay sensitive and bandwidth class. An integrated routing and MAC scheme is proposed in [?]. The MAC protocol divides the traffic into high, normal and best-effort priority classes and defines a traffic service index accordingly. The service index is used by the scheduling algorithm to prioritize traffic. DelayCheck [?] proposes a centralized scheduling and call admission control mechanism for provisioning QoS to VoIP applications. Few important packet scheduling schemes have been discussed in Section ?? of Chapter ?? (page ??). Table ?? on page ?? provides a point-wise comparison of various packet scheduling schemes.

5.4 Classification of Traffic

Taking the delay and bandwidth requirements of various applications into consideration, we classify traffic into four different categories which are as follows-

- **Strict Delay sensitive and Fixed Bandwidth (SDFB)**:

  Real-time symmetric traffic with strict delay bound and constant bandwidth demand are categorized into this traffic class. The periodicity of this class of traffic is uniform. We assign 1^{st} priority to this type of traffic. The International Telecommunications Union (ITU) standard codec G.711.1 with bit rate of $64Kbps$ for VoIP is an example of this type of traffic class. G.711.1 codec is commonly used for best voice quality [?]. The important parameters of G.711.1 codec has been discussed in section ??.

- **Strict Delay sensitive and Variable Bandwidth (SDVB)**:

  Real-time symmetric traffic that require variable bandwidth with respect to time but periodic in nature are included in this traffic category. Video conferencing and videophony belong to this category which produce highly delay sensitive and bandwidth-greedy traffic. This type of traffic is treated as having priority 2. H.323 codec allows video conferencing over the Internet which was developed by the ITU. Using H.323, a high-quality video conference (excellent audio and video) needs about $768Kbps$ of bandwidth on a packet switched network [?, ?].

- **Delay Sensitive and Minimal Bandwidth (DSMB)**:

  Applications requiring minimum bandwidth guarantee but delay sensitive in nature are included in this traffic class. For this category of traffic, the delay bounds are less strict than 1^{st} and 2^{nd} priority classes. Therefore, the traffic flows generated by this category of traffic require bandwidth guarantees without concerning much about delay bound. The traffic belonging to this class is asymmetric in nature. That means, the amount of upward and downward traffic flows are not the same. Examples of such applications are video streaming, video broadcast, audio streaming, etc. We consider this type of traffic as 3^{rd} priority. H.264 video codec is used to stream high quality video over the Internet. H.264 has a mean frame size of $3832bytes$ but after fragmentation, the resultant average MAC service data unit (MSDU) size becomes $1250bytes$ [?].

- **Best-Effort (BE)**:

  This type of traffic or applications have minimal or no QoS requirement in terms of delay, bandwidth, jitter or any other metric. Traffic generated by HTTP, FTP, E-mail, and other similar applications are included in this category.

QoS requirements of VoIP, video conferencing and streaming video and their traffic characteristics are presented in the Tables ?? and ?? of Chapter ??.

5.5 RPS: The Proposed Protocol

The proposed protocol ensures QoS for higher priority traffic through provisioning of timely and more transmission opportunities compared to the lower priority ones. We categorize the real-time traffic into four priority classes as discussed in Section ??. A 2-Phase localized admission control scheme is employed to ensure that the required resources both at link and slot level are available prior to the admission of a traffic flow. Finally, a packet scheduling algorithm schedules the traffic anticipating their arrivals. Arrivals of packets are anticipated from the periodicity of the traffic priority classes.

For the sake of simplicity, let us assume that traffic is divided into m priority classes where rate of packet generation varies for different priority classes. Let us further assume that the packet generation rate of i^{th} traffic class TC_j is represented by $TC_j.g$. The proposed scheduling scheme maintains separate queues to buffer the packets belonging to different category of traffic flows as follows. The packets of highest priority traffic flows i.e., SDFB are buffered in a Low Latency (LL) queue. A High Bandwidth and Low Latency ($HBLL$) queue is used to buffer the next higher priority i.e., SDVB traffic. Traffic demanding high bandwidth i.e., DSMB are buffered in a Bandwidth (B) queue. The lowest priority i.e., best-effort traffic are buffered in a Default queue, D. The buffered packets in various queues are served according to the order of their priority.

Every WiLD link has got a maximum transmission capacity, say LC_{max}. A node requesting admission of a new flow generates a flow request, FR_k specifying its QoS requirements. Each flow is assigned a unique identification number irrespective of its priority class. The information about all the active flows are maintained in a register called *flow register, $FReg$*. The total reserved bandwidth for all the registered flows available in $FReg$ is represented as $FReg.bw$. To anticipate the packet arrival time of a particular flow, RPS uses a parameter called *I-value*. The room for scheduling transmission of an anticipated packet in a given slot is determined by using another parameter known as *K-value*. The maximum number of trials in scheduling packets of a flow is restricted to a limit called max_retry.

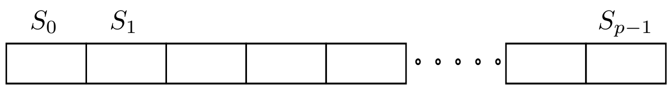


Figure 5-1: A TDMA Frame with p number of Slots

Consider that there are p number of time slots in a TDMA frame. The set of time slots in a TDMA frame, S can be given by $S = \{S_0, S_1, S_2, ..., S_{p-1}\}$ where $S_0, S_1, ..., S_{p-1}$ are the time slots. A TDMA frame consisting of p time slots is pictorially explained in Figure **??**. For the sake of simplicity, only a contiguous block of data slots is shown. For the same reason, guard times between the consecutive slots are not shown in the figure. To ensure assured packet transmission, more than one packet shall not arrive from the same flow in a particular time slot. Therefore, time slot of less than the smallest traffic periodicity ($20ms$ for VoIP) is required to be used. As discussed in 2C (Chapter **??**), we consider that every alternate TDMA slot is assigned to a node for its transmission. A flow is assigned a flow identifier, f_i ; $0 \leq i < n$ where n is the maximum number of flows that can be registered in $FReg$. The bandwidth demand of a flow f_i is represented as $f_i.bw$. Every newly admitted flow is associated with a variable $f_i.tr$ which keeps track of the number of scheduling retries.

5.5.1 Two Parameters: I-value and K-value

In the proposed protocol, we use two parameters- *I-value* and *K-value* to determine the schedule-ability of packets belonging to certain category of traffic flows. The methods of calculating these two parameters are explained below.

Estimation of I-value

Based on the periodicity of traffic, every priority class TC_j is assigned an I-value which is denoted as $TC_j.\alpha$. I-value is the time period during which a flow, f_i belonging to priority class TC_j remains in inactive state. An inactive period of a flow is the time elapsed during which a flow does not generate any packet, i.e., remains inactive. The concept of active and inactive time period of a flow is demonstrated in Figure **??**. It is expressed in terms of number of TDMA time slots. Since, alternate transmission slots are used for transmission, the maximum I-value for a priority class, TC_j represented as $TC_j.\alpha$ can be derived by using the Equation (**??**).

$$TC_j.\alpha \leftarrow (TC_j.g/(2 \times T_s)) \tag{5.1}$$

Where, $TC_j.g$ and T_s represent the packet generation rate of traffic class TC_j and size of a TDMA slot respectively.

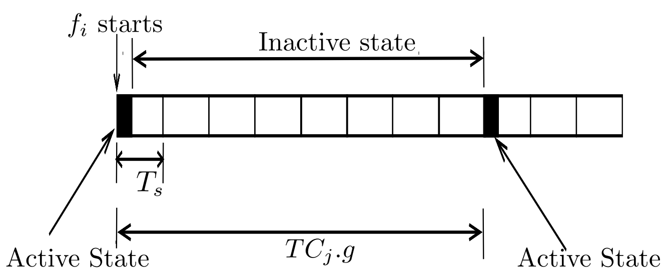


Figure 5-2: Active and Inactive States of a Flow in RPS

While admitting a flow, f_i belonging to priority class TC_j to the flow register $FReg$, it is assigned a flow-I-value denoted as $f_i.\alpha$. The initial value for

$f_i.\alpha$ is given as $TC_j.\alpha + 1$. After each slot, the value of $f_i.\alpha$ corresponding to all registered flows are decremented by 1. Before checking the flows to be scheduled, the value of $f_i.\alpha$ is checked for all the flows available in $FReg$. If the value is found as 1, the process further checks for the *K-value* for slot scheduling.

Estimation of K-value

The estimation of *K-value* depends on the remaining usable time in the current TDMA slot which is denoted as t_r. This parameter is initialized to the value of a slot size, T_s. When a flow with required transmission time t_a is scheduled, the value of t_r is updated using Equation (**??**).

$$t_r = t_r - t_a \tag{5.2}$$

The *K-value* for a given flow of priority class TC_j is calculated as

$$k = \frac{t_r}{t_a} \tag{5.3}$$

If the *K-value* is greater than or equal to 1, the packets of the flow f_i are scheduled; otherwise, the packets are stored in their respective queues and waits for the next opportunity. During its waiting state, it keeps on updating the value of t_r by using Equation (**??**).

5.5.2 Link-Slot (LS) Localized Admission Control

In multi-hop WiLD networks, the effective link capacity of each node gets constrained by the number of children nodes a parent has. In fact, the traffic forwarding capacity of a parent node is shared among its children. In such resource-constrained situations, certain packets need to be given higher priority than others in order to provide QoS. Admission control is an important mechanism to provide some level of QoS guarantees to the ongoing flows. The main goal behind admission control is to either accept or reject a new flow request based on the network's ability to meet the QoS demands of the application. If all the required parameters of a flow request can be met, the admission control mechanism admits the flow; otherwise not.

As the name suggests, the proposed admission process is carried out locally at a node in two phases. In the first phase, a node carries out a *link level* admission control followed by a *slot level* admission control. A flow is registered only when it passes through both the phases successfully. In link level admission control, if the requested flow can be accommodated within the available link bandwidth then the flow is tentatively added to the flow register assigning a unique flow identifier to it. Otherwise, the flow is rejected. Once a flow is added to the register, it is marked as new flow. In the next step, it accommodates the packets of already registered flows. Then, the protocol enters the second phase i.e., slot level admission control. Slot level admission control checks for available room for admitting newly registered flows in the current time slot. If room found then the packet is scheduled in that slot and the flow is mark as registered (final). The packet belonging to the registered flows are scheduled for transmission. Otherwise, the flow is rejected. These two steps can be summarized as follows-

- Phase 1 *(Link Level Admission)*:-
  - If the requested flow can be accommodated within the available link bandwidth then add the flow in the flow register tentatively assigning a unique flow identifier to it, mark it as new flow and go to phase 2.
  - Else reject the flow.
- Phase 2 *(Slot Level Admission)*:-
  - After accommodating the packets of already registered flows, if there is any room for admitting newly entered flows in that slot then schedule the packet and register the flow as final.
  - Else reject the flow.

Algorithm **??** demonstrates the 1^{st} phase i.e., the link level admission process of the LS admission control scheme. The algorithm first checks whether the sum of already committed bandwidth $FReg.bw$ and the requested bandwidth, $f_i.bw$ exceeds the total link capacity, LC_{max}. If not then it generates a flow identifier, f_i for the flow request FR_k and checks the priority class of f_i. If the flow belongs to priority class TC_j, then $f_i.\alpha$ is assigned the value $TC_j.\alpha + 1$. The procedure *Check New Flows* (Algorithm **??**) explains the 2^{nd} phase of the protocol i.e., the slot level admission process. For each newly registered flow, the algorithm first calculates the *K-value* using the Equation (**??**). If the *K-value* is greater than or equal to 1, the algorithm schedules the packet, updates t_r, re-initializes the value of $f_i.\alpha$ to $TC_j.\alpha$, and then confirms the flow. Otherwise, the algorithm discards the flow for the time being and retries to admit it until $f_i.tr$ exceeds max_retry.

Algorithm 5 Link-Slot (LS) Admission Control Algorithm

Input:
$FReg$: Flow Register
FR_k: Flow Request
LC_{max}: Maximum Link Capacity

if $FReg.bw + f_i.bw \leq C$ **then**
    $f_i \leftarrow$ Assign a flow-id for FR_j
    if $f_i \in TC_j : 0 < j \leq m$ **then**
        $f_i.\alpha \leftarrow TC_j.\alpha + 1$
        $f_i.tr \leftarrow 0$
        Register f_i in $FReg$ tentatively
    end if
else
    Reject the flowf_i
end if

5.5.3 Flow-based Anticipatory Packet Scheduling

In TDMA-based MAC, every node is assigned one or more time slots in each TDMA frame. The assigned time slot is a bounded time interval during which a node can transmit as many packets as possible. The main concern of QoS scheduling is to prioritize network traffic in such a way that traffic with more stringent QoS requirements are given higher priority and hence get earlier or better chance of transmission. To meet the QoS requirements of heterogeneous traffic classes, RPS smartly schedules the outstanding packets with different QoS requirements within their respective time intervals allocated to them. A delay-sensitive traffic with higher priority gets scheduled earlier than a delay-tolerant traffic. On the other hand, bandwidth-bound traffic are not scheduled earlier rather a larger share of bandwidth is provided.

If the start time of a flow f_i is δ_i, the time slot number to be used for scheduling the r^{th} packet, denoted as δ_r is given by

$$\delta_r = \delta_i + r \times TC_j.\alpha$$

The proposed flow based scheduling algorithm provides a QoS-aware schedule for prioritized slot utilization. The Procedure **??** (Check_New_Flows)

Algorithm 6 Packet Scheduling Algorithm

Input:
$FReg$: Flow Register
f_i: i^{th} flow

for all $TC_j \in$ Traffic Classes $: 0 \leq j < m$ **do**
    for all $f_i \in TC_j$ in $FReg :$ $0 \leq i < n$ **do**
        if $f_i.\alpha = 1$ **then**
            $k \leftarrow t_r / t_a$
            if $k \geq 1$ **then**
                Schedule packets belonging to f_i
                $t_r \leftarrow t_r - t_a$
                $f_i.\alpha \leftarrow TC_j.\alpha$
            end if
        else
            decrement $f_i.\alpha$ by 1
        end if
    end for
end for
CHECK_NEW_FLOWS($FReg, f_i$)

shows the steps of slot scheduling. For scheduling traffic to a transmission slot, the proposed protocol first checks for flows belonging to the highest priority traffic class say TC_j. If there are registered flows belonging to that class, it picks the flow at the front say f_i. Then it checks the I-value $f_i.\alpha$ for unity value. If found so, it calculates *K-value* using Equation (**??**); otherwise, decrements the I-value by one and check the next registered flow. If *K-value* if found to be greater than or equal to 1; it schedules packets belonging to that flow, updates the I-value and takes the next flow. Otherwise, it checks the next flow until all the flows are visited. If all the registered flows belonging to the priority class are visited then the protocol repeats the same set of steps for the registered flows of the next priority class T_{j+1}. The process continues until all the registered flows of all the priority classes are visited.

Algorithm 7 Procedure for Checking new Flows

Input:
$FReg$: Flow Register
f_i: i^{th} flow

CHECK_NEW_FLOWS $(FReg, f_i)$

for all $f_i \in FReg: \ 0 \leq i < n$ **do**
  $k \leftarrow t_r/t_a$
  if $k \geq 1$ **then**
    Schedule packet belonging to f_i
    $t_r \leftarrow t_r - t_a$
    $f_i.\alpha \leftarrow TC_j.\alpha$
    Confirm flowf_i in $FReg$
  else
    if $f_i.tr \geq max\_retry$ **then**
      Remove flowf_i from $FReg$
    else
      $f_i.tr \leftarrow f_i.tr + 1$
    end if
  end if
end for

5.5.4 An Example Showing Packet Scheduling Procedure

The scheduling mechanism is further demonstrated with an example given in Figure **??**. The flow register, $FReg$ is considered to be containing four registered flows- f_0, f_1, f_2, and f_3 belonging to different priority classes. Assume that the flows f_0 and f_1 are of priority class T_0 and the remaining two flows f_2 and f_3 belong to the priority class T_1 and T_2 respectively. Let us consider that at any given time slot, flows f_0, f_2 and f_3 have their flow-I-values equal to 1. Since the flow f_0 has the highest priority, it is scheduled first as it also has its *K-value* greater than or equal to 1. After scheduling the transmission of f_0, the next higher priority flow remaining is the register is f_2. Since the *K-value* for f_2 is also greater than equal to 1, it gets the chance for transmission. At that instant, if the remaining time t_r is smaller than the required transmission time of the subsequent flow f_3 i.e., the *K-value* is less than 1, the flow cannot be scheduled even though the I-value has reached 1. As a result, the remaining time in the slot is assigned to the available best-effort traffic.

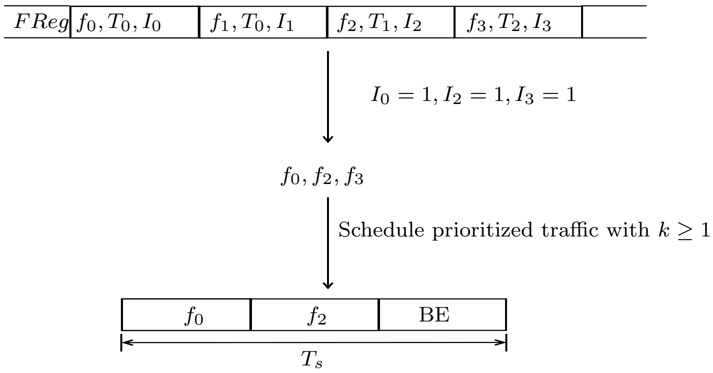


Figure 5-3: TDMA Packet Scheduling: An Example

5.6 Delay Analysis of the Proposed Scheme

When a packet of a particular flow cannot be accommodated in the current time slot, they are buffered in their respective queues based on their priorities. In high load situations, packets belonging to different priority classes remain buffered in different queues. Packets buffered in a queue suffer from scheduling delay and it varies for different scheduling scheme. In this section, we carry out an analysis on the impact of the proposed scheduling algorithm in the latency of different traffic types.

In general, delay in single hop transmission comprises of queuing delay, transmission delay and propagation delay. We estimate the transmission and queuing delay considering the impact of propagation delay to be relatively negligible. To calculate the queuing delay, we have considered a M/G/1/K queuing system for the first three queues viz., LL, $HBLL$, and B queue. According to [?], the average waiting time of n^{th} queue denoted as W_{qn} in M/G/1/K queuing system can be expressed as

$$W_{qn} = \frac{1}{\lambda_n} \sum_{k=0}^{K_n - 1} k p_{d,k} + \frac{K_n}{\lambda_n}(p_{d,0} + \rho_n - 1) - \overline{X}_n, \tag{5.4}$$

Where, λ_n is the Poisson arrival rate, $p_{d,k}$ is the state probability of k packet in the queue, K_n is the maximum size of the queue, ρ^n is the utilization of the queue and $\overline{X_n}$ is the mean service time of packets. The waiting time for the highest priority queue, LL, is independent of the status of any other queues. The waiting time for the other queues depend upon the idle periods of the queues having priorities higher than it. In other words, the utilization of a specific queue depends upon the utilization of other queues having priorities higher than it. The utilization of

n^{th} queue represented as ρ_n can be expressed as-

$$\rho_n = \frac{\rho_n}{(1 - \rho_{n-1})}, n > 1 \tag{5.5}$$

By replacing the value of ρ^n in Equation (??) with the value given by Equation (??), we can easily estimate the value of average waiting time that is the average queuing delay for a packet in each queue. Let the transmission delay and propagation delay be represented ad Tr_d and P_d respectively. Then, the value of Tr_d and P_d can be calculated as-

$$Tr_d = \frac{AP_{size}}{C}$$

$$P_d = \frac{Link\ Distance}{c}$$

where AP_{size} is the average packet size, LC_{max} is maximum capacity of a link and c is the speed of light. Therefore, the total delay in the i^{th} hop link denoted as D_i can be given by

$$D_i = W_{qn} + Tr_d + P_d \tag{5.6}$$

Using Equation (??), the end-to-end delay for h hop distance denoted as D_{mh} can be given as

$$Total\ delay, D_{mh} = \sum_{i=0}^{h} D_i \tag{5.7}$$

Multi-hop end-to-end delay between a given pair of source and destination nodes can be estimated by using Equation (??).

5.7 Simulation and Performance Evaluation

In this section, we have evaluated the performance of the proposed admission control and packet scheduling schemes. Performance of this scheme has been compared with 2C protocol which is presented in Chapter ??. 2C is a link level scheduling scheme which does not employ any packet scheduling mechanism. First, we conduct experiments to determine the optimal slot size for various classes of priority traffic. Using the optimal slot size obtained from this experiment, the throughput and delay performance of different priority classes are measured in subsequent experiments.

5.7.1 Performance Metrics and Simulation Parameters

For performance evaluation, we have considered the following metrics-

(i) Throughput (TP): Throughput refers to the average number of successfully delivered bytes at the destination per second. It is an important metric to provide minimum level of QoS to different priority traffic.

(ii) Delay (D): It is the time difference between the time a packet was delivered at the destination and it was sent by the source. Delay is a key parameter for delay sensitive real-time traffic.

Performance of RPS and 2C MAC protocols are evaluated through extensive simulations using an extended version of NS-2.34 [?] called The Enhanced Network Simulator (TENS) [?]. Throughput and delay performance are measured in normal as well as in saturated load conditions.

Unlike other chapters, wireless half-duplex links with $54Mbps$ bandwidth are used for establishing communication between adjacent nodes. This is done to facilitate more number of test cases by introducing more real-time connections of different priority types. The distance between each link is considered to be $9kms$. A TDMA frame consisting of 100 data slots with guard band of $100\mu s$ is used in this simulation work. VoIP flows with packet size of $160bytes$ maintaining $64Kbps$ data rate with Poisson arrival is used for the simulation of 1^{st} priority traffic. CBR traffic is used for simulating the 2^{nd} priority traffic. To simulate the 3^{rd} priority traffic, we have used VBR traffic as a streaming video. FTP has been used to simulate best-effort traffic. Table **??** gives the details of different simulation parameters considered during simulation. The simulation is carried out for a duration of 300 $seconds$.

Table 5.1: Simulation Parameters for RPS

Parameter	Value
Simulation Area	$50kms \times 50kms$ Flat-grid Area
Application	$CBR, VoIP, VBR, FTP$
Packet Size (VoIP)	$160bytes$ (Payload)
Packet Size (Video Conferencing)	$1250bytes$ (Payload)
Packet Arrival	Poisson
Data Rate (Streaming Video)	$340Kbps$
Bandwidth	$54Mbps$
Link Distance	$9kms$
Routing Protocol	Fixed Routing Protocol
MAC Protocol	2C
Guard Time	$100\mu s$
Queue Length	Application Specific

5.7.2 Results Analysis

Extensive simulation has been carried out to evaluate the performance of RPS protocol. The results obtained are compared with 2C protocol. Throughput and delay performance of RPS protocol is observed in varying traffic load situations with deterministic and probabilistic traffic pattern.

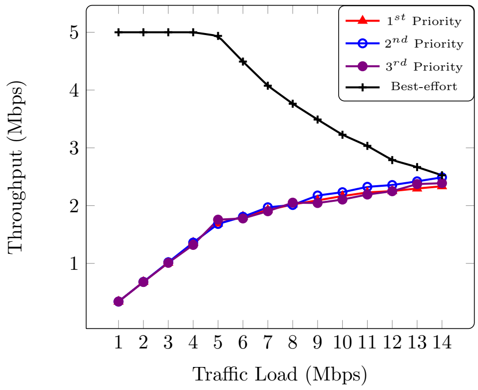
5.7.2.1 Throughput and Delay Characteristics of the Proposed Protocol

In these set of experiments, we analyse the throughput and delay characteristics of the proposed protocol considering probabilistic poisson traffic. Observed protocol performance are presented with reference to real-time traffic load offered.

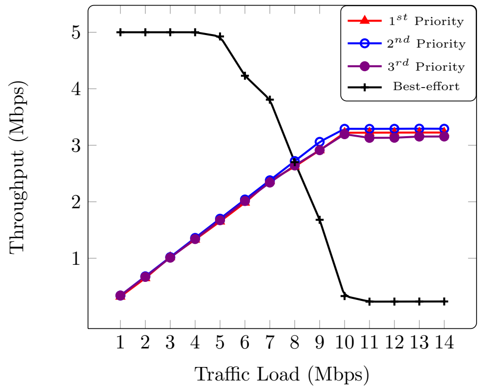
Throughput

Figure **??** presents the throughput characteristics of different classes of priority traffic in 2C and RPS protocols. Throughput of different traffic priority classes in 2C protocol converges very slowly to their maximum values (as shown in Figure **??**). Initially, best-effort traffic grabs a larger share of the overall link bandwidth in the absence of real-time traffic. However, with the increase in real-time traffic load, throughput of best-effort traffic is gradually reduced accommodating real-time traffic.

Throughput achieved by different classes of priority traffic using RPS protocol is shown in Figure ??. It can be clearly observed that at $10Mbps$ load, throughput saturation occurs. Beyond that point, all priority classes of traffic show almost constant throughput. This happens due to the admission control mechanism employed in RPS.



(a) 2C



(b) RPS

Figure 5-4: Throughput performance of different priority classes of traffic using RPS and 2C protocols with probabilistic traffic

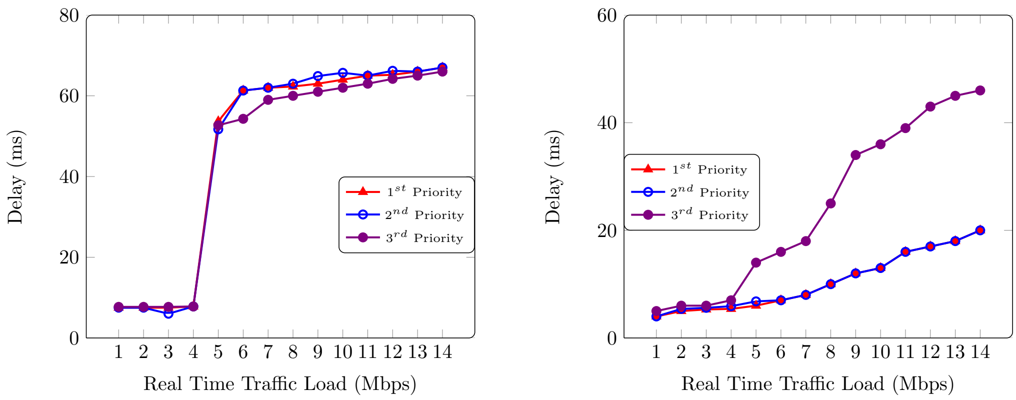
Delay

The end-to-end delay achieved for all classes of priority traffic using 2C and RPS protocols with probabilistic traffic are compared. As shown in Figure ??, the delay of the priority traffic classes are significantly improved by using our proposed protocol. Highest delay improvement is observed in case of 1^{st} priority class of traffic followed by 2^{nd} and 3^{rd}. In 2C, delay performance is shown to be very good in normal load situation. However, at high load, delay jumps to a relatively high level.

By using our scheme, a significant improvement in terms of throughput and delay can clearly be seen from the Figures ?? and ??. A constant throughput performance is delivered by RPS protocol even in high load situation. It shows the effectiveness of the proposed admission control mechanism.

5.7.2.2 Effect of Queue Size over Delay

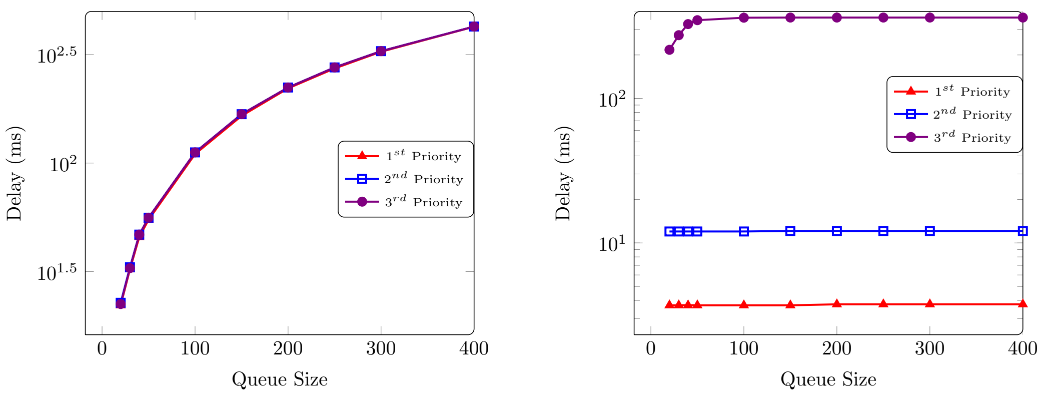
Queue size has got a larger impact in delay performance. In this experiment, we seek to observe the delay performance of different classes of traffic varying the



(a) 2C (b) RPS

Figure 5-5: Delay performance of different priority classes of traffic using RPS and 2C protocols with probabilistic traffic

queue size from 20 to 400. The offered load is considered as constant. In some cases delay is observed to be raising to a very high value. Therefore, the results of this experiments are shown in logarithmic scale for presenting a better view.



(a) 2C (b) RPS

Figure 5-6: Delay Performance over variable Queue Size: RPS vs. 2C

From Figure **??**, it can be seen that in 2C, delay keeps on increasing exponentially with the increase in queue size irrespective of the priority class. However, once the system is stabilized, RPS shows a consistant delay performance for all kinds of priority classes. This can be attributed to the effect of our admission control and packet scheduling mechanisms.

5.8 Conclusion

In this chapter, we have presented a novel packet scheduling scheme which schedules packets taking the QoS requirements of real-time applications into consideration. Grouping the real-time applications into three specific categories, the protocol achieves a fine-tuned control over scheduling of real-time packets. Anticipating the arrivals of real-time packets from their periodicity, the scheduling protocol schedules their packets in the TDMA slots in advance. Using a flow-based localized admission control mechanism, the protocol ensures the quality of service for the ongoing data flows. The proposed protocol has improved delay performance of higher priority traffic extensively. Simulation results have corroborated the efficacy of the proposed packet scheduling and admission control scheme.

In the previous two chapters, we have presented our works on MAC protocol for provisioning QoS in multi-hop WiLD networks. Another QoS enhancement mechanism through packet scheduling has been discussed in the current chapter. In the next chapter, we are going to discuss our last contribution where a QoS-aware gateway-based multi-path routing scheme for multi-hop WiLD networks has been proposed.