Chapter 4

DQBA: A Dynamic QoS-aware Bandwidth Allocation Scheme for Multi-hop WiLD Networks

4.1 Introduction

In multi-hop WiLD networks, unregulated transmissions by various nodes cause congestion around the root node. In provisioning end-to-end QoS, congestion in such networks is considered to be an important issue. In this chapter, we have proposed a dynamic QoS-aware bandwidth allocation scheme to mitigate this problem.

Voice and video based real-time service have become an indispensable part of today's Internet. The prospective real-time applications over WiLD networks such as video-conferencing in rural telemedicine, e-learning, and voice over IP are required to operate while meeting the user expectations. For example, voice quality of most multimedia services involving voice and video transmission deteriorate dramatically if delay increases beyond a certain limit. Similarly, bandwidth-bound applications involving video streaming expect a minimum level of throughput guarantee. Therefore, a rural wireless communication architecture must provide some minimum level of quality of service assurance for smooth functioning of real-time applications.

QoS challenges in multi-hop WiLD networks are a bit different from traditional wired networks. The existing QoS models do not properly fit into multi-hop

WiLD environments due to their architectural differences and several operational constraints. Like other wireless links, the long distance WiFi links are also not reliable due to the factors like signal fading and interference. This unreliability of wireless links create a very dynamic environment where link quality is unpredictable. Moreover, the multi-hop nature of WiLD networks greatly affects the end-to-end throughput and delay of already admitted traffic due to intra-flow and inter-flow interference created among the hops. Schedule-based protocols like TDMA are proven to be better solution for provisioning guaranteed bandwidth in WiLD networks [?, ?, ?]. In a typical TDMA scheme, scheduling of transmissions aim at increasing the overall network performance. In dynamic traffic situations, provisioning of dedicated bandwidth through TDMA scheduling merely solves the QoS issue. A major challenge in QoS provisioning is to schedule access to the medium based on dynamic traffic demands.

In multi-hop WiLD networks, all the links cannot be allowed to transmit at their maximum capacities even if they can do so. Otherwise, congestion is expected to occur around the root node. Therefore, transmission of the nodes towards the bottom of the tree should be restricted to avoid congestion near the root node. In such situation, end-to-end QoS provisioning becomes more challenging. MAC protocols proposed for WiLD networks in [?, ?, ?] do not address this issue rather they focus on maximizing slot reuse among various neighbouring links and thereby improving overall network performance. Although the purpose of multi-hop WiLD and sensor network are different, they resemble in many aspects particularly in their architectures. Taking cognizance of the congestion possibility, research in sensor networks has developed some interesting MAC protocols [?, ?, ?, ?] to avoid congestion and hence achieve high end-to-end data rate. To overcome the congestion problem, most of the protocols propose a hybrid MAC combining CSMA and TDMA protocols. Normally, CSMA and TDMA based protocols are suitable in low and high traffic load situations respectively. To assign relative transmission opportunity, several metrics such as distance from the sink to the node, queue length of children node, and node's slot usage history are used. None of the above-mentioned protocols specifically address the QoS issues of real-time traffic. QDBA [?] reserves one part of the TDMA frame for real-time voice traffic and the other part is kept for the dynamic part of real-time video traffic. Dynamic rescheduling of unused TDMA slots to the needy nodes across different levels is therefore important for end-to-end QoS provisioning.

To this end, we have proposed a dynamic QoS-aware bandwidth allocation scheme, *DQBA* in short to provision QoS for real-time traffic in multi-hop WiLD networks. The proposed scheme classifies the traffic into real-time and best-effort categories. The protocol works in two phases: i) static slot allocation, and ii) dynamic rescheduling of TDMA slots based on QoS demand of the nodes. Initially, the static slot allocation scheme fairly allocates available time slots among all the nodes. At any given instant of time, there may be nodes which are allocated time slots but do not have any data to transmit. At the same time, some other nodes may need more than the allocated data slots for transmission of real-time traffic. This scheme reschedules the unused TDMA slots among the needy nodes in a hierarchical manner by using the parent-child relationship of a tree topology. The parent nodes collect the bandwidth demands of their children nodes and dynamically schedule TDMA slots accordingly. While scheduling time slots for the children, a parent node gives more preference to nodes with QoS demands over the nodes with best-effort. Starting from the bottom part of the tree, this process continues till the 1-hop children of the root node. This protocol avoids congestion in the network which is otherwise inherent in multi-hop WiLD networks due to funneling effect [?].

The rest of the chapter is organized as follows. Section **??** takes a look on the related works. A comparison of dynamic bandwidth allocation schemes proposed for TDMA-based MAC protocols for WiLD as well as for sensor networks are presented in Section **??**. The assumptions, design, and algorithms of the proposed protocol have been discussed in Section **??**. Simulation results of the proposed dynamic QoS-aware bandwidth allocation scheme are presented in Section **??**. Finally, Section **??** provides the conclusion to this chapter.

4.2 Related Works

In the literature of WiLD networks, MAC protocols like 2P [?], WiLDNet [?], JazzyMAC [?], JaldiMAC [?], and Lit MAC [?, ?] have literally changed the face of WiLD networks. Most of them focus on overall network performance improvement. Unfortunately, QoS issues in multi-hop WiLD networks are hardly addressed. TDMA-based MAC protocols proposed in [?, ?] mainly focus on throughput optimization by generating optimal TDMA schedule considering efficient slot reuse. JazzyMAC [?] assigns variable length transmission slots according to the traffic demands of nodes. This protocol is specifically designed to allow neighbours to proceed with parallel independent transmissions without waiting for the marker packet to arrive. It resulted in significant enhancements of network throughput. Unlike the 2P-based MAC protocols, JaldiMAC supports single hop point-tomultipoint network architecture which relies on loose node synchronization. It allows dynamic traffic pattern with varying symmetry ratios to adapt with the asymmetry of Internet traffic and allocates transmission slots based on demands. However, JaldiMAC cannot be scaled up for multi-hop topology with ease and will have similar problems as JazzyMAC.

Sensor network topology is predominantly tree-like; mostly with low bandwidth and short distance links. Unlike traditional wireless networks, wireless sensor network (WSN) consist of spatially distributed autonomous sensors which do not need to communicate directly with the nearest high-power base station. Rather, the sensor nodes cooperatively pass their data through the network to a central location, called sink node, by communicating with their local peers. The nodes forward traffic hop-by-hop as such the major traffic patterns are many-to-one forming a tree [?]. Thus, the sensor network architecture has a phenomenal similarity with multi-hop WiLD networks although traffic characteristics are quite different.

In tree-based networks, nodes closer to the root need to forward more data packets than others. Traditional MAC protocols tend to provide fair access and hence are not suitable in such network architecture. If traditional MAC protocols are used in many-to-one network topology, congestion towards the root node is inevitable. Recognizing this fact in WSN, a number of protocols such as presented in [?, ?, ?, ?, ?, ?, ?] are developed. A hybrid approach using schedule-based medium access in traffic intensive regions and contention-based MAC in low traffic zones is proposed in [?, ?, ?]. Z-MAC [?] acts like a contention-based protocol under low traffic conditions and a schedule-based protocol under high traffic conditions by using the schedule computed by DRAND (Distributed RAND) [?]. It allocates time slots to every node ensuring that no two nodes among a two-hop neighbourhood are assigned the same time slot. In order to improve utilization in low load situation, Z-MAC allows non-owners of a slot to contend for a slot if it is not being used by its owner. Similarly, Funneling-MAC [?] tried to mitigate the funneling problem by a sink-oriented scheduling protocol which is also a hybrid of TDMA and CSMA protocols. It uses TDMA scheduling in the intensity region and employs CSMA in the rest of the network to provide flexibility. It is localized in operation because TDMA only operates in the intensity region close to the sink and not across the complete sensor network. These two protocols employ fixed slot TDMA and hence do not provide any priority to the nodes considering their requirements. I-MAC [?] assigns different levels of priority to different nodes according to their role in the network. During scheduling of any slot, the owner of the slot gets the first priority. The non-owner nodes can compete to use a slot only when the owner node doesn't need it. The chance of getting a slot by a non-owner node also depends on its priority level. Queue-MAC [?] is another hybrid protocol which has addressed the issue of burst network traffic by allocating time slots of dynamic size. In this protocol, packets coming from the children nodes carry their load information through a special field called *queue indicator*. The frame comprises a CSMA and a TDMA component. Initially, a node starts its transmission using CSMA protocol. With an increase in load, the active TDMA period is accordingly extended by adding more time slots to increase the bandwidth. Queue-MAC considers only single hop topology because of which it needs multi-hop extension to fit in WiLD networks.

TreeMAC [?] attempted to solve the congestion problem by using only TDMA-based MAC for the entire WSN. In this protocol, time slots are allocated to the nodes following a 2-dimensional approach. A time cycle is divided into frames and each frame into slots. A parent node determines children node's frame assignment based on their relative bandwidth demand, and each node calculates its own slot assignment according to its hop distance to the sink. Each children node notifies its parent of its bandwidth demand by piggybacking this information in a routing beacon message. Making use of queue length of all the sensor nodes, iQueue-MAC [?] assigns TDMA slots of variable size for packet transmission. iQueue-MAC uses CSMA and TDMA protocols in light and high load situations respectively. It integrates a variable namely *TDMA period* to provide adaptive data transmission based on children node's queue length. Utilization-Based Scheduling [?] used Spatial-TDMA (STDMA) based dynamic channel access mechanism to increase throughput in wireless mesh networks. In accordance with the node's slot usage history and packet-queue occupancy, each node is assigned a dynamic weight value which approximates the node's demand for transmission slots in the next frame. The number of times TDMA slots assigned to each node in a single frame is proportional to its weight. To allocate bandwidth dynamically for real-time traffic, a QoS-aware Dynamic Bandwidth Allocation (QDBA) scheme for WiMAX based networks is proposed in [?]. QDBA scheme divides a TDMA frame into two parts, one is steady part for real-time voice traffic and the other is the dynamic part for real-time video traffic. The base station allocates bandwidth to the subscriber stations based on the QoS requirements of the connections.

The MAC protocols discussed above have mostly addressed the congestion issue by employing a TDMA-based MAC at high traffic load. TDMA-slot assignment is further improved by incorporating adaptive or variable TDMA which allocates time slots based on demands. The traffic demand is decided from the role of a node in the network and explicit information such as queue length, slot usage history, etc. However, none of the above protocols consider provisioning QoS for the real-time traffic with dynamic bandwidth allocation in a precise manner. This motivates us in designing a dynamic bandwidth allocation scheme which will ensure end-to-end QoS for real-time traffic.

4.3 Comparison of Existing Dynamic Bandwidth Allocation Schemes

Table **??** of Chapter **??** (page **??**) provides a brief point-by-point comparison of the existing dynamic bandwidth allocation schemes proposed for wireless mesh and sensor networks. Dynamic bandwidth allocation schemes are mostly used in high load situations of many-to-one communication networks. Various schemes discussed in Section **??** on page **??** mainly includes TDMA-based slot allocation according to the demand of the nodes. We are motivated to design a QoS-aware dynamic bandwidth allocation scheme for multi-hop WiLD networks because of the following two reasons: i) many-to-one architecture of WSN resembles with multi-hop WiLD networks and congestion scenario similar to or even worse than sensor networks is anticipated in such networks, and ii) none of the dynamic bandwidth allocation schemes found in the literature provide full-fledged QoS support to real-time traffic over multiple hops.

4.4 The Proposed Dynamic QoS-aware Bandwidth Allocation Scheme

In provisioning QoS for real-time traffic, the proposed scheme aims at meeting the QoS demands for upward traffic which usually suffers from congestion close to the gateway node due to funneling effect [**?**]. The proposed scheme works in two phases. In the first phase, guaranteed TDMA slots are statistically allocated to all the nodes of the network. It regulates various links' transmission by introducing an operational constraint called *Restricted Simultaneous Operation (R-SynOp)*. In the second phase, transmission opportunities for QoS-bound traffic are enhanced through sharing of the unused bandwidth dynamically among the needy nodes. In this section, we have discussed the assumptions, structure of the TDMA frames, and the design of proposed scheme in details.

4.4.1 Protocol Assumptions

In multi-hop WiLD network configuration, wireless nodes are equipped with multiple radios co-located on the same tower. We consider tree topology in our model which can easily be formed either constructing it physically or by logically converting the mesh/graph topology. Figure **??** depicts a typical tree topology. Using this diagram, various concepts and assumptions related to the proposed scheme are explained.

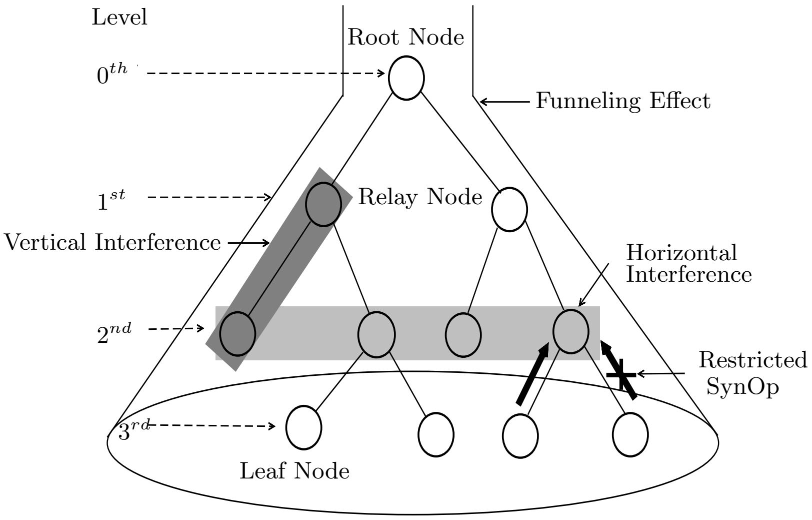


Figure 4-1: Explaining Various Concepts Related to DQBA in a Tree Topology

(i) **Use of Single Channel**
All the radios installed in a multi-hop WiLD network use only a single channel for transmission and reception. Use of single channel reduces the overhead of channel assignment and provides a rigid interference model. Using a single channel for the backbone leaves the rest of the channels for local access which can minimize radio frequency pollution.

(ii) **Restricted SynOp**
As discussed in section **??**, it is possible to carry out Simultaneous Synchronous Operation (SynOp) in multi-hop WiLD networks. However, for upward traffic, allowing all the children to transmit simultaneously by performing SynOp may introduce congestion in the ancestral links which may in turn degrade the network performance. To avoid such a situation to occur, the sibling nodes are not allowed to perform SynTx in forwarding traffic to

their parent nodes. To accommodate this concept, we have modified SynOp and termed as *Restricted-SynOp (R-SynOp)* which restricts the simultaneous transmission in the upward direction, that is children to parent. Thereby, it prevents the parent or its predecessor nodes from being congested. However, the use of restricted SynOp in the 1^{st} level nodes is an exception as the root node is assumed to be connected with very high bandwidth link. The concept of Restricted-SynOp is demonstrated in Figure **??**.

(iii) **Funneling Effect in WiLD Networks**

WiLD networks using traditional MAC exhibit a unique funneling effect where traffic generated in the children nodes travel hop-by-hop towards the root in a many-to-one traffic pattern. This combination of hop-by-hop communications and data forwarding to the root node creates choke points on the free flow of traffic, particularly in the nodes nearer to the root. The funneling effect [**?**] leads to increased transit traffic intensity and delay as events move closer towards the root node. It may attribute to significant packet collision, congestion, and loss leading to collapse of the network. Funneling effect is shown pictorially in Figure **??**.

(iv) **Node Synchronization**

All the nodes in the network are assumed to be tightly synchronized. Due to the existence of clock drift and lack of perfect synchronization, all the nodes in the network are required to be synchronized periodically.

(v) **Horizontal Interference**

With the use of point-to-point directional links, nodes at the same level of a tree topology do not interfere with each other. Therefore, we have not considered any horizontal interference which exists in networks using omnidirectional antennas as mentioned in [**?**]. The concept of horizontal interference is explained in Figure **??** where the nodes shaded at the same horizontal level and hence do not interfere with each other. This consideration greatly enhances the slot reuse capability of the network.

(vi) **1-hop Vertical Interference**

Considering single channel operation, a node cannot perform Mix-Tx-Rx [**?**] operation. In half-duplex wireless links, a node performing SynTx makes all its 1-hop neighbours to perform SynRx operation. Thus, any vertical 1-hop neighbour violating this rule leads to interference. As shown in the Figure **??**, the nodes shaded vertically are in the same vertical interference zone. This consideration limits the slot reuse capability of the network but by the use of point-to-point links and SynOp, interference is limited among the 1-hop neighbours only.

4.4.2 Frame Structure

In this scheme, a TDMA frame is broadly divided into two parts: synchronization interval and service interval. Figure **??** describes the format of a superframe indicating the distinct elements of it.

Synchronization Interval

The synchronization interval comprises of control and contention slots which are used to synchronize the nodes in the network. The control slots are used to transmit control information such as TDMA frame and bandwidth grant. The number of control slots required to disseminate control information to all the nodes in a network is equal to the depth of the topology tree. Contention slots are used only by the non-root nodes (i.e., relay and leaf nodes) for the purpose of sending node join requests and bandwidth demands.

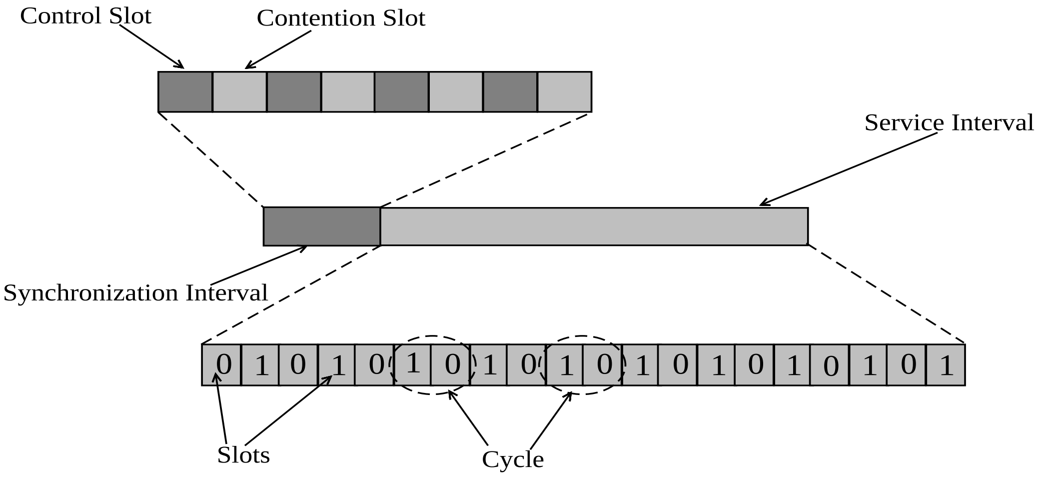


Figure 4-2: A Customized Superframe Structure

Service Interval

Service interval is the time elapsed between two consecutive synchronization intervals during which data transmissions are scheduled. The service interval is equi-partitioned into unit slots which are necessarily even in numbers. Since transmissions of all the nodes of a network can be scheduled within two slots, we have merged two consecutive unit slots to form a time cycle, T_{cycle}. Hence, the service interval can be visualized as a collection of time cycles.

4.4.3 The Protocol (DQBA)

Let a multi-hop WiLD network be represented as a tree $T = (V, E)$ where, V is the set of nodes and E is the set of links in the network. Let $n \in V$ be any arbitrary node. $Adj[n]$ and $Child[n]$ represent the number of adjacent links and the number of children nodes of the node n respectively. The protocol distinguishes three different types of nodes in the network and entrusts different responsibilities to them. The different categories of nodes are: i) root node, ii) relay node and iii) leaf node.

The root node R is is a special node which satisfies the condition $Adj[R] = Child[R]$ and acts as the central coordinator of the network. It carries out the task of constructing TDMA superframe, generating control packets, disseminating TDMA frames, multi-hop node synchronization and static slot allocation process. Number of adjacent links for a relay node one more than the number of children it has. It receives control packets, processes and forwards them to its children. In addition, it forwards node join request to its parent and initially carries out the static slot allocation process for its 1-hop children nodes too. These nodes perform dynamic slot assignment on receipt of the traffic demands from their children nodes. Relay nodes are responsible for sending/receiving bandwidth request/release to/from their parent nodes. A leaf node doesn't have any children and always has exactly one adjacent link. Leaf nodes are the end-points of the network which carry out the task of receiving control packets, generating node join request and sending bandwidth demands to their parent nodes.

We logically divide a multi-hop WiLD network into 1-hop clusters as shown in Figure **??**. All the clusters exhibit a parent-child relationship and have identical behaviour except at the cluster containing the 1^{st} level nodes. The k^{th} cluster in level l consists of r_k^l number of nodes in such a way that $(r_k^l - 1)$ nodes are 1-hop children of a given parent node. Let us assume that at a given level l in the network, there are m number of clusters. The parent node in the k^{th} cluster of level l is represented as $n_{p_k}^l$ and the i^{th} child in that cluster is represented as $n_{i_k}^l$ such that $0 < i < r_k^l$.

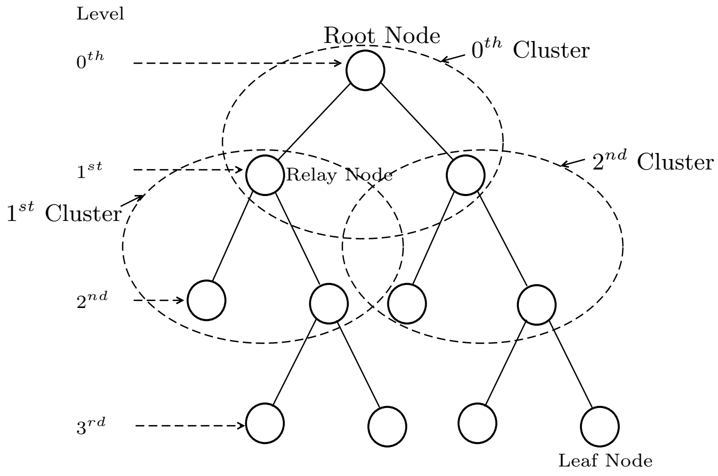


Figure 4-3: A Cluster-based Network Architecture

The initial slot allocation and dynamic QoS-aware slot scheduling phases of the proposed scheme are explained below-

4.4.3.1 Initial Static Slot Allocation

During the network initialization, a static slot allocation process is carried out. The basic task behind this slot allocation process is to equally distribute the time cycles of a parent node among its children nodes of a given cluster. If the parent node, $n^l_{p_k}$ of a cluster has been allocated a transmission slot T_i in a given time cycle, T_{cycle} where, $i \in \{0,1\}$, the other slot in the T_{cycle}, T_j, where $j \in \{0,1\}$ and $T_i \neq T_j$ can be occupied by any other children node of that cluster. If T_j is allocated to node $n^l_{i_k}$, no other sibling nodes of $n^l_{i_k}$ shall be assigned the same time slot for transmission to their parent node, $n^l_{p_k}$.

The root node which acts as the central coordinator, starts the slot allocation process. With the assumption that the root node has greater transmission capabilities in forwarding traffic outside the network, the root node shares the T_{cycles} equally between itself and all of its 1-hop children nodes. Therefore, if the total number of slots in a TDMA superframe is γ, the slot share of each 1-hop children can be given by

$$\gamma_{n^1_{i_k}} = \frac{\gamma}{2}$$

Here, $\gamma_{n^1_{i_k}}$ represents the bandwidth share of i^{th} node belonging to k^{th} cluster of level one.

Now, as all the 1-hop children of the root node get the slot share equal to their parent, they can further share half of their allocated slots among their children. The slots allocated to a node needs is shared equally among its children nodes. Therefore, the bandwidth share received by the i^{th} child in k^{th} cluster of level l denoted as $\gamma_{n^l_{i_k}}$ can be given by,

$$\gamma_{n^l_{i_k}} = \frac{\gamma_{n^l_{p_k}}}{r^l_k}$$

Algorithm 3 Static Slot Allocation Algorithm

```
Input:
$T_{fr}$: TDMA Frame
$n_k$: Parent of $k^{th}$ Cluster
$Child[0..(n-1)]$: Children of $n_k$
if ($n_k$ = Root Node) then
    for all $T_{cycle}$ in $T_{fr}$ do
        Assign $T_i$ to $n$ and $T_j$ to $Child[0..(n-1)]$ such that $T_i \neq T_j$ ; $0 \leq i, j \leq 1$
    end for
else
    for all $T_{cycle}$ in $T_{fr}$ do
        if $T_i$ is assigned to $n_k$ then
            Assign $T_j$ to any one of the nodes in $Child[0..(n-1)]$ such that $T_i \neq T_j$ ; $i \geq 0; j \leq 1$ in round-robin fashion
        end if
    end for
end if
```

This static slot allocation process is subsequently carried out by all the nodes which has at least one children node. The initial slot allocation process is illustrated in Algorithm **??**. The algorithm first checks whether a node is root or non-root and then it starts allocating slots. The root node allocates every next slot in a frame to itself and the remaining to all its children nodes. Thus, sharing equal bandwidth between itself and its 1-hop children nodes. In case of a non-root node, every alternate slot is allocated to one of the children nodes in a round robin fashion. This algorithm ensures that no two nodes at 1-hop distance get the same slot for transmission which would otherwise have resulted in 1-hop vertical interference.

4.4.3.2 Dynamic Slot Scheduling

The proposed protocol introduces a dynamic slot rescheduling scheme based on bandwidth demands of the children nodes. Bandwidth demands of children nodes are placed to parent node through sending Traffic Indication Map (TIM). After receiving the TIM's from its children, a parent node tries to allocate time slots according to their demands. If a parent node is not able to allocate required number of slots to its children nodes, it prepares a TIM specifying the requirement and sends to its immediate parent node. This protocol classifies the demands of the children nodes into two broad categories: QoS demand (*Q-demand*) and Additional demand (*A-demand*). The Q-demand indicates the total bandwidth demand for the delay and bandwidth sensitive real-time traffic such as VoIP and videoconferencing whereas A-demand indicates the requirement of non real-time (best-effort) traffic. Both the demands are specified in terms of number of slots. The TIM's are sent in the last slot allocated to the node for transmission either through special packets or by piggybacking in data packets. In the absence of slots for transmission, the TIM's are sent in contention slot allocated to a node.

On receipt of TIM's from all the children, a parent node starts the dynamic slot rescheduling process and tries to fulfill the bandwidth demands while preparing transmission schedule for its children. The rescheduling process prioritizes Q-demands over A-demands. Therefore, a parent node first schedules the Q-demands of all of its children and then tries to schedule the A-demands. After generating the schedule, the parent node sends it to all of its children nodes. The scheduling process is highly localized where a parent prepares transmission schedules for its 1-hop children and distributes it without burdening or influencing other nodes with the exception in certain situations.

Consider cluster k belonging to level l. Let the slot allocated to parent of the cluster, $n^l_{p_k}$ be $\gamma_{n^l_{p_k}}$ and the demand from i^{th} child of the cluster be $\beta_{n^l_{i_k}}$. Then, the slots allocated to a children node, $n^l_{i_k}$ can be calculated by using the Equation (**??**).

$$\gamma_{n^l_{i_k}} = min(\beta n^l_{i_k}, \gamma_{n^l_{p_k}} \times \frac{\beta_{n^l_{i_k}}}{\sum_{j=0}^{r^l_k} \beta_{n^l_{i_k}}}) \tag{4.1}$$

In situations when the Q-demands of all the children of a node is less than the allocated bandwidth, it releases the additional bandwidth to its parent in order to enable the use of those unused slots by other nodes. In such cases, a maximum of 80% of the total available bandwidth are released. The remaining 20% is reserved for future communication by the node. This phenomenon is termed as *Bandwidth Release*. On the other hand, when the total demand of all its children exceeds the allocated bandwidth, the available bandwidth is shared among its children based on their demands and the additional bandwidth request which could not be served are sent to its parent. The process of requesting additional TDMA slots is similar to that of a children node placing traffic demands to its parent. Dynamic slot scheduling process is eventually started by the leaf nodes and continues till 1-hop children of the root in a hierarchical fashion.

Algorithm **??** presents the working of the dynamic slot allocation process. The cluster head of cluster k, n_k is distributing a total number of γ_{n_k} slots of a TDMA frame, T_{fr} among its children. The algorithm first checks in which frame the node has a slot allocated to it and proceeds with allocating the other slot to any one of its children. The bandwidth γ_{n_k} is divided between Q-demand and A-demand traffic in the ratio 80:20. The time slots allocated for Q-demand and A-demand traffic are represented by $\gamma_{n_k}^Q$ and $\gamma_{n_k}^A$ respectively. After receiving the Q-demand (α_i) and A-demand (β_i) from all of its n children, $Child[0..(n-1)]$, the parent node computes the cumulative Q-demand, α^c and A-demand, β^c. Out of the available slots in $\gamma_{n_k}^Q$, it serves the Q-demands of all of its children nodes first. After serving the Q-demands, the A-demands of all the children nodes are served from the 20% bandwidth earmarked for A-demand, $\gamma_{n_k}^A$. Any slots remaining unused after slot allocation for Q-demands, are also allocated to A-demands.

The RESOURCE_RELEASE procedure releases the excess time slots to its parent for use by the higher level clusters. If the Q-demands of the children nodes cannot be met, the cumulative slot demand ($\alpha^c + \beta^c$) is placed to its immediate parent nodes for additional slot allocation. This process is carried out by RESOURCE_REQUEST procedure. A parent node allocates Q-share of the children nodes in round robin fashion. Therefore, the unused slots of a children node automatically get shared among the other needy nodes. Sending additional slot demand to the parent and allocating unused slots in round robin fashion implicitly serves the purpose of *Resource Request* and *Resource Grant*.

Algorithm 4 Dynamic Slot Scheduling Algorithm

```
Input:
$T_{fr}$: TDMA Frame
$\gamma_{n_k}$: Slots assigned to $n_k$
$\alpha_i$: Q-demand of $i^{th} node$
$\beta_i$: A-demand of $i^{th} node$
$\gamma_{n_k}^{Q} \leftarrow 0.8 \times \gamma_{n_k}; \gamma_{n_k}^{A} \leftarrow 0.2 \times \gamma_{n_k}; n_{child}$ : child number
$i \leftarrow 0; \alpha^c \leftarrow 0; \beta^c \leftarrow 0; n_c hild \leftarrow 0$
while $i < n$ do
    $\alpha^c \leftarrow \alpha^c + \alpha_i$
    $\beta^c \leftarrow \beta^c + \beta_i$
    increment $i$
end while
for all $T_{cycle}$ in$T_{fr}$ do
    if $T_i$ is allocated to $n_k$ then
        $m \leftarrow (n_c hild + +)\ mod\ n$
        $r \leftarrow 0$
        while $r < n$ do
            if $(\alpha_m > 0\ \ \&\ \ \gamma_{n_k}^{Q} > 0)$ then
                Allocate $T_{(i+1) \bmod 2}$ to $Child[m]$
                decrement $\alpha_m, \gamma_{n_k}^{Q}$
                if $\gamma_{n_k}^{Q} = 0$ then
                    RESOURCE_REQUEST($\gamma_{n_k}^{Q}, \gamma_{n_k}, \alpha^c, \beta^c$)
                end if
                break loop
            end if
        end while
        if $r = n$ then
            $r \leftarrow\ 0$
            while $r < n$ do
                if $(\beta_m > 0\ \ \&\ \ (\gamma_{n_k}^{A} + \gamma_{n_k}^{Q}) > 0)$ then
                    Allocate $T_{(i+1) \bmod 2}$ to $Child[m]$
                    decrement $\beta_m, \gamma_{n_k}^{A} + \gamma_{n_k}^{Q}$
                    break loop
                end if
            end while
        end if
    end if
end for
if $(\gamma_{n_k}^{Q} + \gamma_{n_k}^{A}) > 0$ then
    RESOURCE_RELEASE($p_{n_k}, \gamma_{n_k}$)
end if

procedure RESOURCE_REQUEST($\gamma_{n_k}^{Q}, \gamma_{n_k}, \alpha^c, \beta^c$)
    $\gamma_{n_k} \leftarrow \alpha^c + \beta^c$
end procedure

procedure RESOURCE_RELEASE($\gamma_{n_k}^{Q}, \gamma_{n_k}^{A}, \alpha^c, \beta^c$)
    Release the remaining $(\gamma_{n_k}^{Q} + \gamma_{n_k}^{A})$ number of Slots
    $\gamma_{n_k} \leftarrow (\alpha^c + \beta^c) - (\gamma_{n_k}^{Q} + \gamma_{n_k}^{A})$
end procedure
```

4.4.4 An Example Showing the Working of DQBA

Figures **??** and **??** illustrate the working principle of DQBA scheme with the help of suitable diagrams. Let us assume that there are five nodes altogether in the network where R is the root node, A is an intermediate node, and B, C and D are leaf nodes. R begins the static slot allocation process by sharing the available time slots equally between itself and its 1-hop neighbours. Hence, out of the total eight slots available, four alternate slots [2, 4, 6, 8] are allocated to each the 1-hop nodes of R i.e, A and B and the remaining four [1, 3, 5, 7] are retained by itself. It may be noted that nodes A and B can simultaneously transmit during the assigned time slots as discussed in Section **??** of Chapter **??**. Since R is assumed to have a high bandwidth connectivity, no restriction is imposed in the transmission of level-1 nodes. Therefore, allocation of time slots to level-1 nodes is a special case of the proposed scheme. However, the same is not true for the nodes of other levels of the tree. Since, node A has two children nodes i.e., C and D, half of the node A's time slots [4, 8] are allocated to both C and D retaining [2,6] slots for itself. As shown in Figure **??**, nodes C and D will get two slots each.

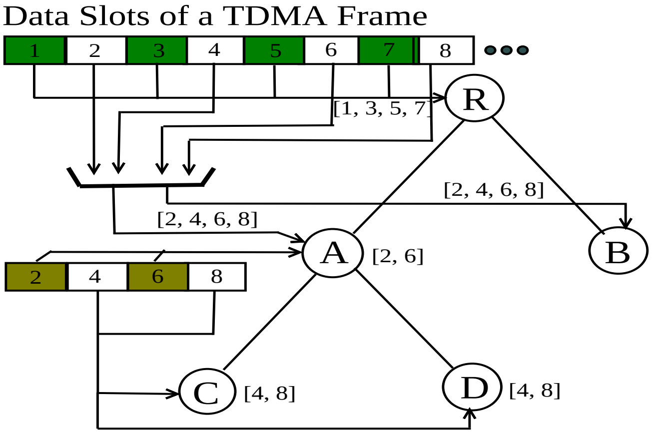


Figure 4-4: A Figure Depicting Static Slot Allocation Process

To describe the working of the dynamic bandwidth allocation phase of the proposed scheme, let us assume that node B has no bandwidth demand, C has only best-effort traffic (A-demand), and D has real-time traffic (Q-demand) to transmit. After receiving the bandwidth demands from the children nodes C and D, the parent node A assigns three time slots to C and one to D. Since node C has Q-demand, it is given priority over B and hence three slots are assigned to it taking one away from the node C. As DQBA reserves 20% bandwidth for best-effort traffic, a minimum of one slot is allocated to node D.

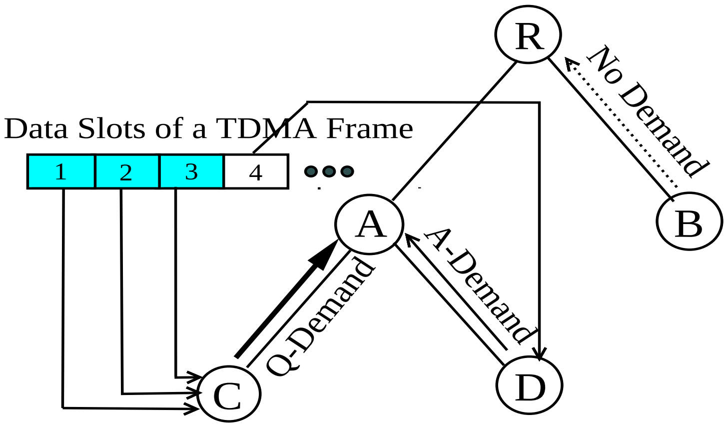


Figure 4-5: A Figure Depicting Dynamic Slot Allocation Process

4.5 Simulation and Performance Evaluation

In this section, we have evaluated the performance of the proposed DQBA scheme. Performance of DQBA scheme is compared with TreeMAC [**?**]. TreeMAC is a very similar protocol proposed for wireless sensor networks in which parent node allocates dynamic bandwidth to children nodes according to their demands. DQBA treats QoS-bound and best-effort traffic demands differently whereas TreeMAC does not. Another difference between these two approaches is that TreeMAC initializes the nodes in the networks by using a CSMA approach whereas DQBA statically allocates initial time slots according to the distance of a node from the root.

4.5.1 Performance Metrics and Simulation Parameters

The following metrics have been considered for performance evaluation of our proposed protocol:

(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 service in a network.

(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 node. Delay is a very essential parameter for delay sensitive real-time traffic.

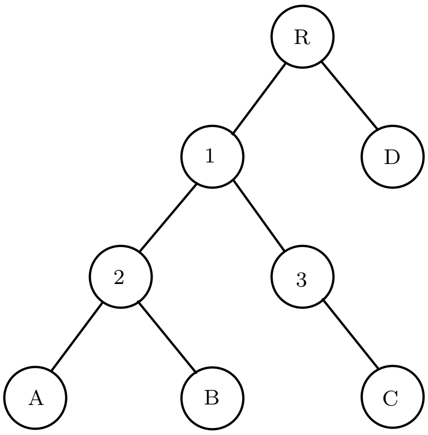


Figure 4-6: Simulation Topology for DQBA

Various experiments have been conducted using NS-2 [?] simulator to analyze the performance of DQBA in terms of throughput and delay of real-time as well as best-effort traffic. The simulation is carried out for a duration of 300 *seconds*. Table **??** presents the different parameters considered along with their values for the simulation purpose. CBR traffic has been used to introduce real-time (Q-demand) and best-effort (A-demand) traffic load.

Table 4.1: Simulation Parameters for DQBA

Parameter	**Value**
Traffic Types	CBR
Packet Size (CBR)	$1250 bytes$(Payload)
Packet Interval (CBR)	$33ms$
Routing Protocol	Fixed Routing Protocol
Simulation Area	$50kms \times 50kms$ Flat-grid Area
Radio Propagation Model	Two Ray Ground Reflection Model
Bandwidth	$11Mbps$
Antenna Type	Grid Parabolic Antenna
Antenna Gain	$24dBi$
Distance per Hop	$9kms$
No. of Nodes (Max.)	8
TDMA Slot Time	$4ms$
Guard Time	$100\mu s$
TDMA Queue Length	100

A network topology as shown in Figure **??** is considered for the simulation purpose. Node R is the root node whereas nodes 1, 2, and 3 are intermediate, and nodes A, B, and C are leaf nodes. From this topology, different cases with 1-hop, 2-hop, and 3-hop network topology are created. We represent the real-time and best-effort traffic from a node Y as RT(Y) and BE(Y) respectively. Throughput and delay performance of the proposed scheme is analyzed in two different situations:

i) when all the children nodes of a cluster have equal bandwidth demand. This scenario is simulated by offering similar Q-demands and A-demands from both the sub-trees of the root. The demands generated are gradually increased by adding more numbers of CBR connections, and ii) when Q-demand is available only in a few children nodes. The other nodes may or may not have A-demand. This scenario is created by offering both Q-demands and A-demands from the left sub-tree and only A-demands from the right sub-tree. In a similar setup, all the above experiments have been carried out using TreeMAC protocol. Finally, the throughput and delay performance of DQBA is compared with TreeMAC.

4.5.2 Throughput and Delay Performance

We have evaluated the throughput and delay performance of DQBA and TreeMAC protocols in this section. Traffic load is gradually increased by adding more number of video conferencing connections (real-time traffic). Video conferencing is chosen as real-time traffic load as it is both delay and bandwidth sensitive. It introduces packets of size $1250 bytes$ at an interval of $33 ms$. The load of best-effort traffic is equally increased with real-time traffic. Since delay performance has varied in a larger range, they are presented in logarithmic scale.

4.5.2.1 With Uniform Traffic Load from all the Sub-trees

Uniform traffic load situation is created by introducing equal amount of A-traffic and Q-traffic from both left and right sub-trees. Throughput and delay performances in 1, 2, and 3 hop scenarios are analyzed in these traffic situations. Considered simulation topology and the experiment results are discussed in the respective sections of all the considered scenarios.

Throughput and Delay Performance in 1-hop Scenario

The simulation topology and traffic pattern for this experiment is shown in Figure **??**. From the Figure **??**, it can be observed that throughput continues to increase till the load reaches the saturation point. Throughput saturation occurs approximately at $4 Mbps$ aggregate load. This is true for both TreeMAC as well as DQBA. As the nodes A and B are 1-hop children of R, throughput saturation for the links $A \rightarrow R$ and $B \rightarrow R$ occur at around $4 Mbps$ aggregate load. Beyond

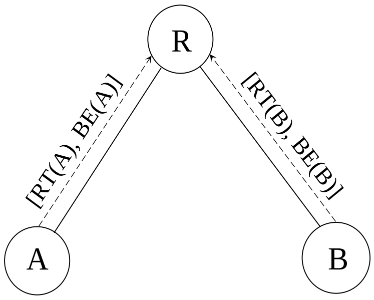
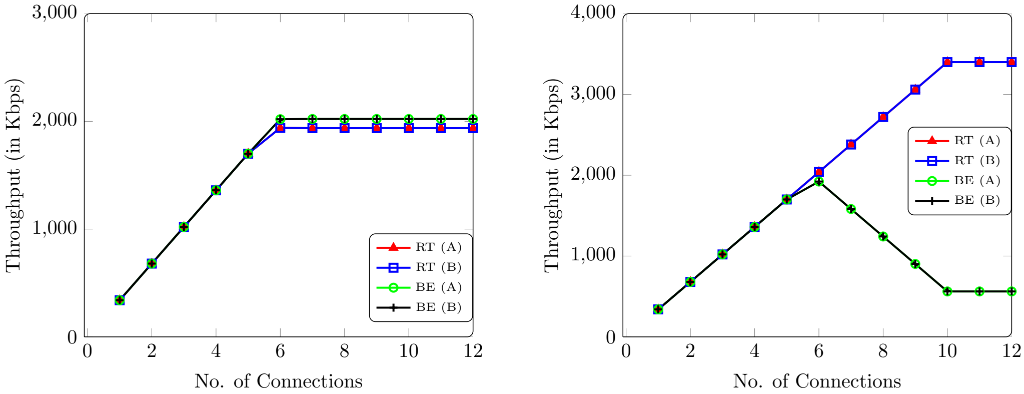


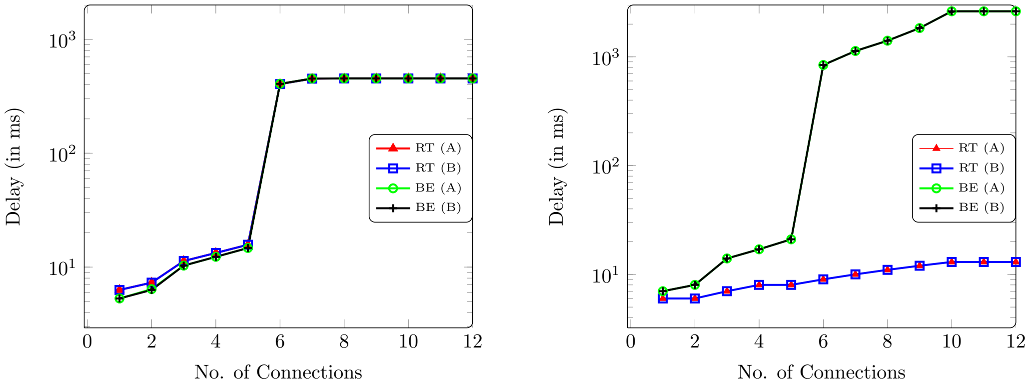
Figure 4-7: Network configuration for 1-hop scenario with uniform load from both the children nodes



(a) TreeMAC **(b)** DQBA

Figure 4-8: Throughput achieved by RT and BE traffic with uniform load from both the children in 1-hop topology

the saturation point, throughput of both types of traffic show invariable performance in TreeMAC. As our scheme provides priority to Q-demand over A-demand while scheduling slots, performance or real-time traffic is not affected till the total bandwidth ($3.4 Mbps$) is exhausted. It may be noted that a minimum of 20% bandwidth has been kept reserved for BE traffic. This is shown in Figure **??**.



(a) TreeMAC **(b)** DQBA

Figure 4-9: Delay of RT and BE traffic with uniform load from both the children in 1-hop topology

Delay performance of both the protocols have been shown in Figure **??**. In normal offered load, both TreeMAC and DQBA show excellent delay quality. Once the network is saturated with traffic, delay of both RT and BE traffic quickly reach a very high value in TreeMAC (see Figure **??**). DQBA improves delay performance of RT traffic attaining value as low as $13ms$. This is shown in Figure **??**.

Throughput and Delay Performance in 2-hop Scenario

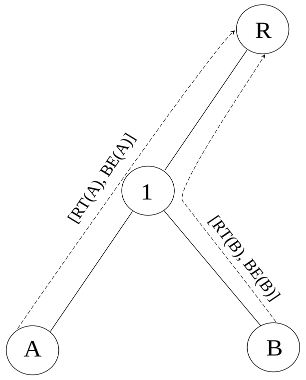
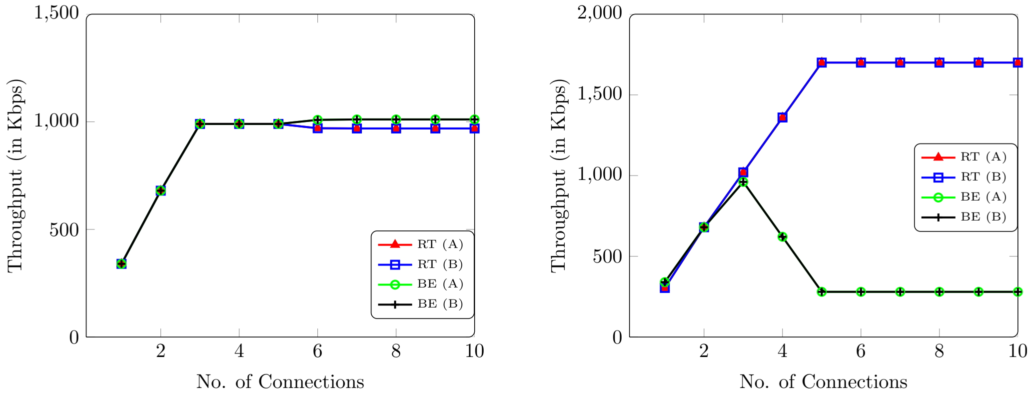


Figure 4-10: Network Configuration for 2-hop scenario with uniform load from both the children of a sub-tree

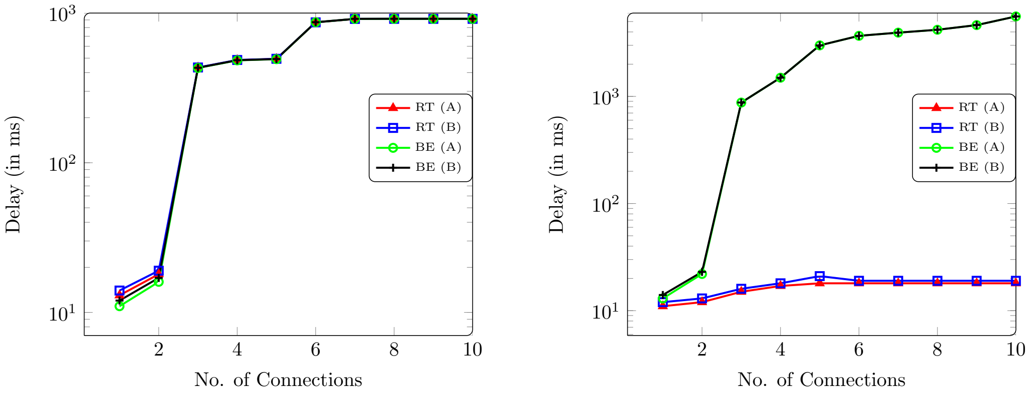
The 2-hop simulation scenario considered in this experiment is presented in Figure **??**. Real-time and best-effort traffic are introduced from both the leaf nodes A and B. Here, traffic from the nodes A and B shares the bandwidth of the link $1 \rightarrow R$.



(a) TreeMAC **(b)** DQBA

Figure 4-11: Throughput achieved by RT and BE traffic with uniform load from both the children of a sub-tree in a 2-hop topology

As shown in Figure **??**, with increase in the number of connections in TreeMAC, throughput achieved by real-time and best-effort traffic constantly increases in low load. Beyond the saturation point, throughput remains consistent around a certain value. Figure **??** demonstrates an excellent improvement in the throughput of RT traffic over A-traffic. Since DQBA provides priority to RT traffic, it exhibits constant throughput for RT traffic compromising the BE traffic for both the nodes.



(a) TreeMAC **(b)** DQBA

Figure 4-12: Delay of RT and BE traffic with uniform load from both the children of a sub-tree in a 2-hop topology

2-hop delay performances of both the protocols are demonstrated in Figure **??**. Up to two connections of BE and RT each, TreeMAC displays perfect delay characteristics. But beyond that point, delay value is increased to an unacceptable level for all types of traffic (Figure **??**). Delay performance of RT traffic has been quite improved by our proposed scheme even in very high load situations. This is as shown in Figure **??**.

Throughput and Delay Performance in 3-hop Scenario

To analyze throughput and delay performance in 3-hop configuration, traffic flows are considered as shown in Figure **??**. Nodes A and B transmit both RT and BE traffic to the root node R via the relay nodes 1 and 2. Node C transmits only RT traffic to node R. In this case, The nodes A and B shares the link $2 \rightarrow 1$ which in turn shares the link $1 \rightarrow R$ with C.

Figure **??** presents the throughput performance of TreeMAC in the traffic scenario shown in **??**. The throughput of the link $2 \rightarrow 1$ gets saturated with number of connections close to 2 for all individual traffic flows. Once the saturation point is crossed, RT and BE throughput of both A and B nodes remain similar but with diminished value. However, throughput of node C continues to gain as it is positioned one level higher than A snd B nodes and hence gets more bandwidth. In DQBA (Figure **??**), throughput of both RT and BE traffic are streamlined even

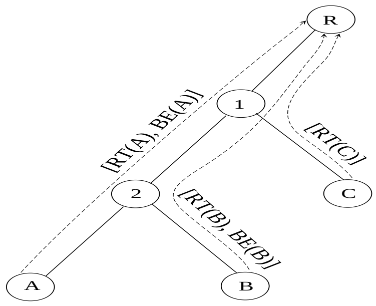
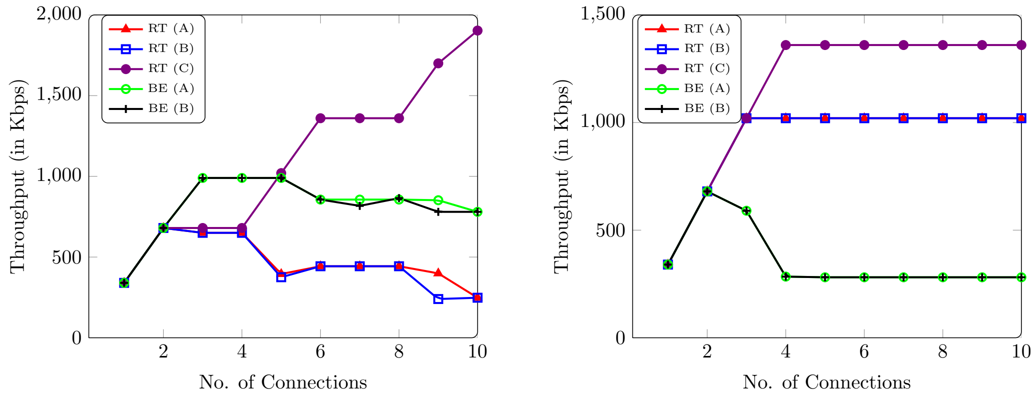


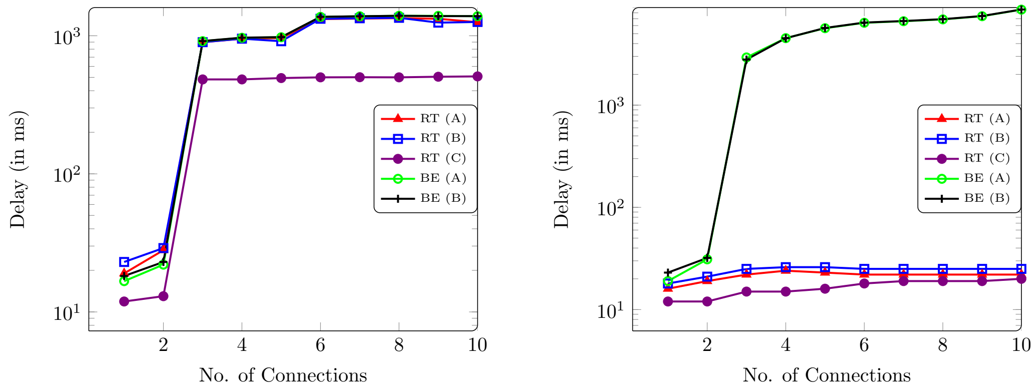
Figure 4-13: Network configuration for 3-hop scenario with uniform load from both the children of a sub-tree



(a) TreeMAC **(b)** DQBA

Figure 4-14: Throughput achieved by RT and BE traffic with uniform load from both the children of a sub-tree in a 3-hop topology

after the saturation point. RT traffic from node C shows better throughput than the others as it gets larger share being a 1-hop node of the root R.



(a) TreeMAC **(b)** DQBA

Figure 4-15: Delay of RT and BE traffic with uniform load from both the children of a sub-tree in a 3-hop topology

As shown in Figure **??**, 3-hop topology shows similar delay performance to 2-hop. In normal load, both the protocols show outstanding end-to-end delay character. However, as the load goes beyond the saturation point, delay for RT traffic reaches an unacceptable level. In DQBA (Figure **??**), the same has been improved to a great extent by compromising BE traffic performance.

4.5.2.2 With Skewed Real-time Traffic Load from a single Sub-tree

In our proposed protocol, a node needing more time slots for RT traffic can carry the unused time slots of other nodes. To simulate this setting, we have introduced Q-traffic and A-traffic from the nodes belonging to left sub-tree and only A-traffic from the right sub-tree.

Throughput and Delay Performance in 1-hop Scenario

Figure **??** presents the 1-hop network topology which comprises of two children nodes A and B and a parent node R. Both RT and BE traffic are introduced from A but only A-traffic is given from node B.

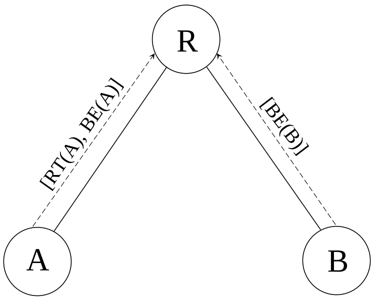
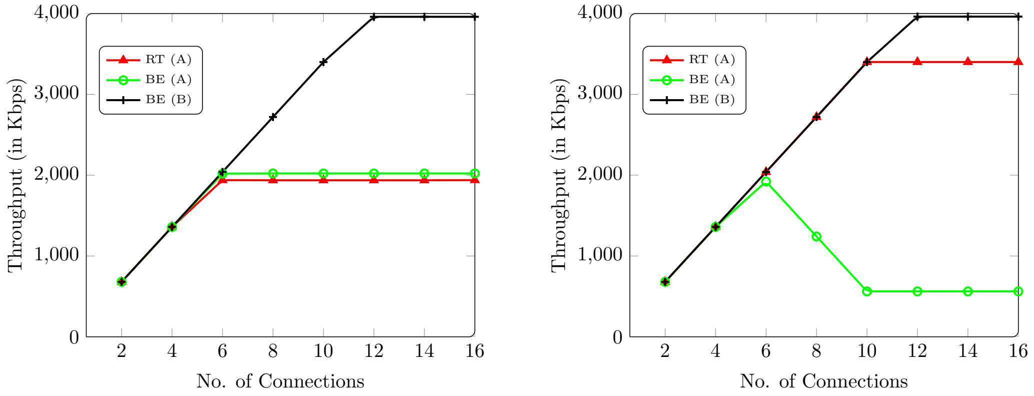


Figure 4-16: Network configuration for 1-hop scenario with skewed traffic load from single side of a sub-tree

Figure **??** presents the throughput performance of TreeMAC and DQBA. The bandwidth of $A \rightarrow R$ link is shared by the RT and BE traffic generated from A. BE traffic originated from B shows higher throughput as it is the only connection using the bandwidth of the link $B \rightarrow R$. Similar performance is also observed in DQBA (Figure **??**). When the link $A \rightarrow R$ is shared, we see from Figure **??** that the performance of real-time and best-effort traffic originated from the node A remains almost the same in TreeMAC. But with our scheme, the real-time traffic achieves better throughput than best-effort which is shown in Figure **??**. Once the link gets saturated, bandwidth of real-time and best-effort traffic settle according to their maximum allocated bandwidth share.

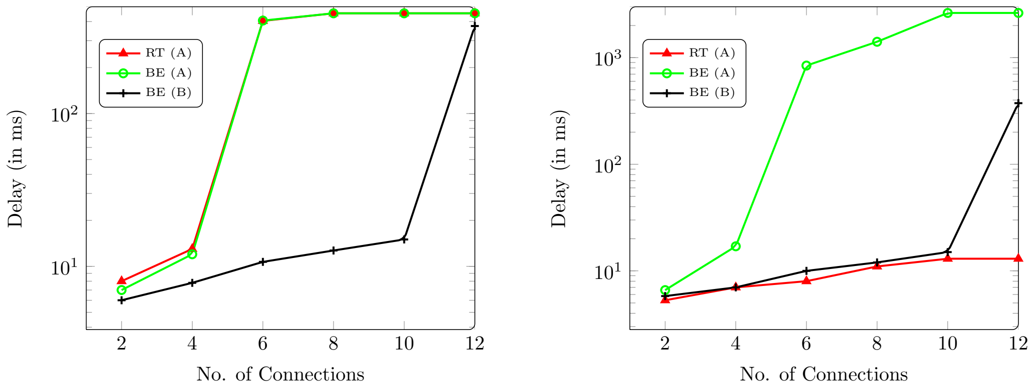
In TreeMAC, good delay performance is observed in low load situation as shown in Figure **??**. Beyond the saturation load, delay value quickly moves to a



(a) TreeMAC **(b)** DQBA

Figure 4-17: Throughput achieved by RT and BE traffic load from single side of a sub-tree in a 1-hop topology

very high level. It establishes the unsuitability of TreeMAC for real-time traffic in multi-hop WiLD networks. On the other hand, DQBA shows superior real-time traffic delay which is in the order of 13ms.



(a) TreeMAC **(b)** DQBA

Figure 4-18: Delay of RT and BE traffic with RT traffic load from single side of a sub-tree in a 1-hop topology

Throughput and Delay Performance in 2-hop Scenario

As shown in Figure **??**, node A transmits both RT and BE traffic whereas B transmits only BE traffic. In this topology and traffic pattern, it is interesting to see how DQBA utilizes the non-utilized time slots of the other sibling nodes.

Throughput performance of TreeMAC and DQBA in 2-hop topology under uneven load situation is presented in Figure **??**. Throughput performance of TreeMAC as shown in Figure **??** shows a steady increase of throughput in normal load. But in higher load, throughput curbs normally without showing any concern

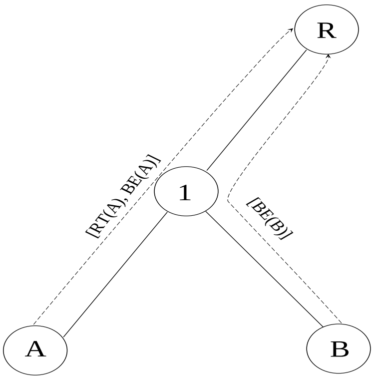
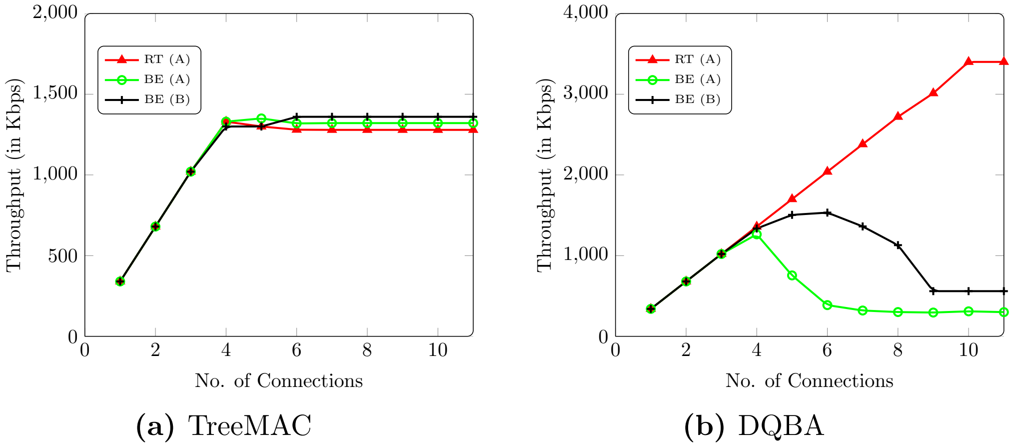


Figure 4-19: Network configuration for 2-hop scenario with skewed traffic load from single side of a sub-tree in 2-hop topology



(a) TreeMAC **(b)** DQBA

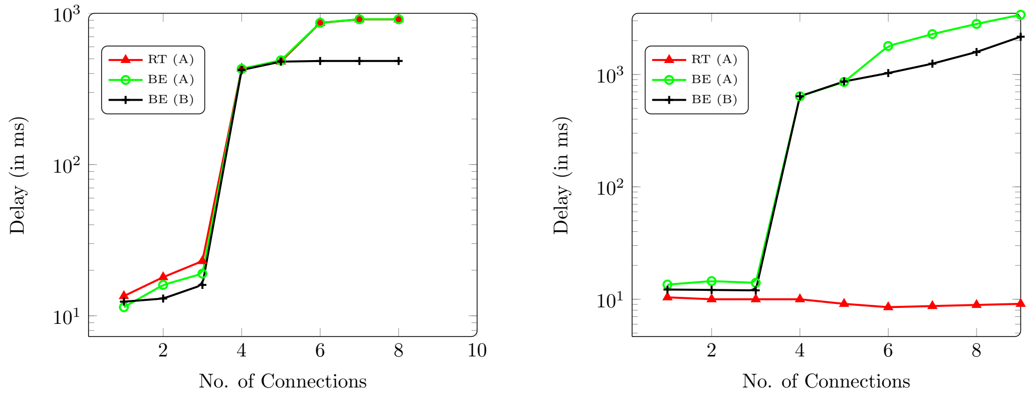
Figure 4-20: Throughput achieved by RT and BE traffic load from single side of a sub-tree in a 2-hop topology

for RT traffic. In this case, the maximum throughput achieved by RT traffic from node A is $1276 Kbps$. DQBA improves this figure up to $3400 Kbps$ which is shown in Figure **??**. It is exciting to observe that the unused time slots of node B are taken away by node A in our proposed protocol. A minimum of 20% time slots are kept reserved to avoid node starvation which is normally used by the best-effort traffic.

Delay performance of both the protocols are more or less similar to the 1-hop case. DQBA maintains a very small end-to-end delay even in high traffic load. Delay performances of both the protocols are demonstrated in Figure **??**.

Throughput and Delay Performance in 3-hop Scenario

In this experiment, we have considered the simulation topology and traffic pattern as shown in Figure **??**. RT traffic is introduced only from the node A whereas BE



(a) TreeMAC **(b)** DQBA

Figure 4-21: Delay of RT and BE traffic with RT traffic load from single side of a sub-tree in a 2-hop topology

traffic is added from all the three leaf nodes, i.e., A, B and C to R.

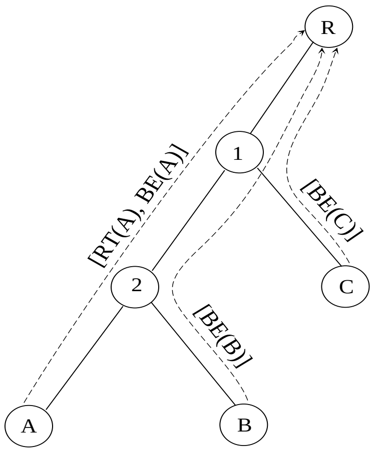
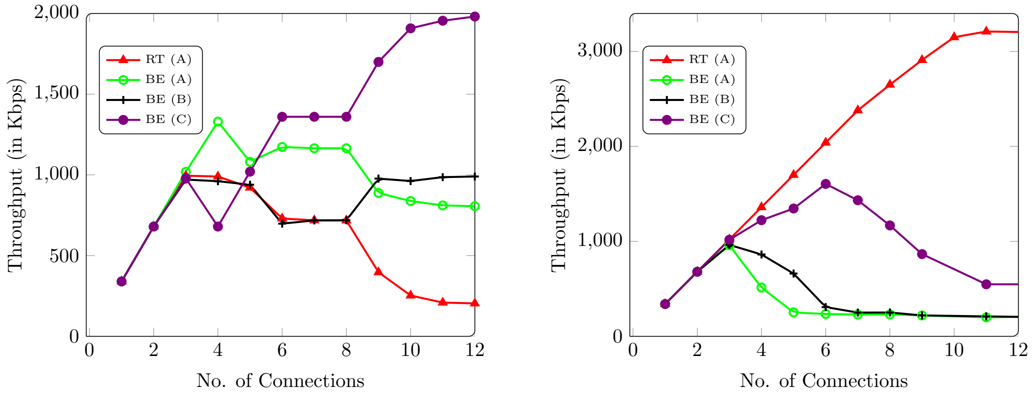


Figure 4-22: Network configuration for 3-hop scenario with skewed traffic load from single side of a sub-tree in a 3-hop topology

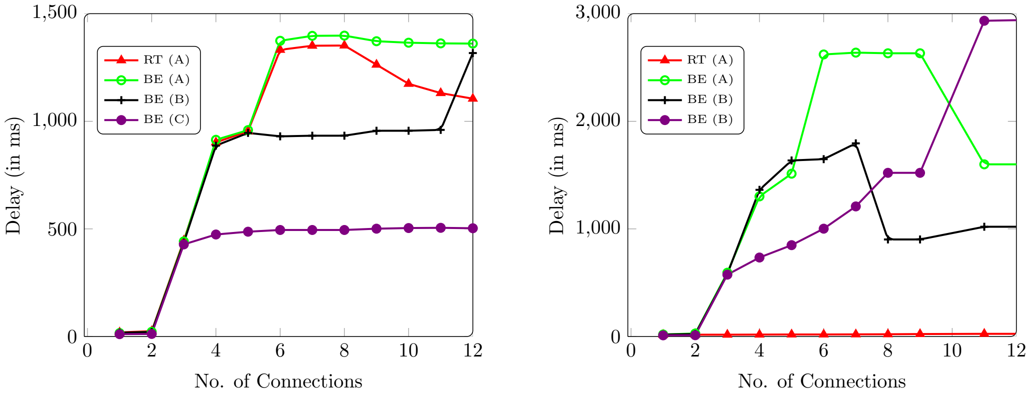


(a) TreeMAC **(b)** DQBA

Figure 4-23: Throughput achieved by RT and BE traffic load from single side of a sub-tree in a 3-hop topology

Throughput performance of TreeMAC in 3-hop topology having single side RT load is similar to 2-hop performance except that the throughput saturation occurs at a lower load (close to $900Kbps$). This is due the hop distance of the source node to the root and link sharing feature of multi-hop WiLD networks. In the proposed protocol, the throughput of real-time traffic increases with the increase in corresponding load whereas the throughput of best-effort traffic diminishes beyond the saturation point (Figure **??**).

Figure **??** demonstrates the delay characteristics of TreeMAC and DQBA in 3-hop scenario with RT traffic in one of the sub-trees only. Even with a very small number of connections, TreeMAC exhibits very high delay for all types of traffic (Figure **??**). In comparison, as shown in Figure **??**, DQBA provides much better delay performance for real-time traffic in the similar setting.



(a) TreeMAC **(b)** DQBA

Figure 4-24: Delay of RT and BE traffic with RT traffic load from single side of a sub-tree in a 3-hop topology

4.6 Conclusion

In this chapter, we have presented a dynamic QoS-aware bandwidth allocation scheme for multi-hop WiLD networks which addresses the congestion problem and hence facilitates QoS support for real-time traffic. The proposed dynamic slot scheduling mechanism efficiently distributes the unused bandwidth among the more needy nodes.Giving higher preference to the real-time traffic, the proposed protocol ensures end-to-end throughput and delay guarantees for the real-time traffic. A dedicated 20% bandwidth is reserved for best-effort traffic in each link to avoid node starvation. The simulation results show that the proposed protocol achieves a substantial improvement in throughput and delay of real-time traffic.

The contributions of this chapter can be summarized as follows:

- It solves the congestion problem which usually occurs close to the gateway node of multi-hop WiLD networks.
- The non-utilized time slots of a node are carried over multiple hops easily which provides a feel of almost using a dedicated link when less number of nodes operate in a network.
- In the best case, almost two times real-time connections can be supported than that of TreeMAC.
- Throughput and delay characteristics of real-time traffic has been improved significantly even in high load situation.
- A minimum of 20% bandwidth is reserved for each node which avoids node starvation problem.

In this chapter, we have presented a dynamic slot scheduling scheme to solve the congestion problem in multi-hop WiLD networks. In the next chapter, we discuss about a packet scheduling scheme for TDMA MAC protocols which supports fine-tuned QoS in WiLD networks.