Chapter 3

2C: A TDMA-based MAC Protocol for Multi-hop WiLD Networks

3.1 Introduction

Due to interference and several other constraints pertinent in multi-hop WiLD networks, overall network performance optimization is considered to be a challenging task. In resource-constrained WiFi-based long distance mesh networks, the protocols are expected to critically utilize the network resources in enhancing network performance. The MAC protocol plays a very important role in optimizing the utilization of available transmission links. Through spatial reuse, the wireless shared medium can be used optimally and thereby overall network performance can be boosted to a great extent. Improved network performance also facilitates better support for QoS-bound real-time applications such as video-conferencing, telemedicine, e-learning, VoIP (Voice over IP), etc. In multi-hop WiLD networks, transmissions of different radios located in the same tower are prone to interference. With proper setting and tuning of various parameters, simultaneous synchronous transmission (SynTx) or reception (SynRx) by all the co-located radios are achievable. They termed this operation as Simultaneous Synchronous Operation (SynOp). Based on this concept, 2P [?], WiLDNet [?], and JazzyMac [?] proposed distributed scheduling approaches to achieve interference-free link scheduling in WiLD networks. These protocols implemented the change of mode of operation between SynTx and SynRx by using a special packet called *marker packet*. To switch from SynRx to SynTx, a node is required to receive marker packets from all of its neighbour nodes. After completion of a node's transmission, the node sends its marker packet to all of its neighbours. Due to the dependency of marker packets, transmission of different links cannot be efficiently scheduled for transmission. Hence, these protocols fail to keep all the WiLD links busy all the time even if there is traffic to transmit. The real-time applications demand a certain level of QoS from the underlying networks. But, it is difficult to guarantee QoS in networks not providing any assured transmission opportunity to the nodes.

In this chapter, we have proposed a simple but efficient TDMA-based MAC protocol called *2C* (2-colour) for maximizing overall network efficiency through link scheduling. The nodes in a multi-hop WiLD network are logically arranged into a tree; the root node being the centralized coordinator. The main idea behind this MAC design is to enable SynOp at all nodes at all times. In the first phase of the scheduling algorithm, all the nodes of a network are synchronized with respect to a common time and colour. A colour represents either SynTx or SynRx operation of SynOp. The colour synchronization prepares the platform for link scheduling. The interference problem in WiLD network is addressed by colouring the network graph. In this case, we consider a tree topology which is 2-colourable. With the use of strict time and colour synchronization, the scheduling process schedules the transmission of all the nodes in the network in just two data slots, which provides a dedicated 50% slot reuse. As a result, 2C maximizes bandwidth utilization by achieving higher degree of dedicated slot reuse than any other relevant MAC protocols such as 2P [?], JazzyMac [?], and Lit MAC [?]. Fair transmission opportunity is also provided to all the nodes which improves end-to-end delay over multiple hops.

The rest of the chapter is organized as follows. Section **??** discusses the related works. A comparison of the state of art TDMA-based MAC protocols for WiLD network is presented in Section **??**. Section **??** gives the detail about the design of 2C protocol. It explains the network synchronization and node scheduling algorithms. In Section **??**, network synchronization overhead and saturation throughput for WiLD link have been theoretically calculated. Simulation and performance evaluation of 2C protocol is presented in Section **??**. It also gives a performance comparison of 2C and 2P protocols. Finally, Section **??** concludes this chapter.

3.2 Related Works

Many TDMA-based MAC protocols are found in the literature of WiLD networks [?, ?, ?, ?] addressing various performance issues. The main factor of low performance in multi-hop WiLD network is interference caused by the co-located radios [?]. With the proper setting of network parameters, SynOp by all the co-located radios can be attained [?]. Using the concept of SynOp, a number of TDMA-based MAC protocols are proposed. 2P [?], a loosely synchronized Two Phase TDMA-based MAC protocol which realized SynOp in a distributed manner. Synchronization between a node and its neighbours is achieved through the use of marker packet. WiLDNet [?] is built upon 2P with the enhancement of an adaptive loss-recovery mechanism that uses a combination of Forward Error Correction (FEC) and bulk acknowledgements to significantly reduce the loss rate and to increase the end-to-end File Transfer Protocol (FTP) throughput.

2P suggested that equal-sized static transmission slots for both SynTx and SynRx are beneficial for multi-hop traffic forwarding. However, in such situations, traffic of dynamic nature cannot be accommodated efficiently. With the use of dynamic slot adaptation based on traffic demands, JazzyMac [?] reported to have achieved superior throughput over static TDMA approaches like 2P [?] and WiLDNet [?]. The throughput of asymmetric traffic flows has been improved by providing longer transmission time to larger traffic loads. But, the benefit of dynamic slot adaptation is limited to a single hop topology only. Moreover, the use of dynamic slot does not provide any specific advantage in forwarding symmetric traffic over multi-hop networks. It can clearly be reckoned that the exchange of marker packet in 2P and 2P-based MAC protocols puts additional overhead during every phase change. Further, the loss of marker packet may leave a part of a network unsynchronized for a period of time until it is resolved; which may degrade the overall network performance. In marker packet based synchronization, the use of dummy filler bytes to maintain synchrony can be listed as another shortcoming of the protocol which puts additional overhead to the system operation leading to unnecessary power wastage. Dhekne et al. [?] proposed a TDMA-based MAC protocol for WiLD network which uses multi-hop time synchronization to synchronize the nodes in the network. But the protocol has lesser degree of slot reuse as the number of TDMA data slots which are assigned to each node is equal to the total number of slots available divided by the number of active nodes in the network. Thus, with the increase in number of active nodes, the data slots in a TDMA frame available for each node will decrease, resulting in performance degradation.

LiT MAC [?, ?] proposed a centralized scheduling scheme in which slot is allocated to a flow meeting the end-to-end delay requirement. It integrates a routing scheme and employs a flow based admission control mechanism at the MAC layer. To start a new flow, a node request time slot from the root node. After getting the request, the root node registers the flow and allocates time slot in the TDMA schedule for it. An Even-Odd link activation framework for WiMAX has been proposed in [?]. The Even-Odd framework employs a centralized approach which activates each link in alternate time slots and applies sub-channelization to adjust link bandwidths in order to allow access points to receive simultaneously over multiple links. The framework eliminates interference possibility and maps a wireless back-haul into an equivalent half-idle wire-line network. However, sub-channelization may not properly fit in WILD networks having a limited number of available channels.

3.3 Comparison of Existing TDMA-based MAC Protocols for WiLD Networks

Table ?? on page ?? provides a brief comparison of the existing TDMA-based MAC protocols proposed for WiLD networks. The protocols discussed in Section ?? (page ??) addresses various issues such as dynamic link scheduling, network performance optimization through tuning of protocol parameters, QoS provisioning, etc. From the comparative studies carried out, it can be observed that only LiT MAC has considered the end-to-end delay as parameter. The centralized admission control mechanism used in it justifies further scope of work considering end-to-end throughput and delay as parameter. Timely forwarding of packets by the intermediate nodes is crucial for QoS provisioning in multi-hop networks. This motivates us to design an efficient MAC scheme that we will discuss in the subsequent sections.

3.4 The 2C Protocol

2C is a TDMA-based MAC protocol designed for multi-hop WiLD networks. The design is greatly influenced by the 2P MAC protocol [?]; inheriting the concept of SynOp from it. 2C eliminates the use of loose synchronization which is available in 2P-based protocols. The aim of 2C is to improve the performance of multi-hop WiLD networks by scheduling interference-free overlapped transmission of nodes as much as possible. The proposed protocol implements a link scheduling algorithm which uses a colouring scheme to colour all the nodes of the network.

A multi-hop WiLD network can be considered as a tree represented as $T(V, E)$ where V is the set of nodes and E is the set of point-to-point WiLD links. Set V consists of three types of nodes- root node, relay node and leaf node. Let, L be the total number of levels and l be any arbitrary level in the network. The i^{th} node in the l^{th} level is denoted by $n_{i,l}$. $(n_{i,l}, n_{j,l-1})$ and $(n_{i,l}, n_{k,l+1}) \in E$ denote the links from the i^{th} node in level l to the j^{th} and k^{th} node in level $l-1$ and $l+1$ respectively. The root node, R takes the responsibility of initialization and synchronization of the whole network with respect to time and colour. This synchronization process is periodically carried out once the network is initialized. R accomplishes the task of constructing TDMA frames, generating control packets and sends it to its first hop neighbours. A *relay node* receives control packets, synchronizes itself with the immediate higher level node and carries out node joining processes for the nodes belonging to its next level. *Leaf nodes* are the end-points of the backhaul network which receives control packets, generates node join request, assigns colour to itself and synchronizes its clock. The assumed network architecture is depicted in Figure **??**. There may be more than one root node for higher redundancy and capacity, but we focus on single gateway-based network in our protocol.

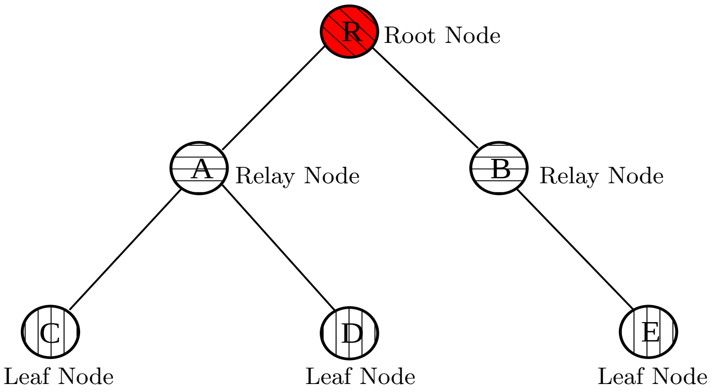


Figure 3-1: Types of Nodes in WiLD Network Architecture

3.4.1 Protocol Assumptions

The 2C protocol makes the following assumptions:

(i) **Synchronous Operation (SynOp)**
A node can perform SynTx or SynRx operation using the radios installed in it.

As discussed in Section **??** of Chapter **??**, each node in WiLD network employs a separate radio for each point-to-point link. Such links suffer from side-lobe interference and cannot operate independently in the same channel [**?**]. Raman et al. [**?**] suggests that SynOp is possible by any node in multi-hop WiLD network. However, mix of transmit and receive operation is not feasible due to the physical proximity between the radios and the presence of antenna side-lobes. SynTx and SynRx operations are feasible if the angular difference between any two adjacent directional point-to-point links are greater than a given value θ. Value of θ depends on network parameters such as Signal to Interference and Noise Ratio (SINR) which is in turn dependent on beam width and side lobe pattern of the antennas used in the link. The concept of SynRx, SynTx and mix of receive and transmit (Mix-Tx-Rx) operations related to SynOp are diagrammatically explained in Figure **??** (adopted from [**?**]).

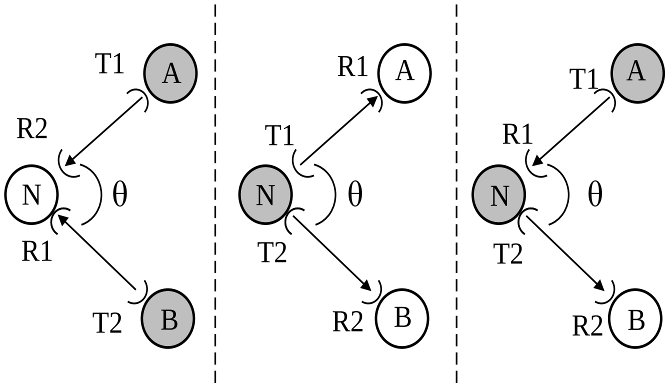


Figure 3-2: Simultaneous Synchronous Operation (SynOp)

(ii) **Tight Node Synchronization**
The protocol assumes that it is possible to achieve tight synchronization among all the nodes available in a multi-hop WiLD network. However, due to the existence of clock drift and lack of perfect synchronization mechanism, nodes in the network are required to be re-synchronized periodically.

(iii) **Topology Dependence**
Correct operation of SynOp requires the topology to be bipartite. This constraint is because of the fact that at any instant of time if a node is in SynTx mode, all its neighbours have to be in SynRx mode and vice-versa. This implies that the topology should not have any odd cycles, i.e., it should be bipartite.

iv) **Single Channel Approach**
In multi-hop WiLD networks, nodes are equipped with multiple radios and thus allowing the use of separate non-overlapping channels on each link. However, getting inspired from 2P, WiLDNet, and JazzyMac, we consider the use of single channel in our design. Reserving a single channel for the WiLD network reduces the effort of channel assignment. Further, the remaining channels can be used by any local access links.

3.4.2 TDMA Frame Format and Packets Used

The standard TDMA frame and packet formats are redefined to serve various purposes of the proposed protocol. These are as follows.

TDMA Frame

A TDMA frame comprises of control slots, contention slots and data slots. Each frame has got a maximum limit for each type of slots. *Control slots* are used to transmit control packets. The number of control slots needed to disseminate the control packets to all the nodes of a network is equal to the height, h, of the topology tree. *Contention slots* are designed to be used only for the non-root nodes (i.e, relay and leaf nodes) for sending node join request. The number of contention slots in a frame is equal to the number of control slots. In a particular contention or control slot, all the nodes corresponding to a given level of the tree get the same slot at the same time. *Data slots* are used for actual data flow across the network. A typical TDMA frame structure is shown in Figure **??**.

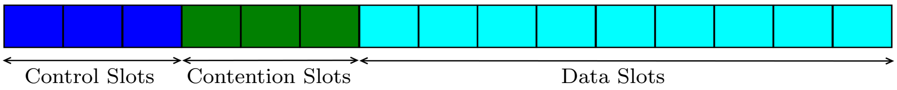


Figure 3-3: TDMA Frame Format

In every TDMA time slot, a guard band of about 100 μs is used. Actual transmission time of a slot is calculated by subtracting the guard band size from the slot size. These actual TDMA slots are allocated to various nodes for the purpose of transmitting control and data packets. A guard band of 100 μs is sufficient for solving offset in time synchronization of two neighbouring nodes.

Packet Types

Four different types of packets viz., control packet, node join packet, registration confirmation, and data packet are used in our protocol. These packets use customized 802.11 header formats to serve the protocol requirements. *Control packets* are exclusively generated by the root node to disseminate control information to the other nodes (relay and leaf nodes). The control packet contains multi-hop synchronization information, information like TDMA frame length, slot length and the transmission schedule for control and contention slots. The main purpose of control packet is to synchronize the non-root nodes with the root node, assign colour to the current node and help determine their parent in the tree. On the other hand, the *node join packets* are transmitted during a contention slot and are used by those nodes that want to join the network. This type of packet contains two additional fields- Parent Index (PI) and Node Index (NI). PI is the identification of the node to which a node wants to join. NI represents the identification of the node wanting to join the network. *Registration confirmation packets* are used to send node joining confirmation message. It uses control slots for transmission. *Data packets* contain the actual network layer data that need to be transmitted across the network.

3.4.3 The Scheduling Algorithm

In 2C, the link scheduling process is dynamically carried out at each node independently. The effectiveness of the scheduling process greatly relies on tight synchronization among all the nodes of the network. In view of this, the scheduling scheme is designed to work in two phases: *Node Synchronization* and *Link Scheduling*. The synchronized network created by the node synchronization algorithm is used by the link scheduling algorithm to schedule link's transmission. The node synchronization and link scheduling algorithms are discussed below.

3.4.3.1 Node Synchronization

2C synchronizes all the nodes of the network with respect to time and colour by using an interference-aware node colouring algorithm. A colour represents the mode of operation of a node. Let us assume that the colours C_1 and C_2 represent SynTx and SynRx operations respectively. Initially, the node colouring algorithm (Algorithm **??**) clears the existing colours of all the nodes if any. In the first step, the root node, R is assigned a colour, say C_1. Node R time stamps its current time into a control packet and transmits it along with the assigned colour to all of its neighbour nodes. After receiving the control packet, the nodes synchronize themselves with the given time and assign the second colour, i.e., C_2. This ensures that the colour of the root node and its adjacent nodes, i.e, nodes of first level of the tree are not the same. In a similar fashion, colour and time stamp value propagates level by level in such a way that no two adjacent nodes belonging to consecutive levels are assigned the same colour. Finally, the algorithm terminates when all the nodes in the network are synchronized and assigned a colour.

Let us assume that the colour of a node i belonging to level l is given by $n_{i,l}.colour$. For the links $(n_{i,l-1}, n_{j,l}), (n_{j,l}, n_{k,l+1}) \in E$, while assigning a colour to $n_{j,l}$, the colouring algorithm sets the constraints- (i) $n_{i,l-1}.colour \neq n_{j,l}.colour$, and (ii) $n_{i,l-1}.colour = n_{k,l+1}.colour$. That means, if $n_{i,l-1}.colour = C_1$ then $n_{j,l}.colour \neq C_1$. Since, a tree topology is 2-colourable; the algorithm can colour the entire network using only two colours. The algorithm mitigates the self-interference problem if the nodes with different colours are provisioned with non-overlapping transmission slots. In a network, if X is the number of nodes having same colour C_1 at any instant of time, it indicates that X number of nodes can transmit simultaneously and the remaining $(V - X)$ number of nodes will remain in receive mode.

3.4.3.2 An Example Showing Node Synchronization Procedure

The process of network synchronization with respect to time and colour is explained in Figure **??**. For the sake of simplicity, a simple 2-hop network topology is considered. Initially, the root node R generates a control packet with a frame consisting of three slots: two control slots and one contention slot. During the first control slot, R sends a node join advertisement to its neighbours (Figure **??**). Each node, receiving this control packet, synchronizes its clock with R and assigns the opposite colour of R to itself. After getting synchronized, all the nodes

Algorithm 1 Node Colouring Algorithm

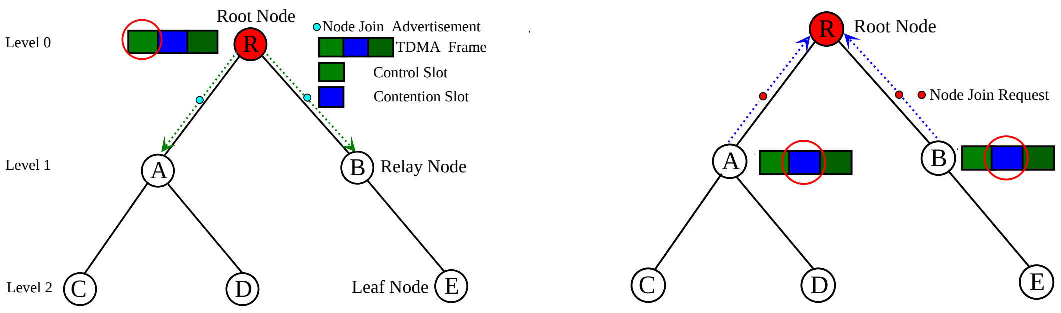
Input:
R: Root Node
V: Set of Vertices
ST: Stack

```
Assign Colour $C_1$ to $R$
Empty the stack $ST$
Push $R$ into $ST$
while $S$ is not empty do
    Pop $ST$ and assign the value to $u$
    for all $V_k$ adjacent to $u$ do
        if $V_k$ is not coloured then
            if $u$ coloured with $C_1$ then
                Assign colour $C_2$ to $V_k$
            else
                Assign colour $C_1$ to $V_k$
            end if
            Push $V_k$ into $ST$
        end if
    end for
end while
```

of level-1 send node join request to R during the contention slot (Figure **??**). On receiving node join requests, R confirms node joining by sending a registration confirmation message in the second control slot as shown in Figure **??**. With this, synchronization process of level-1 nodes gets completed. In a similar way to the root node, level-1 nodes perform node joining of level-2 nodes. For this purpose, all the level-1 nodes prepare a control frame comprising of two control and one contention slots. Now, in its first control slot, each of the level-1 nodes send node join advertisements to their neighbour nodes (Figure **??**). On receiving the node join advertisement, the level-2 nodes sends node join request to level-1 nodes in the next contention slot as shown in Figure **??**). Finally, level-1 nodes send registration confirmation message to the level-2 nodes in the second control slot. In this fashion, all the nodes of a network are synchronized level by level. Data packets containing the actual network layer data can then be transmitted in the subsequent data slots.

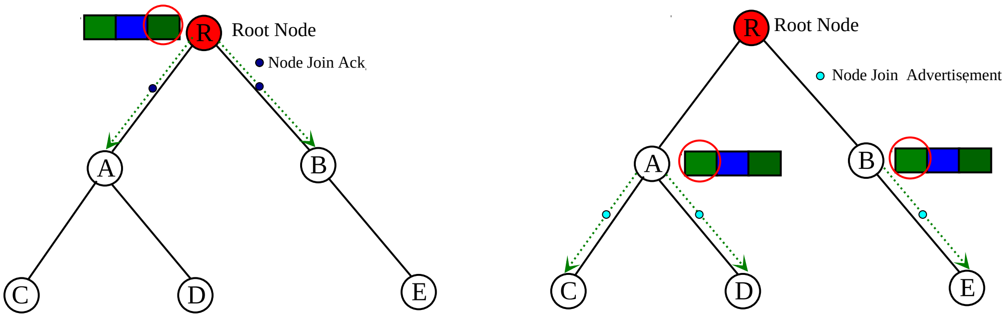
3.4.3.3 Link Scheduling

After getting all the nodes synchronized with respect to time and colour, the link scheduling algorithm schedules the SynTx and SynRx operations in alternate data



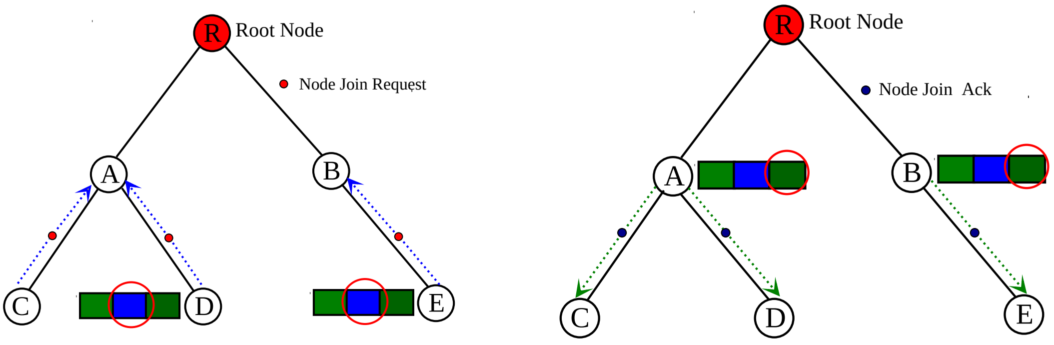
(a) The Root node sends Node Join Advertisement

(b) Level-1 nodes send Node Join Request to R



(c) Node R sends Registration Confirmation Message to Level-1 nodes

(d) Level-1 nodes send Node Join Advertisement (for Level-2 nodes)



(e) Level-2 nodes send Node Join Request to Level-1 nodes

(f) Level-1 nodes Registration Confirmation Message to Level-2 nodes

Figure 3-4: Network Synchronization in 2C: An Example slots. Each node in the network uses the scheduling algorithm (Algorithm **??**) to schedule their transmission dynamically.

The algorithm works as follows. When a node encounters a data slot, the algorithm first checks its colour. If the colour is found to be C_1, the node changes its mode to SynTx and starts transmission immediately. On the other hand, if the colour is C_2, it changes its mode to SynRx and waits for incoming packets. At the end of each data slot, the colours of all the nodes get inverted. Thus, transmission is scheduled in alternate data slots only.

Algorithm 2 Link Scheduling Algorithm

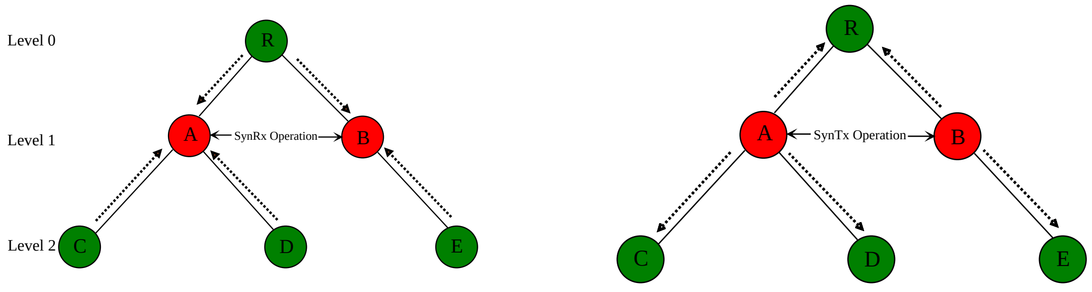
```
Input:
$V$: Set of Vertices
$TF_r$: TDMA Frame

for all Data Slots, $S_i \in TF_r; 1 \leq i \leq p$ do
    if $V_j$ coloured with $C_1$ then    # $C_1$: A Colour
        Switch $V_j$ to SynTx and send data
    else
        Switch $V_j$ to SynRx
    end if
    Wait for end of slot
    INVERT($V_j$)
end for

procedure INVERT($V_j$)
    if $V_j$ coloured with $C_2$ then    # $C_2$: A Colour
        Invert the colour of $V_j$ to $C_1$
    else
        Invert the colour of $V_j$ to $C_2$
    end if
end procedure
```

Assume that in a TDMA frame TF_r, there is a set of data slots S, $S = \{S_1, S_2, S_3, ..., S_p\}$ with cardinality p. A link scheduling algorithm is a mapping from the set of links E to the set of slots T, where $T \subseteq S$. Here, transmission of all the nodes can be accommodated by spatial reuse. The main concern of a link scheduling algorithm is to increase the degree of mapping and reduce the cardinality of T.

In this case, the cardinality of T is equal to the number of colours required to colour the entire network (which is 2 with the use of the interference-aware colouring algorithm). And, the degree of mapping is equal to L as all links can remain active at all times by performing SynTx in level, l and SynRx in level,



(a) Node's State during Data Slot t

(b) Node's State during Data Slot $t+1$

Figure 3-5: Node's State Transition between SynTx and SynRx

$l-1$ and $l+1$ simultaneously for $l = \{1, 2, ..., L-1\}$. Therefore, the scheduling algorithm requires two slots to schedule the transmission of all the links. The set S can be partitioned into S^{odd} and S^{even} such that $S^{odd} \cap S^{even} = \phi$. The scheduling algorithm sets an operational constraint on each link while scheduling transmissions. Any link, $(n_{i,j}, n_{k,q}) \in E$ having transmission $n_{i,j} \rightarrow n_{k,q}$ in S^{odd} switches to $n_{k,q} \rightarrow n_{i,j}$ in S^{even}. Thus, the link scheduling algorithm ensures that every alternate transmission slot is allocated to each node. If X number of nodes having colour C_1 transmit during the slots in S^{odd} then the remaining $(V - X)$ number of nodes coloured with C_2 will transmit during the slots in S^{even}, thus avoiding interference.

3.4.3.4 An Example Showing Link Scheduling Procedure

Figure **??** provides a brief insight into the working principle of the link scheduling procedure. As shown in Figure **??**, all the nodes with colour C_1 are in SynTx and the remaining nodes with colour C_2 are in SynRx mode in the first data slot. The protocol ensures that if one end of a link is in transmitting mode then the other end must be in receiving mode. The traffic originating from node R, C, D, and E are transmitted in the same slot which attributes to slot reuse. Hence, multiple links actively take part in parallel transmission resulting in maximum bandwidth utilization.

Subsequently, in the next data slot as shown in Figure **??**, all the nodes will switch their colours, i.e., the nodes previously in transmitting mode will switch over to receiving mode and vice-versa. The traffic received by the nodes A and B from nodes R, C, D, and E during the first slot are now forwarded towards their destinations during the immediate next slot. Assured forwarding of traffic in the immediate next slot ensures improved end-to-end throughput and delay over multiple hops.

3.5 Theoretical Analysis

In this section, we theoretically analyse the parameters: (i) the time required for synchronizing the entire network and completing the joining process for all the nodes and, (ii) the saturation throughput of $11 Mbps$ WiLD link in our setting.

3.5.1 Network Synchronization and Node Joining Time

As discussed in Subsection **??**, the root node R initiates the process of network synchronization by sending a node join advertisement to its immediate neighbours. All the level-1 nodes receiving the control packets synchronize themselves with R and send node join requests in the subsequent contention slot. R registers the first hop nodes and sends registration confirmation message through the next control slot. Once the first hop nodes are registered, they can register the nodes of immediate next level. Subsequently, when the second hop nodes receive node join advertisement from the first hop nodes, they send node join request and get registered with level one nodes. After completing the registration of the second hop nodes, level one nodes send registration confirmation message through the next control packet. This process continues until nodes of all the levels join the network. The number of slots required for joining nodes per hop or level is three. Two control slots are required for transmitting node the join advertisement and registration confirmation message and one contention slot is needed for the transmission of node join request message. Hence, it is clear that the number of slots required for the node joining process increases with the increase in the height of the topology tree at the rate of 3 slots per hop. Therefore, the minimum number of control and contention slots required to fully synchronize a network having height h, denoted as N^h_{sync} can be given as-

$$N^h_{sync} = 3 \times h$$

Considering a time slot, t_s to be comprised of transmit time and guard time, the minimum time required to fully synchronize the network, NS_{time} can be given by

$$\begin{aligned} NS_{time} &\approx N^h_{sync} \times t_s \\ &\approx 3h \times t_s \end{aligned} \tag{3.1}$$

Therefore, from equation (??), it is clear that

$$NS_{time} \quad \propto \quad h \tag{3.2}$$

where t_s is considered to be constant.

It is observed that the value NS_{time} is directly dependent on the height of the topology tree h. In case of transient link failure and loss of synchronization, the affected nodes can rejoin the network in the next TDMA frame.

3.5.2 Saturation Throughput

As discussed in Section **??**, any one of the two nodes connected by a link can transmit when its colour is C_1 during the start of any data slot. When the current slot is over, the transmitting node immediately inverts its colour. In this process, each node gets every alternate data slots of the TDMA superframe. Assume that there are 100 numbers of data slots in a TDMA frame. Out of the 100 available slots, the number of usable data slots each node gets is $100/2 = 50$ per frame. The number of packets that can be sent during a data slot depends on size of the time slot used and the size of the packets accommodated therein. Table **??** shows the transmit time for the various parts of a packet except the Physical Layer Convergence Protocol (PLCP) header, at $11Mbps$ data rate. The transmit time for PLCP header is calculated at $2Mbps$ rate. Equation (**??**) shows that the theoretical saturation throughput for a WiLD link with a slot size of $5ms$ and guard time of $100\mu s$ is $4.48Mbps$.

Table 3.1: Time Required to Transmit Various Portions of a Packet

Description	Size($Bytes$)	Time (μs)
Payload	1400	1018.18
IP Header	20	14.54
TDMA MAC Header	30	21.81
CRC Header	4	2.90
PLCP Header	24	96
Total	1478	1153.43

Given,

$$\begin{aligned}
Packet\ Size, Packet_{size} &= 1400\ bytes \\
Slot\ Size, Size_{t_s} &= 5ms\ =\ 0.005s \\
Guard\ Time, G_t &= 100\mu s \\
Total\ no.\ of\ Data\ Slots &= 100
\end{aligned}$$

The number of data slots per frame per node, S_f is

$$S_f \quad = \quad \frac{100}{2} = 50$$

The effective transmission per slot, TS_{eff} is

$$TS_{eff} \quad = \quad Size_{t_s} - G_t = 4900\mu s$$

$$\begin{aligned}
Number of packets/slot, NP_s &= 4900/1153.43 \\
&= \lfloor 4.28 \rfloor \\
&= 4
\end{aligned}$$

$$\begin{aligned}
Frames/sec, F_r &= \frac{1}{Total\ slots \times Slot\ time} \\
&= \frac{1}{100 \times 0.005} \\
&= 2
\end{aligned}$$

We know that,

$$\begin{aligned}
Packets/sec, P &= F_r \times S_f \times NP_s \\
&= 2 \times 50 \times 4 \\
&= 400
\end{aligned}$$

Therefore,

$$\begin{aligned}
Throughput &= P \times Packet_{size}\ bytes/sec \\
&= 400 \times 1400\ bytes/sec \\
&= 4.48\ Mbps \qquad (3.3)
\end{aligned}$$

3.6 Simulation and Performance Evaluation

In this section, the performance of our proposed protocol is evaluated and compared with 2P [?]. We have considered 2P [?] for performance comparison as it is the closest protocol of 2C and many other protocols such as [?,?] are also designed based on it. 2P is a distributed protocol which schedules links transmission based on the status of its neighbouring nodes. On the other hand, in 2C, the nodes are initially synchronized with the root node first and then link transmission is scheduled by the nodes individually. The TDMA frame used in this purpose consists of 100 numbers of data slots. Number of control slots and contention slots in a frame varies depending upon the height of the tree. The TDMA slot size ranges from $2ms$ to $30ms$ with a guard time of $100\mu s$.

First, we conduct experiments to find out the optimal slot size for various applications. Using the optimal slot size found in this experiment, the throughput and delay performance of various types of traffic are measured in the subsequent experiments.

3.6.1 Performance Metrics and Simulation Parameters

For performance evaluation, we have considered the following metrics-

(i) Optimal Slot Size (SS_{opt}): It is the size of a TDMA slot for which the network provides the best performance. It is usually given in milliseconds.

(ii) 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 quality in a network.

(iii) 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 very important parameter for delay sensitive real-time traffic.

(iv) Network Synchronization Time (N^{h}_{sync}): This parameter measures the time required to synchronize a network (tree topology with height h). In TDMA-based networks, time required to synchronize a network and accuracy of synchronization derive the guard time between slots which greatly impacts on network performance.

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

Table 3.2: Simulation Parameters for 2C

Parameter	**Value**
Types of Traffic	CBR, FTP
Packet Size	$1400 bytes$ (Payload)
Packet Arrival Pattern (CBR)	Poisson
Routing Protocol	Fixed Routing Protocol
Radio Propagation Model	Two Ray Ground Reflection Model
Simulation Area	$50kms \times 50kms$ Flat-grid Area
Bandwidth	$11Mbps$
Antenna Type	Grid Parabolic Antenna
Antenna Gain	$24dBi$
Hop Distance	$9kms$
No. of Nodes (Max.)	24
Guard Time	100 μs
TDMA Queue Length	200

The simulation topology comprises of 24 nodes forming a tree with links extending upto a distance of $9kms$. Constant Bit Rate (CBR) and File Transfer Protocol (FTP) traffic flows are used to carry out the throughput and delay analysis. UDP flows generated by CBR traffic with packet size $1400bytes$ with poisson arrival rate is used to saturate the bandwidth of a $11Mbps$ WiLD link. On the other hand, the TCP flows are generated by a FTP traffic with packet size of 1400 *bytes* which is sufficient enough for saturating a link. Table **??** presents the various considered simulation parameters along with their values. The simulation is carried out for a duration of 300 *seconds*.

3.6.2 Results Analysis

In this section, we have presented the results of various experiments conducted while evaluating the performance of 2C and compared with 2P. The experiments and their results are presented one by one which are as follows.

3.6.2.1 Optimal TDMA Slot Size

Smaller slot size is known to be better for lower delay and jitter but bad for throughput efficiency. To find the optimal TDMA slot size for various applications over multi-hop WiLD networks, the following two experiments are performed.

Throughput vs. Slot Size

FTP throughput performance is measured for different TDMA slot sizes varying from $2ms$ to $30ms$ in single as well as multi-hop scenarios. As shown in the Figure **??**, after showing consistent performance for smaller slot sizes, a clear declination in FTP throughput is observed with higher slot sizes for both 2P and 2C MAC protocols. The throughput is adversely affected over multiple hops which is due to higher end-to-end delay in receiving acknowledgement packets. 2C outperforms 2P in single as well as in multiple hop cases employing different slot sizes. Highest throughput is seen with TDMA slot of size ranging between 3 to 5 milliseconds for 1-hop, 4-hop, and 6-hop cases. When slot size is increased beyond $5ms$, throughput is observed to be declining.

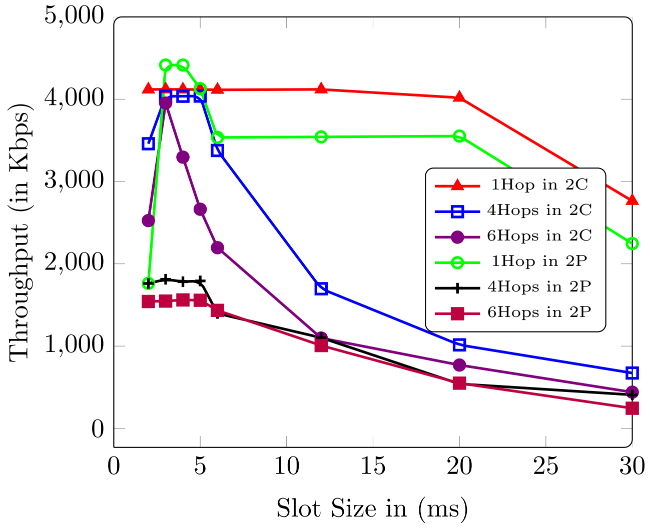


Figure 3-6: FTP Throughput with increasing Slot Size: 2C vs. 2P

Delay vs. Slot Size

To evaluate the impact of slot size over delay, we have considered VoIP application as it demands strict delay for its proper functioning. This experiment is conducted varying the slot size between $2ms$ to $40ms$ with 35 numbers of VoIP connections.

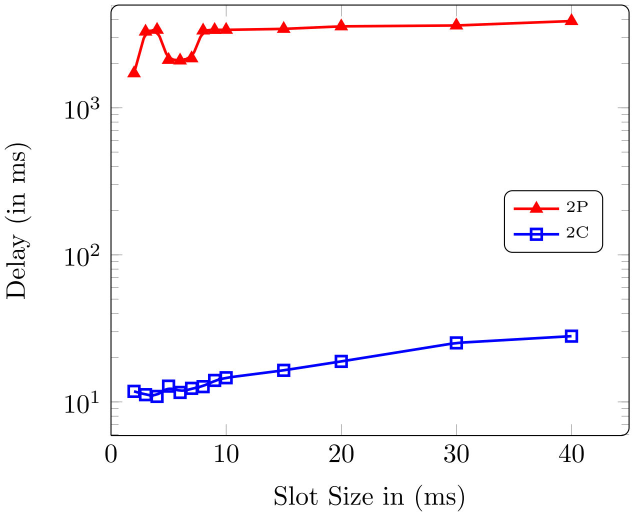


Figure 3-7: VoIP Delay with increasing Slot Size: 2C vs. 2P

The distance between the source and the destination node is considered constant at 4-*hops*. It can be observed from the Figure **??** that with the increase in slot size, delay increases proportionally.

From the above two experiments, it has been observed that the optimal throughput is achieved with TDMA slots of size ranging between 3 to 5 milliseconds whereas the delay value keeps on increasing continuously with the increase in slot size but well within required limit. Therefore, we have used slot of size $5ms$ size for the following experiments.

3.6.2.2 Throughput and Delay Characteristics

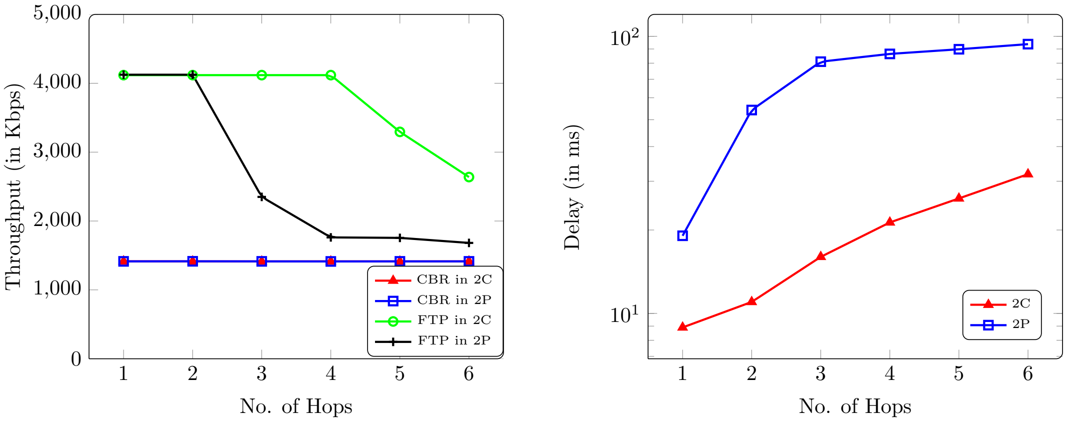
This part of experiments are conducted in two scenarios- at *normal load* and *saturated load*. These experiments aim to observe the impact of distance between the source and destination nodes over end-to-end throughput and delay. Here, we consider source-destination distance in terms of number of hops. To measure the throughput characteristics, we first analyze the throughput for CBR and FTP traffic by varying the number of hops. Throughput and delay is calculated only after the network is initialized properly. Throughput characteristic is observed for number of hops ranging from 1 to 6 with TDMA slot size of $5ms$ and guard time of $100\mu s$.

At Normal Load

In this section, we have evaluated the throughput and delay characteristics of 2P and 2C protocols in normal load situation. End-to-end delay is observed with varying path length in terms of number of hops.

Throughput

CBR and FTP throughput performance at normal load are presented in Figure **??**. CBR throughput of both 2P and 2C are seen to be very close for all the cases. However, difference is observed in FTP performance of both the protocols. Up to 2-hops distance, both 2C and 2P display more or less similar throughput. But, beyond that point FTP throughput in 2P suddenly falls below half whereas 2C continues to perform consistently up to 4-hops. A considerable drop in FTP throughput is observed in 5-hop scenario. The reason for difference in multi-hop throughput performance is marker packet dependency of 2P because of which the acknowledgement packets do not get immediate chance for transmission in the intermediate nodes over multi-hop network topology.



(a) Throughput **(b)** Delay

Figure 3-8: Throughput and Delay Characteristics at Normal Load: 2C vs. 2P

Delay

Figure **??** shows that delay of CBR traffic in 2C is well within the delay bound for delay-sensitive traffic. A tolerable increase in delay is observed with increasing number of hops. On the other hand, 2P shows in abysmal delay performance with more number of hops. In 2C, maximum delay recorded is about 31.8ms in the case of 6-hop distance. The reception and transmission in 2C occurs during the consecutive time slots which ensures strict end-to-end delay over multiple hops. But, due to marker packet dependency in 2P, packets could not be forwarded immediately leading to increase in end-to-end delay.

At Saturation Load

In this section, we have evaluated the throughput and delay characteristics of 2P and 2C protocols in saturated load situation. The other settings of these experiments are same as normal load.

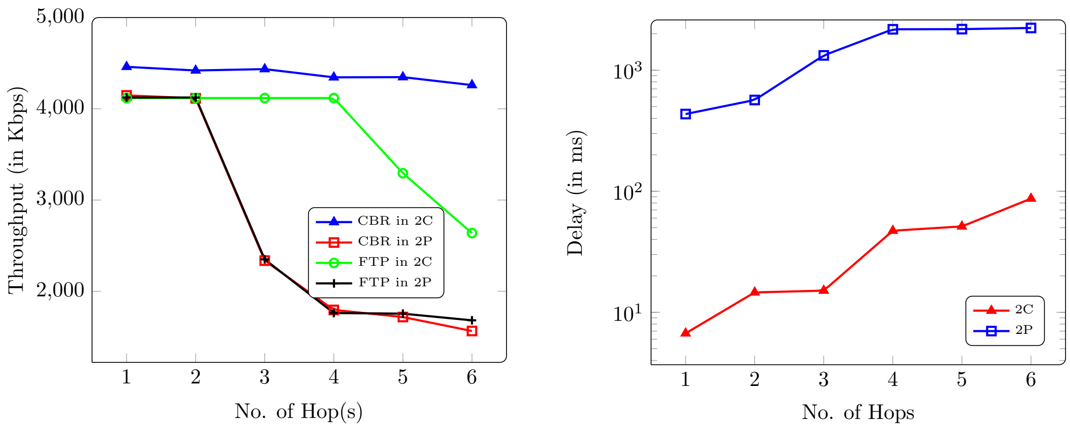
Throughput

From the Figure **??**, it can be observed that the CBR throughput does not show much declination in 2C with increase in hop distance. On the other hand, the throughput of 2P falls drastically after 2 hops. As the 2C protocol guarantees every alternate slots to each node, the end-to-end throughput has not reduced significantly over multiple hops. FTP throughput in 2P is shown to be very similar with CBR. But in 2C, the same decreases linearly as compared to CBR throughput over multiple hops. This is because of the fact that an increase in the number of hops increases the round trip delay and end-to-end error probability.

For a single WiLD link, both 2C and 2P protocols show saturation throughput of 4.46 and 4.15$Mbps$ respectively. It is apparent that 2C offers much better throughput than 2P. This value is close to the maximum achievable throughput (4.48$Mbps$) as computed in Section **??**.

Delay

Similar to normal load situation, a high divergence in delay performance between 2P and 2C is noticed. Beyond 3-hops distance, end-to-end delay in 2P is observed to be exceptionally high (in the range of 2 *Seconds*). It indicates that delay-bound real-time traffic over multiple hops may not suit 2P. In 2C, delay is found to be well within the prescribed limit (as can be seen in **??**).



(a) Throughput **(b)** Delay

Figure 3-9: Throughput and Delay Characteristics at Saturated Load: 2C vs. 2P

3.6.2.3 Network Initialization Overhead

In this section, we have evaluated network initialization overhead of 2C protocol. Network initialization overhead refers to the time required in registration and synchronization of all the nodes with the gateway. Network initialization overhead is measured in terms of time taken to complete the node joining process which encompasses both registration and synchronization activities. This experiment is carried out for tree topology with depth 1 to 6.

No. of Hops vs. Node Joining Time

In 2C, the gateway node of WiLD network initiates the node joining process. The process of joining nodes from different levels incurs certain delay over multiple hops. The overhead in the node joining process is analyzed keeping TDMA slot size constant at $5ms$. Figure **??** shows the node joining time for different hop distances from the root node. From the figure, it is apparent that node joining time linearly increases with the increase in number of levels of the topology tree. It validates the theoretically estimated node joining time as per Equation (**??**) on page **??**. This parameter is not measured for 2P as it is a distributed protocol and hence this feature is not applicable to it.

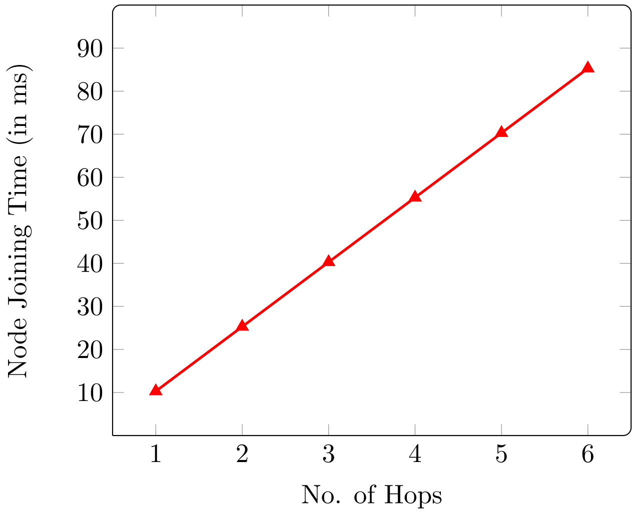


Figure 3-10: No. of Hops vs. Node Joining Time in 2C

3.7 Conclusion

In this chapter, we have proposed an efficient TDMA-based MAC protocol for multi-hop WiLD networks to enhance end-to-end network performance. Using the concept of SynOp in multi-radio configuration, allocation of overlapped transmission slots through multi-hop time and colour synchronization process, 2C protocol achieves much better slot reuse. Dependency issue of marker packet based synchronization which exists in 2P-based protocols has been resolved. The proposed protocol implements an interference-aware node colouring algorithm. Based on the colour information available, all nodes generate dynamic TDMA schedules locally which implicitly overlaps transmission of nodes with non-interfering links.

The contributions of this chapter can be summarized as follows:

- 2C protocol outperforms 2P in terms of end-to-end throughput and delay. Throughput of CBR and FTP has been enhanced by 67.54% and 44.5% respectively. Delay performance is also improved radically.
- This scheduling protocol efficiently avoids interference and keeps the communication links busy for almost 100% of the time.
- 2C supports nodes upto 4-hops away from the gateway node with consistent throughput and delay performance.
- Since equal opportunity is provided to both upstream and downstream traffic, some level of QoS requirements of symmetric real-time applications will be inherently met by this protocol.

After working on link level scheduling for providing network level QoS through enhancement of end-to-end performance, we investigate QoS provisioning for real-time applications in resource-constrained multi-hop WiLD network in the subsequent chapters. In the next chapter, we discuss about a dynamic bandwidth utilization scheme for multi-hop WiLD networks which is proposed for supporting QoS in such networks.