A New Backoff Strategy Using Topological Persistence

In Wireless Networks

by

Sai Vishnu Kiran Bhyravajosyula

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved September 2013 by the
Graduate Supervisory Committee:

Violet R. Syrotiuk, Chair
Arunabha Sen
Andrea Richa

ARIZONA STATE UNIVERSITY

December 2013

ABSTRACT

Contention based IEEE 802.11 MAC uses the binary exponential backoff algorithm (BEB) for the contention resolution. The protocol suffers poor performance in the heavily loaded networks and MANETs, high collision rate and packet drops, probabilistic delay guarantees, and unfairness. Many backoff strategies were proposed to improve the performance of IEEE 802.11 but all ignore the network topology and demand. Persistence is defined as the fraction of time a node is allowed to transmit, when this allowance should take into account topology and load, it is topology and load aware persistence (TLA). We develop a relation between contention window size and the TLA-persistence. We implement a new backoff strategy where the TLA-persistence is defined as the lexicographic max-min channel allocation. We use a centralized algorithm to calculate each node's TLA-persistence and then convert it into a contention window size. The new backoff strategy is evaluated in simulation, comparing with that of the IEEE 802.11 using BEB. In most of the static scenarios like exposed terminal, flow in the middle, star topology, and heavy loaded multi-hop networks and in MANETs, through the simulation study, we show that the new backoff strategy achieves higher overall average throughput as compared to that of the IEEE 802.11 using BEB.

DEDICATION

Dedicated to my parents

for their support and faith in me.

ACKNOWLEDGEMENTS

I am very grateful to my advisor Dr. Violet R. Syrotiuk for guiding me through out my research. Her constant support, suggestions, and patience helped me towards the successful completion. I sincerely thank her for encouraging me to take up this research.

I am very thankful to my committee members Dr. Andrea Richa and Dr. Arunabha Sen for their extreme guidance, time and support for my research. Their exceptional co-operation helped me with a smoother dissertation process.

I would also like to thank my lab-mates Abraham Aldaco and Jonathan Lutz for taking out their time and guiding me when I needed the help.

I am thankful to other faculty members in Arizona State University, who made a significant impact on my learning process. I am grateful to Dr. Kevin Howard for giving me the opportunity to be a part of his organization and work with some brilliant people who helped me improving my technical skills.

My special thanks to my room-mates, friends especially Urvish, Raj, Sriram, Anuj, Nitish who gave memorable moments through out my journey at ASU. Further, I would like to thank Sumanth and my family members for their encouragement and love.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Overview

In wireless networks, a single broadcast channel is shared between several geographically distributed nodes. The medium access control (MAC) layer plays an important role in providing the channel access to the nodes. In doing so, two major objectives of MAC are: maximization of the throughput and minimization of the latency. Another goal which might be equally important is fairness. The scheme used by MAC for the channel access influences the overall performance of the network.

Because of the bursty nature of data traffic in wireless networks, we focus our research on the contention based MAC protocols where transmitters compete for access to the shared channel; the MAC protocol decides which node among the transmitters acquires the channel. The dominant MAC protocol in this arena is the IEEE 802.11.

IEEE 802.11 is a contention based MAC protocol using the *distributed co-ordination function* (DCF) for the channel access. DCF uses the idea of the carrier sensing, and random backoff strategies for the channel access. With this scheme, a node tries to access the channel after detecting the channel as idle. If the channel is busy, to avoid the collisions, the node is required to wait for a random period of time before it retries. IEEE 802.11 uses the *Binary Exponential Backoff* (BEB) as its backoff strategy. In the BEB strategy, if a node is involved in a collision, then the *Contention Window* (CW) size of the node is doubled until it reaches the maximum CW_{max}. On successful packet transmission, the CW size is reset to the minimum CW_{min}. The node chooses a random number in the range [0,CW-1] and waits as many mini-slots before trying to access the channel again.

1.2 Weakness of the IEEE 802.11 MAC Protocol

Because of the BEB scheme in IEEE 802.11, the protocol suffers from many weaknesses. Due to the contention window resetting mechanism involved in the BEB, a node which sends the packet successfully, has a great advantage to occupy the channel again. Because of this, in the saturated networks the bandwidth usage is unfair leading to the short-term fairness issues and a decrease in the overall throughput.

Moreover the performance of the 802.11 protocol with BEB backoff in *Mobile Ad hoc Networks* (MANETs) with high load is quite far from the ideal because of the influence of the dynamics in the networks including topology and load as well as interactions with higher layer protocols [2].

The IEEE 802.11 MAC protocol behaves greedily in accessing the channel. As a result, the number of packet collisions increases in scenarios involving exposed terminals and in chain topology scenarios [7], thereby reducing the overall throughput.

1.3 Problem Statement

To overcome the BEB issues many improvements were proposed to the backoff strategies in IEEE 802.11 such as MILD, LMILD, EIED, SBA which involve information exchange and complicated computations that ignore the network topology or demand.

Persistence is defined as the fraction of time that a node is permitted to transmit. We take a different approach by first identifying the ideal persistence for each node in a network with a given topology and traffic loading and then use these persistence values to develop a MAC protocol. Specifically, we intend to use the lexicographic max-min channel allocation to determine the persistence at which a node accesses the channel. That is, the persistence influences how a node backoff, and it ultimately takes into account both topology and load.

The channel allocation problem is treated as a resource allocation problem where transmitters are demands and receivers are the resources. In order to maximize the persistence across the network, the allocation satisfies the following properties:

1. The allocation is feasible.
2. No transmitter is allocated more than the demand.
3. No transmitter can monopolize the channel.
4. Each transmitters allocation is maximized subject to the first three properties.

In [19], lexicographic max-min allocation is shown to satisfy these properties and is used to develop a schedule based MAC protocol. In our research, we focus on using the lexicographic max-min allocation to instead develop a new backoff strategy for the contention based IEEE 802.11 protocol. We evaluate the performance of the new backoff strategy by comparing it with IEEE 802.11 using BEB. We believe the idea of using a metric of persistence that takes into account both topology and load in backoff strategy will improve the network performance in wireless networks.

1.4 Overview of the Solution

We propose a technique to implement the new backoff strategy and hence its contention window size for IEEE 802.11 where each node's persistence is defined by the lexicographic max-min channel allocation as its topological persistence for wireless networks. To implement this strategy, we used the centralized approach proposed in [19] to calculate the topological persistence values of all the nodes in a given network, and then convert the topological persistence for each node into a contention window size. A performance evaluation of the new backoff strategy is carried out first for static wireless networks and then introducing the node mobility and changes in load.

We conclude that the new backoff strategy using topological persistence for the wireless networks delivers better performance in single-hop networks, multi-hop with heavy load networks and in MANETs of high density with heavy loads.

1.5 Document Organization

This chapter introduced our problem statement and gave an overview of our proposed solution. The remainder of the document is structured as follows. Chapter 2 describes the related work for the existing IEEE 802.11 based MAC protocols focusing different backoff strategies, and persistence based protocols. Chapter 3 introduces the algorithms that we use in the implementation of the new backoff strategy for IEEE 802.11. The network simulator (`NS2`) setup, experimental scenarios, and the simulation results are presented in the Chapter 4. And finally, Chapter 5 summarizes and concludes the thesis presenting a few suggestions for future work.

Chapter 2

Related Work

2.1 Overview

Our research focus is on the development of a new backoff strategy for IEEE 802.11 in MANETs. Therefore, we first set the network context by defining a MANET. Then, we introduce the IEEE 802.11 protocol by discussing aspects it uses such as carrier sensing, the 4-way handshake and its binary exponential backoff algorithm. Other research has sought to address the fairness problem by proposing changes to the backoff strategy [4] [12]. We propose to use a new backoff strategy based on the topology and load -aware persistence and define this concept here.

2.2 MANETs

A *Mobile Ad hoc Network* (MANET) is a self-configuring network of mobile nodes which are connected by and communicate through a wireless medium. As the nodes are mobile, network topology may change quickly and unpredictably. All the nodes are provided with wireless transceivers that use omni-directional half duplex antennas. Connectivity between any two nodes in MANET depends on the their position, transmission and reception range, transmission power, in addition to other characteristics of the transceivers and environmental factors.

In the early 1970's, MANETs were called *packet radio* networks and were sponsored by the *Defense Advanced Research Projects Agency* (DARPA) because these networks suit the war zone environment where the communication infrastructure is not available [9]. Even though MANETs were primarily designed for military purposes, they suited the communication system for disasters such as earthquake hit areas [25].

Important properties of a MANET are:

Dynamic Topology

As the nodes in a MANET are independent and mobile, each may voluntarily join or leave the network, or move away from transmission range of a node forming partitions. Also because nodes are typically battery operated, a node may fail to respond because of battery failure. Thus the topology of a MANET is dynamic and unpredictable.

Multi-hop Transmission

In a MANET, transmission between two nodes occurs if the nodes are in the transmission range of each other (single hop). Otherwise, transmission between the nodes should occur through some other intermediate nodes which forward the packets from a transmitter to a receiver (multi-hop). In contrast to the wired or cellular networks, MANETs cannot use the routers for forwarding because they do not use such infrastructure. Hence each node must develop the knowledge of the other nodes which are in its transmission and reception range, so that each node can act as a router to forward the packet to the next hop on the path between its source and destination.

Open Medium

All the mobile nodes of a MANET share the same wireless channel for transmission. Therefore the MAC protocol plays a major role in co-ordinating channel access among the nodes for the transmission. Contention based MAC protocols such as IEEE 802.11 use the idea of carrier sensing and backoff strategies for co-ordinating the channel access, which we discuss in section 2.3.

To improve the channel utilization, a MAC protocol tries to maximize the spatial reuse i.e, maximize the concurrent transmissions. One idea proposed to improve the spatial reuse is to reduce the transmission range by reducing transmission power [17].

Two challenges faced in MANETs to improve spatial reuse in this manner are: the difficulty of finding the optimal transmission range because of the dynamic topology, and the exposed terminal problem which we discuss in section 2.3.

2.3 CSMA

IEEE 802.11 is a contention based MAC protocol using the *distributed co-ordination function* (DCF) for the channel access. DCF is a *carrier sensing multiple access* (CSMA) protocol that requires a node to listen to the channel before transmitting. If the channel is found busy during an interval called the *DCF Interframe Space* (DIFS) interval, the node defers its transmission by a period called the *backoff period*. But this scheme suffers from two classical problems: the *hidden terminal* and the *exposed terminal* problems. The main reason for these problems in CSMA is information asymmetry at the sender and the receiver [4].

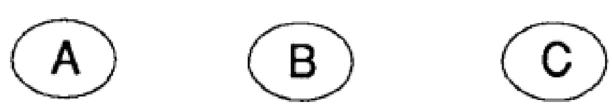

Figure 2.1: Node B is in range of both A and C, but A and C are not in range of each other.

Consider 3 nodes A, B, and C as shown in Figure 2.1. Node A is in carrier sensing range and transmission range of node B but not of node C. Similarly, C is in range of B but not of A. Suppose that when A tries to sense the channel, it detects the channel is free and starts transmitting to node B. Now if C wants to transmit to node B, it senses the channel is free as it does not detect the AB transmission. Therefore, node C also starts transmission thereby resulting in collisions at B. In this case, the node A is hidden from C. This scenario is called the *hidden terminal problem.*

Now, consider the same 3 nodes in Figure 2.1, but suppose that node B is the transmitter and node A is the receiver. If node C wants to transmit to some node other than B, it senses the channel and detects it is not free as node B is transmitting. Hence node C defers its transmission though it may have been able to transmit without interfering with B's transmission to A. In this case, the node C is exposed by the BA transmission but is unable to transmit even though it would not interfere with the reception at A or transmission at B. This scenario is an example of the classical *exposed terminal problem.*

To overcome these problems, improvements including control messages, and new backoff schemes were introduced. DCF employs collision avoidance and the binary exponential backoff strategy to overcome the hidden terminal problem, as we describe next.

CSMA/CA

In *CSMA with collision avoidance* (CSMA/CA) a 4-way handshake between the transmitter and one-hop receiver is introduced. If a transmitter senses the channel as idle for a DIFS interval then it sends an *Ready To Send* (RTS) packet which is answered by a *Clear To Send* (CTS) packet from the receiver. After completing this part of the handshake, the transmitter knows that the receiver is free and sends a data packet. If the receiver receives the data packet correctly, it responds with an *acknowledgement* (ACK) completing the handshake. In this way collisions can be avoided at the receiver. This 4-way RTS/CTS/DATA/ACK handshake is depicted in the Figure 2.2 [8].

A small time interval called a *short interframe space* (SIFS) follows each part of the handshake. A SIFS interval is shorter than DIFS so that a node does not interpret that the channel is free. If there is a collision, that is two or more transmissions overlapped in time then no ACK is sent and all the transmitters involved in the collision backoff to

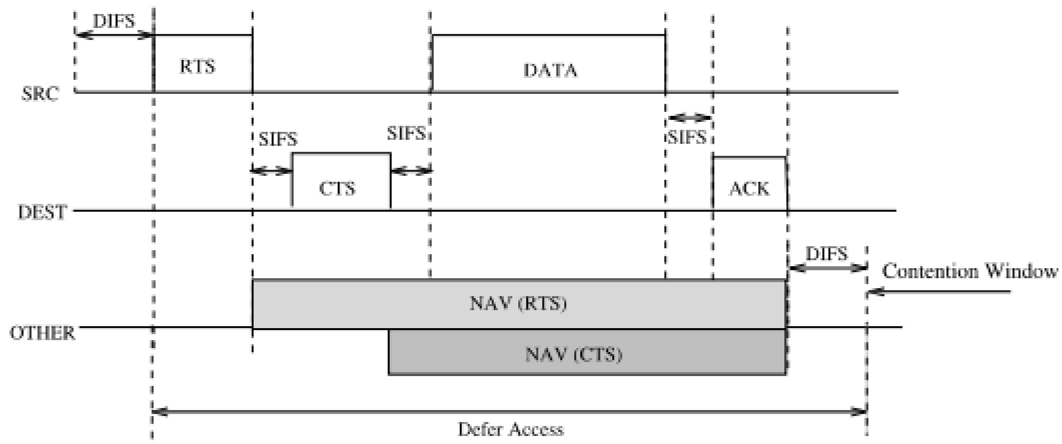

Figure 2.2: CSMA/CA channel access.

avoid further collisions [30]. CSMA/CA employs the *binary exponential backoff* (BEB) for deferring the transmission.

Binary Exponential Backoff (BEB)

The binary exponential backoff mechanism selects a random number of slots uniformly from a node's contention window to wait before attempting to retransmit. On every collision, the contention window size is doubled until it reaches its maximum size CW_{max} (1024). On a successful transmission, the contention window size is reset to its minimum size CW_{min} (32) [15]. That is,

$$CW = \begin{cases} min\{2 * CW, CW_{max}\}, & \text{On collision.} \\ CW_{min}, & \text{On success.} \end{cases}$$

BEB suffers from a fairness problem. It favors a node that has been successful. As a result, these nodes have high throughput because the CW value of a successful transmitter is reset to CW_{min}, giving it maximum chance to transmit again quickly. Hence to overcome this issue, several changes to the backoff strategy were proposed.

MILD

To prevent the fairness problem of BEB, in MACAW a new backoff scheme called the *Multiplicative Increase and Linear Decrease* (MILD) is introduced [4]. In this scheme, upon each collision the CW value is increased by multiplying a factor greater than one (1.5) and upon successful transmission, CW is decreased linearly (by 1). As the CW value is decreased linearly, the transmission chance for the most recently successful node is reduced. Further, MILD also tries to reflect the level of contention by copying backoff values among all contending nodes (overhearing nodes) to ensure fairness [4]. That is,

$$CW = \begin{cases} min\{1.5 * CW, CW_{max}\}, & \text{On collision.} \\ max\{CW - 1, CW_{min}\}, & \text{On success.} \\ CW\ of\ the\ packet\ transmitter, & \text{On overhearing successful packet.} \end{cases}$$

The MILD strategy has been found to suffer with a few issues. Backoff copying among the contending nodes provide fairness only in homogeneous networks [4].

It also suffers low performance due to the few occasional collisions that occur in networks with lighter loads [23] because with these few collisions, the CW value quickly reaches to CW_{max}. Multiplicative increase of the CW value leads to high waiting periods which lowers the network performance [12].

For example, consider a topology shown in Figure 2.3 where the nodes are in a star topology. Let node E, which can be viewed as the access point, try to transmit to one of the receivers by sending RTS. If that receiver fails or moves out of the transmission range of E, it will wait for some time before retransmitting. After all the maximum retrial attempts, if the CTS is not received, the node will backoff increasing the CW value by multiplying with 1.5 thinking there is no response because a collision had occurred. Later if any successful transmission occurs, all the other nodes that are in the transmission range of the access point, will overhear and copy the CW value of the access point. Further backoff makes the CW value of the access-point reach CW_{max} quickly and at this point, if any transmission succeeds, then all the nodes in the network copy the CW_{max} of the access point decreasing the overall network performance.

Specifically, if CW_{min}=16, CW_{max}=1024 and a node's CW reaches CW_{max}, using MILD it takes 1008 successful transmissions to reach CW_{min} [23].

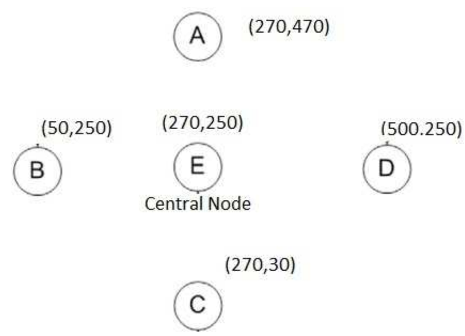

Figure 2.3: Star Topology.

Therefore to improve the network performance in wireless networks, and to overcome the copy problem of MILD; backoff strategies like *Linear/Multiplicative Increase and Linear Decrease* (LMILD), *Sensing Backoff Algorithm* (SBA), and *Exponential Increase and Exponential Decrease* (EIED) are proposed.

LMILD

To address the problem of copying in MILD and also to improve the fairness, LMILD is proposed [12]. LMILD is similar to MILD but uses some extra information about the packet collisions on the channel. That is, LMILD uses the idea that a node can overhear successful/collided packet transmissions.

In LMILD, each node increases the *CW* by multiplying with a factor m_t on collisions, and on a successful transmission decreases the *CW* linearly by l_s units. Along with this, any node overhearing a collision increases the CW by l_c units. That is,

$$CW = \begin{cases} min\{\ m_t * CW, CW_{max}\}, & \text{On collision.} \\ min\{\ CW{+}l_c, CW_{max}\}, & \text{On overhearing collisions.} \\ max\{CW - \ l_s, CW_{min}\}, & \text{On successful transmission/overhearing success.} \end{cases}$$

The LMILD scheme overcomes the CW copy mechanism that is used in the MILD scheme. In the LMILD scheme, every node knows the status of the channel. This scheme is based on the assumption that all neighboring nodes are able to detect the collided packets [12] which is actually not true in the real world scenarios.

EIED

To improve the network performance in wireless networks and to overcome the contention window copy problem as well as the linear decrease problem in MILD, the *Exponential Increase Exponential Decrease* (EIED) scheme is proposed [23]. In EIED, the values of the backoff factors r_I and r_D by which the contention window size is changed are considered as the exponents of 2, because of which the algorithm is termed as Exponential Increase and Exponential Decrease. In the EIED scheme, when a packet transmitted is involved in a collision then the *CW* of the node increases by a *backoff factor* r_I, and on a successful transmission *CW* is decreased by another *backoff factor* r_D. That is,

$$CW = \begin{cases} min\{r_I{*}CW, CW_{max}\}, & \text{On collision.} \\ max\{CW/r_D, CW_{min}\}, & \text{On success.} \end{cases}$$

In MILD, linear decrease of the CW size over a successful packet transmission is very slow and was leading to the lower network performance in the scenarios like star topology.

Hence to overcome this slow decrease of the CW size, in EIED the CW is exponentially decreased. To achieve a better performance than MILD using EIED, in [23] the backoff factors r_I and r_D are initialized as 2 and $\sqrt{2}$, respectively. Using these values in $EIED$, if CW_{min}=16, CW_{max}=1024 and if a node's CW reaches CW_{max}, it takes 12 successful transmissions to reach CW_{min} [23].

SBA

To address the fairness problems of BEB and low network performance due to the copying problem of MILD, the *Sensing Backoff Algorithm* (SBA) is proposed in [13]. In this scheme, backoff interval is modified according to the results of the sensed channel activities. In *SBA* backoff scheme, when a packet transmitted is involved in a collision then the node multiplies the CW by a value called α where ($\alpha > 1$) and upon successful transmission both sender and receiver change their CW values by multiplying with a value θ where ($\theta < 1$). All the other nodes, which overhear (sense) the successful transmission are required to decrease their CW by β steps where each step is defined as the transmission time of the packet given as γ. That is

$$CW = \begin{cases} min\{\alpha * CW, CW_{max}\}, & \text{On collision at transmitter.} \\ max\{\theta * CW, CW_{min}\}, & \text{On success at both transmitter and receiver.} \\ max\{CW\text{-}(\beta * \gamma), CW_{min}\}, & \text{On overhearing successful packets.} \end{cases}$$

Both SBA and EIED decreases the CW exponentially. But in EIED CW value changes with either the collision or the transmission of packet ignoring the case of overhearing. In SBA even the overhearing nodes decrease the CW exponentially according to transmission time of the packet. SBA improves the fairness by adding the feature of sensing or overhearing, at the same time not involving in the migration of the CW values to the overhearing nodes [13].

2.4 Persistence

In [19], *persistence* is defined as the percentage of time a node is permitted to transmit. Persistences influence network performance. The MAC protocol used will directly influence the persistences, regardless of whether it is contention-based, schedule-based, randomized, or deterministic. The *occupancy* of the channel is defined as the fraction of time that a node spends in transmitting over the channel, i.e, *occupancy* is the amount of time a node actually transmits. From [19], when every node is saturated and employs each transmission opportunity, occupancy is the node's persistence. The instantaneous occupancy for a single packet transmission is given by

$$\frac{\mathit{Transmit\ Time}}{\mathit{Idle\ Time} + \mathit{MAC\ Latency} + \mathit{Transmit\ Time}}$$

where

- *Transmit Time* is the time required to transmit the packet,
- *MAC Latency* is the time a packet spends in the MAC layer queue for transmission, and
- *Idle Time* is the time spent in waiting for a packet to transmit.

Persistence-based Protocols

The protocols which try to explicitly determine the persistence and use it for channel access are the *persistence-based protocols*. A class of MAC protocols operate with fixed persistence, dividing time into slots and transmitting in a fixed percentage of them. The *p-Persistent CSMA* protocol works as follows [16]: A node senses the channel at the beginning of each slot. If the channel is detected as idle, the node transmits the packet in that time slot independently with probability p. Note that each node has the same probability p.

Few protocols negotiate both persistences and schedules realizing them, usually with the objective to maximize spatial re-use, or minimize average or maximum delay. FPRP [29], CATA [26] and SEEDEX [22] are among the very few protocols that determine transmitter's persistence explicitly based on either the network topology, i.e; neighbor information or the demands. In all three protocols, explicit reservations for transmission slots are made, and the reservation schedules are sent to all the nodes in the network to avoid collisions. The reservations are made by exchanging the information with the two-hop neighbourhood.

FPRP use a five phase information exchange for reserving the time slots and exchanging the schedule information in a distributed fashion. CATA also does the reservation of the collision free time slots for transmission. CATA uses a different slot and frame structure where each slot is divided into 4 control mini-slots and 1 data mini-slot. Using these mini-slots a node performs the reservation. To consider the neighbors during the reservation, CATA choose the frame length to be larger than number of nodes in two hop distance. SEEDEX also tries to make the slot reservations by using a random schedule generated by a pseudo-random number generator and exchanging the seeds among neighbors to identify the neighbor's schedule. Detailed information about the FPRP phases, about the frame structure and the reservation process in CATA, and the SEEDEX process are described next.

FPRP

The *Five-Phase Reservation Protocol* (FPRP) [29] is a single channel, time division multiple access (TDMA)-based broadcast scheduling protocol, presented for mobile ad hoc networks. The protocol makes the reservations for channel access as well as the node broadcast scheduling. FPRP's reservations are fast and efficient with a very low probability of conflict. FPRP permits multiple reservations to be made at various parts of the network simultaneously where reservation involves nodes within two hop distance. All the reservation process is done locally at a node, so FPRP is insensitive to the network size. The five phases in FPRP are:

1. *Reservation Request Phase.* In this phase, a node willing to make a reservation sends a *Reservation Request (RR)* packet with probability p.

2. *Collision Report Phase.* If a node receives multiple RR packets then it sends a *Collision Report (CR).*

3. *Reservation Confirmation Phase.* If a node receives no CR from the *Collision Report* phase, it sends a *Reservation Confirmation (RC)* packet to all the nodes which are one hop away to inform them about the reservation.

4. *Reservation Acknowledgment Phase.* All the nodes which received an *RC* in the *Reservation Confirmation* phase, acknowledge by sending a *Reservation Acknowledgment (RA)* packet.

5. *Packing/Elimination Phase.* Every node that is two hops from a transmitter which has made its reservation since the last *Packing/Elimination* phase, sends a *Packing Packet (PP)* to inform there is a recent success three hops away. The transmitter, after sending a data packet, sends an *Elimination Packet (EP)* with a probability of 0.5, giving a chance for the next transmitter (neighbor) which is waiting.

CATA

A protocol based on topology-dependent transmission scheduling, named *Collision-Avoidance Time Allocation* (CATA), is introduced for MANETs in [26]. CATA allows nodes to contend for and reserve time slots by means of a distributed reservation and handshake mechanism. Each slot reserved is divided into 5 mini-slots of which first 4 are the *control mini-slots* (CMS1-CMS4) and the last mini-slot, called a *data-mini-slot* (DMS), is meant for data as shown in Figure 2.4 from [26].

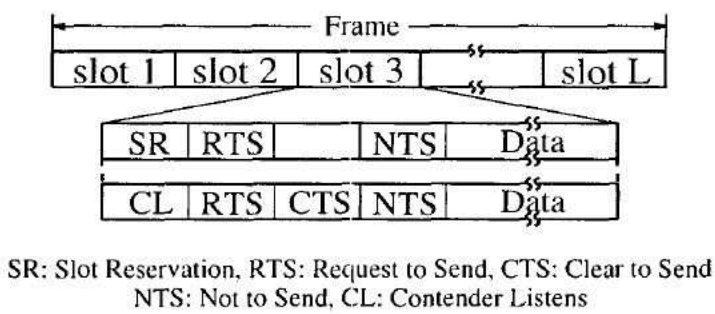

Figure 2.4: Division of CATA frame [26].

In the Figure 2.4, reservation of the slots is shown. The sender of an intended reservation sends a *slot reservation packet* (SR) in *CMS1* only if it is not engaged in data exchange during the *DMS* of current slot. The source listens over the channel to ensure there is no busy tone (slot not reserved) and the slot is free; it sends an RTS during *CMS2*. If an RTS for unicast is received correctly at the intended receiver, the receiver responds with a CTS during *CMS*4; otherwise, no CTS is sent in *CMS*4. If a CTS is received in the *CMS4*, sender node waits to ensure there is no failure in the reservation request due to the noise detection during *CMS*4. If noise is detected, *not-to-send packet* (NTS) is send by the sender during *CMS*4 indicating reservation failure. After the *CMS*4 time, sender starts transmitting during *DMS* time. If a unicast sender detects a successful reservation with the reception of the CTS. Data can flow during the *DMS* of the current slot, and the same slot in subsequent frames, until the unicast flow is terminated

In the case of multicast or broadcast, if a node receives a correct RTS for during *CMS*2, then receiver remains quiet during *CMS*3 and *CMS*4. Otherwise, receiver sends an NTS during *CMS*4 to any potential broadcast being made, indicating reservation of slot cannot be done. If the broadcast sender does not receive an NTS during *CMS*4, it concludes that the reservation is successful. Otherwise, the reservation is not successful.

To consider the neighbor information, the frame length used in CATA is larger than number of nodes in the two-hop neighbourhood, which in worst case will be $\text{Min}\{d^2+1, N\}$ where d is the maximum node degree of the network and N is the number of nodes in the network [26].

SEEDEX

To overcome the problem of poor scaling in the performance of ad hoc networks, the *Seed Exchange* (SEEDEX) protocol is proposed [22]. The main idea of SEEDEX is to use a random schedule which is generated using a pseudo random number generator with different seeds. By exchanging only seeds within its two-hop neighborhood (not entire schedules), the nodes provide the information about their schedules to their neighbors, giving a chance for each node to choose the transmission slots according to the reserved schedules. Moreover SEEDEX does not employ any backoff counters [22]

If a node T has a packet to transmit to R, it first waits for a slot at which T is in the *possibly Transmit* state and node R is in the *listen* state. The node T transmits with a probability p= Min $\{\frac{\alpha}{n+1}, 1\}$, where n is the number of neighbors of node and α is a constant whose value is chosen according to the traffic; and R refrains from transmission with probability *1-p*, where p is the transmission probability of the node T. In the light traffic scenarios, value of α is set to 2.5, and to 1.5 in heavy traffic scenarios [22].

Further extensions to SEEDEX are done to obtain *SEEDEX With Reservations* (SEEDEX-R) [22] which incorporates the usage of RTS and CTS. In SEEDEX-R, a transmitter uses SEEDEX to do the reservations for transmitting RTS instead of data which is replied to with a CTS by the receiver. After the handshake, the transmitter will now send a data packet which is followed by an ACK packet from the receiver.

Thus, in SEEDEX by exchanging only the seeds, collision free reservations are made.

2.5 Topological Persistence

From [19], the problem of channel allocation is treated as a resource allocation problem, in which transmitters are modeled as demands and receivers are modeled as resources. The *lexicographical max min* allocation is defined as an allocation where any value x in the allocation cannot be increased without decreasing some other value x' which is smaller than or equal to x [28].

Let $x = (x_1, x_2, \ldots, x_n)$ satisfy $x_1 \leq x_2 \leq \ldots \leq x_n$, and $y = (y_1, y_2, \ldots, y_n)$, satisfy $y_1 \leq y_2 \leq \ldots \leq y_n$. Then x is lexicographically greater than y if there exists an index k, $1 \leq k \leq n$, such that $x_i = y_i$ for all $1 \leq i < k$ and $x_k > y_k$. For example, the vector (3,3,3,3) is lexicographically greater than both (1,10,1000,1001) and (2,3,4,5) but not (3,3,3,4) or (4,5,6,7) [18].

The problem of channel allocation is viewed as the resource allocation problem. Consider R to be a set of N resources, with capacity $C = (c_1, c_2, \ldots, c_n)$ and D the set of M demands, with magnitudes $W = (w_1, w_2, \ldots, w_m)$. A resource allocation s = $(s_1, \ldots, s_M)$ is lexicographically max-min if the vector is feasible and is lexicographically greatest among all feasible allocations when each is sorted in the increasing order [21]. Resource $j \in R$ is required by demands $D_j \subseteq D$. Each demand i utilizes the capacity of all resources in R_i equally and simultaneously. An allocation is feasible if total allocation of a resource j is less than or equal to the total capacity of resource i.e, $\sum_{i \in D_j} s_i \leq c_j$ and each individual allocation of resource j should be less than or equal to the desired demand i.e, $s_i \leq w_i$ for all $i \in D$.

Each transmitter i has a demand w_i which is its desired persistence. Together all the demands may not be able to be satisfied. So the lexicographic max-min allocation to i, s_i which we call its topological persistence, computes the max persistence node i can have while satisfying all the properties in [19]:

1. The allocation is feasible.
2. No transmitter is allocated more than the demand for it.
3. No transmitter can monopolize the channel (allocation should be fair).
4. Each transmitters allocation is maximized subject to the first three properties.

For example, consider the topology with 10 nodes as shown in Figure 2.5 in [19]. For every node i, both desired persistence of i, w_i and the topological persistence value s_i are provided as labels. The allocation s of topological persistence values is feasible because the persistence allocated is less than or equal to the capacity. The double circled nodes 2, 4, 5, 7, and 8 indicate the saturated resources. We observe that the demands of nodes 1, 2, 4, 7, and 8 are satisfied and those of nodes 3, 5, 6, 9, and 10 are not satisfied. For example, node

3 wanted to transmit 100% of time (w_3=1.00) but because it has 5 neighbors, each with its own demands, it is impossible to allocate node 3 its demand while satisfying the four properties. Similarly, we obtain node 2 topological persistence to be equal to the desired persistence (s_2=w_2) which indicates that we satisfied the demand of the node 2. Node 2 is allocated its demand because majority of its neighbors have a very low desired persistence and moreover the node 2 resource is saturated with allocation s_2 while satisfying the four properties of the allocation. Other node 5 has the allocated persistence to be less than the desired persistence ($s_5 < w_5$) because the capacity for the node 5 is saturated with the s_5 as it satisfies all the four properties with this allocation.

From the Figure 2.5, the topological persistence of nodes 3, 6, and 9 cannot be increased because they are constrained by the saturated nodes 8, 4 respectively. Node 8 is connected to 8 neighbours, and the total persistence of its neighbours and itself would come up to 1.0 $(0.16+0.13+0.16+0.16+0.01+0.1+0.1+0.16+0.02)$. So if we try to increase the persistence of Node 3 then according to lexicographic max-min allocation, we might end up reducing the persistences of nodes 8 which is unfair and the allocation would be infeasible. Similarly node 4 is connected to the nodes 3, 6, 9. The total persistence of its neighbours and itself would come up to 1.0 $(0.16+0.10+0.10+0.16+0.16+0.13+0.16+0.02+0.01)$. Increase in the persistence value of node 3 will result in an infeasible allocation. A Complete step-by-step explanation of the topological persistence computation for this topology is discussed in section III and in Table 3.1.

There are two approaches proposed in [19] for calculating the topological persistence, a *centralized algorithm* and a *decentralized or distributed algorithm.*

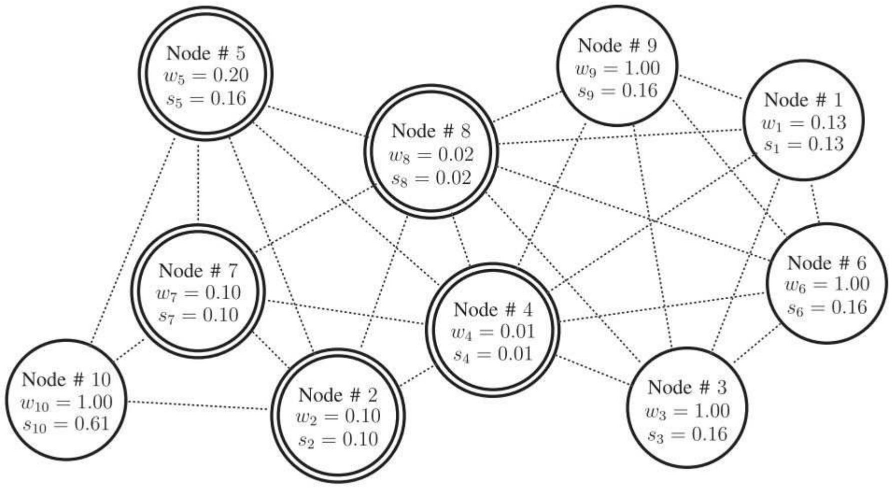

Figure 2.5: Example network with topological allocation.

Centralized Approach

In the centralized algorithm, a single node calculates the topological persistence values of all the nodes in the network after collecting the topology information and later, also the demands of each node. The centralized algorithm to compute topological persistence, takes the adjacency matrix of the given network topology, and the initial demand values as input and returns topological persistence values for each node as output. Later, the algorithm is extended to also consider demand values that consider traffic load by considering the arrival rate, queue size and the slot length. The algorithm recursively provides a small non-zero increment to the allocation, ε, to each node and the demand left over is updated accordingly. The algorithm terminates when either all the demands are satisfied or when nodes have been bottlenecked by saturated resources. The algorithm assumes the network topology is stable i.e, does not change while it executes. Finally each node obtains its allocation, which it interprets as its persistence, calculated considering both topology and demand. A more detailed description with pseudocode of the algorithm is given in Chapter III.

Distributed Approach

In the distributed algorithm, each individual node calculates its persistence locally. A distributed auction is used, where each receiver corresponds to an auctioneer and each transmitter corresponds to a bidder. Auctioneer i holds an auction for channel allocation at node i. All the neighbors of the node i respond to the auction by participating in bidding. The auctioneers start with an initial bid to which neighboring bidders respond with their claims. If a bidder's claim reaches its desired persistence, it terminates. An auctioneer after receiving the bids and claims, responds by either closing its auction or increasing its offer. The process of increasing the claims and bids repeats until either all the bidders have claimed their desired persistence or all have been limited at an auction whose capacity has been completely allocated [19]. The synchronous distributed algorithm in [19] is extended to a more general asynchronous distributed algorithm in [20].

2.6 Summary

In this chapter we provided the background for a few contention based MAC protocols for MANETs, specifically about the CSMA and CSMA/CA protocols, and different backoff strategies including BEB, MILD, LMILD, EIED and SBA. Then we discussed the concept of persistence, and a few protocols that explicitly try to calculate the persistence such as CATA, SEEDEX, and FPRP. Finally we introduced the *topological persistence* as the *lexicographic max-min allocation*, along with two different strategies proposed for its calculation. We implement a backoff strategy for IEEE 802.11 that uses the topological persistence to estimate the CW size based on the centralized algorithm. In the next chapter, we provide the details of how we convert the persistence into a CW size, and how the channel is being accessed according to the persistence that takes into account both the topology and the traffic load.

Chapter 3

Backoff using Topological Persistence

3.1 Overview

In this chapter we provide the details of the algorithms that we use in the implementation of the new backoff strategy based on topological persistence. We start the discussion by providing details of the algorithm which explains the functionality of the centralized node in the centralized approach which we use. Later we provide the details of the centralized algorithm that we use for the calculation of topological persistence. We introduce the proposed algorithm that converts the topological persistence into contention window size, the effect of which is that the backoff strategy takes into account both topology and load. We finally provide the details of how a node accesses the channel using the new backoff that considers both topology and load.

3.2 The New Backoff Strategy

We use a centralized approach in calculating the topological persistence and CW values. In Algorithm 1 we provide the details of how a single/central node calculates persistence of all nodes in the network by collecting the information about both the network topology, and traffic load of each node as its demand.

In Algorithm 1, the central node starts by calculating the desired persistence of all nodes and store the results in the vector desired persistence W. The desired persistence or the demand values are initially set to 1. This ignores the traffic load of each node in persistence computation and is interpreted as a demand of 100% of the channel resulting in an allocation that is as much as each node can get considering topology only. Later to incorporate the traffic load in persistence computation, we estimate the demand value using Algorithm 2, described in section 3.3. Then the central node calculates the topological persistence of all the nodes in the network by calling Algorithm 3, discussed in section 3.4, and stores them in the vector S. From the values obtained, the contention window size (cw) of each node is calculated by calling Algorithm 4 described in section 3.5 and stored in the vector CW.

Inputs to the Algorithm 1 are the set of nodes, the capacity vector containing the capacity values of each node, the arrival rate vector containing the arrival rate at each node, the queue size vector which contain the number of packets en-queued in the queue of each node, the packet size which is a constant set to 256 bytes, the slotlength which is the time taken for a mini-slot, set to 20 microsec, and maximum contention window size CW_{max} set to 1024 , and minimum contention window size CW_{min} is set to 32; the output of the algorithm is the contention window size values of all the nodes which are later distributed to the respective nodes. Initially the capacity vector C=(1,1,...,1).

The Algorithm 1 assumes that the network is stable during the calculation of topological persistence and CW values, i.e., no inputs change while the algorithm executes. In the case of static network scenarios, Algorithm 1 is called only once at the beginning of the simulation whereas in the case of mobile network scenarios, the algorithm is called periodically with a regular interval of time (10 msec). We choose the interval time as 10 msec to satisfy two conditions: the centralized algorithm assumes the network as stable during execution, and, moreover, the computation of the centralized algorithm to obtain the topological persistence and CW size values takes approximately 10 msec. Thus any changes in load are also incorporated with the same period.

Algorithm 1 Centralized_Calculation

function CENTRALIZED_CALCULATION($V, A, Q, Pksize, slotlength, C, T, CW_{min}, CW_{max}$)

Inputs:

- V = Set of nodes.
- A = Arrival rate vector A=(a_1,a_2,...,a_n)
- Q = Queue size vector Q=(q_1,q_2,...,q_n)
- *PkSize* = Packet size (256 bytes).
- slotlength = 20 microseconds
- C = Capacity vector C=(c_1,c_2,...,c_n)
- $T = n \times n$ Adjacency Matrix.
- CW_{min} = Minimum CW value (32).
- CW_{max} = Maximum CW value (1024).

Result:

- Contention window size vector CW = (cw_1,cw_2,...,cw_n)

$W \leftarrow 1$ // Only topology is considered.
if (isDemandToApply == true) **then**
// To estimate the demand based on the traffic load to incorporate demand in the persistence computation.
$W \leftarrow$ get_Desired_Persistence(V, PkSize, A, Q, slotlength)
end if
$V_{active} \leftarrow V$ ▷ Initialize active nodes V_{active} to V
// get the topological persistence of all the nodes in network.
$S \leftarrow$ Compute_TopologicalP(V,C,V_{active},W,T)
// get the contention window size of all the nodes in network.
$CW \leftarrow$ Calculate_CW(V, CW_{min}, CW_{max}, S)
return **CW**
end function

3.3 Estimate the Desired Persistence W

The desired persistence values of all nodes that are used in Algorithm 3, can be taken as 1, which ignores the traffic load in the persistence computation and considers topology alone. To incorporate the traffic load in the persistence calculation, the traffic load at each node is estimated using Algorithm 2, and is called from Algorithm 1.

Inputs to the Algorithm 2 are the set of nodes V, the packet-size *PkSize*, the arrival rate vector A, the queue size vector Q, and the slot length; the output is the desired persistence vector W.

The estimated demand for each node has two parts: the first comes from the recent en-queue rate (demand1); the second comes from the number of packets that are staged in the queue (demand2). By summing both parts, and multiplying the result by the slot length gives an estimated demand in terms of packets/slot. We later normalize the result so that the total traffic load will be equal to 1. To overcome the zero values for the desired persistence in the case of nodes being receivers (not sources) or idle nodes, we assign a minimum desired persistence of 0.01 to all the nodes. Finally the normalized desired persistence vector is returned.

3.4 Calculate Topological Persistence

We now discuss the details of the calculation of the topological persistence values using the centralized approach proposed in [19]. In Algorithm 3, a single node calculates the topological persistence. Algorithm 3 is called from the Algorithm 1.

Inputs to the Algorithm 3 are the set of nodes V, the adjacency matrix T to know the neighbors of each node, the demand vector W which indicates the desired persistence of each node, the set of active nodes V_{active}, and a capacity vector C. The output is the vector of topological persistences S which contains the persistence value for each node [19].

Algorithm 2 getDesiredPersistence

function GET_DESIRED_PERSISTENCE($V, PkSize, A, Q, slotlength$)

▷ //Get all the initial desired persistence demands.

Inputs:

- V = Set of nodes.
- $PkSize$ = packet size (256 bytes).
- A = arrival rate vector A=(a_1,a_2,...,a_n) in bps
- Q = Queue size vector Q=(q_1,q_2,...,q_n)
- slotlength = 20 microsec

Result:

- Desired_persistence values W = (w_1,...,w_n)

$Sum \leftarrow 0$
// Finding load as pkts/sec
for each node i in V **do**
  // demand1 is the en-queue rate in terms of pts/sec
  $demand1 \leftarrow \frac{a_i}{PkSize}$
  $demand2 \leftarrow q_i$ ▷ // number of packets in the queue.
  $w_i \leftarrow (demand1 + demand2) \times slotlength$
  if $w_i == 0$ **then**
    $w_i = 0.01$
  end if
end for
// Normalize the demands such that total traffic load = 1.
for each node i in V **do**
  $Sum \leftarrow Sum + w_i$
end for
for each node i in V **do**
  $w_i \leftarrow \frac{w_i}{Sum}$
end for
return **W**
end function

Algorithm 3 ComputeTopologicalP

function COMPUTE_TOPOLOGICALP(V, C, V_{active}, W, T)

▷ //Compute the topological persistence of all nodes.

Inputs:

- V = Set of nodes.
- C = Capacity Vector C=(c_1,c_2,...,c_n).
- Active nodes $V_{active} \subseteq V$
- W = Desired Persistence W=(w_1,w_2,...,w_n)
- $T = n \times n$ Adjacency Matrix.

Result:

- Topological_persistence values S = (s_1,...,s_n)

S=(0,0,...,0)

//Largest non-zero ε to be allocated to all demands in V_{active}

$\varepsilon \leftarrow 1$

for all $j \in V$ **do**

  if $T_{i,j} == 1$ *for some* $i \in V_{active}$ **then**

    Add node i to set J

  end if

  $\varepsilon \leftarrow min\{\varepsilon, \frac{c_i}{|J|}\}$

end for

for all i in V_{active} **do**

  $\varepsilon \leftarrow min\{\varepsilon, w_i\}$

end for

// Increase the persistence of all the nodes by assigning ε

// Remove all the satisfied demands from V_{active}

for all i in V_{active} **do**

  $s_i \leftarrow \varepsilon$

  $w_i \leftarrow w_i - \varepsilon$

  if $w_i == 0$ **then**

    $V_{active} \leftarrow V_{active}/i$

  end if

  for all j in V **do**

    if $T_{i,j}$==1 **then**

      $c_j \leftarrow c_j - \varepsilon$

    end if

  end for

Algorithm 3 ComputeTopologicalP (continued ...)

```
        for all j in V , i in V_active do
            if c_j = 0 and T_{i,j}=1 then
                V_active ← V_active/i
            end if
        end for
    end for
    if V_active ≠ ϕ then
        S ← S + ComputeTopologicalP(V, C, V_active, W, T)
    end if
    return S
end function
```

Algorithm 3 starts by setting an initial smallest non-zero increment to allocation, ε, which is to be granted to each node. To find out the smallest possible non zero allocation and also to reduce the number of recursive steps, we initialize the ε to 1. The variable V_{active} indicate all the active nodes whose persistence needs to be calculated. Initially, V_{active}=V and the capacity vector C=(1,1,...,1). For each node j in the set V, algorithm finds all the neighbors J in the set V_{active}. Algorithm now tries to find the minimum allocation value by comparing the ε with the ratio of capacity of the node and size of set J (min1). The desired persistence of each i in V_{active} is examined to determine the smallest demand (min2).

The small non-zero allocation, ε to be assigned is the smallest of min1 and min2. At each recursive step the ε obtained is granted to each node in V_{active} and the capacity vector is updated to reflect this allocation. Nodes that have either reached their desired persistence or that have been blocked by saturated resources are removed from V_{active}. Finally the procedure terminates when V_{active} is empty. Proof to show that the algorithm terminates within a finite number of steps is discussed in [19]. At each recursive step at-least one node's demand is satisfied.

For the topology with 10 nodes shown in the Figure 2.5, the initial inputs to the Algorithm 3 are the unsatisfied set V_{active}=$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, the capacity vector

C= {1, 1, 1, 1, 1, 1, 1, 1, 1, 1} and initial demand vector W= {0.13, 0.1, 1, 0.01, 0.20, 1, 0.10, 0.02, 1, 1 }.

In the first iteration, we get the ε value as 0.01, the smallest initial demand (from node 4) in the entire network and assign this to all the unsatisfied nodes reducing the demand by ε. This satisfies the demand of node 4 and hence it is removed from the unsatisfied set V_{active}. In the second iteration, the smallest non-zero allocation ε returned by the algorithm is 0.01 (from the remaining demand of node 8). This allocation is given to all the unsatisfied nodes and the satisfied node 8 is removed from the unsatisfied set, V_{active}={1, 2, 3, 5, 6, 7, 9, 10} and the demand vector is adjusted as W={0.11, 0.08, 0.98, 0, 0.18, 0.98, 0.08, 0, 0.98, 0.98}. As the set V_{active} is not empty, algorithm continues and the ε value we obtain is 0.08 (remaining demand of node 2 and node 7). We provide this allocation to all the unsatisfied nodes in the network which satisfies node 2 and node 7, and removing both these nodes from the unsatisfied set which will be V_{active}={1, 3, 5, 6, 9, 10 } and update the demand vector as W= { 0.03, 0, 0.9, 0, 0.1, 0.9, 0, 0, 0.9, 0.9}. Algorithm continues with iteration 4 and the smallest non-zero allocation ε = 0.03 (remaining demand of node 1) is given to all the nodes in the set V_{active}, removing the satisfied node 1 from the V_{active}={3, 5, 6, 9, 10 } and updating the demand vector as W={0, 0, 0.87, 0, 0.07, 0.87, 0, 0, 0.87, 0.87}. As V_{active} is not empty, the algorithm continues with iteration 5.

We observe the minimum demand value in the W vector as 0.07 but from the remaining capacity at node 8 and number of its unsatisfied neighbors, the smallest allocation ε we get is 0.12/4=0.03 where 0.12 is the remaining capacity at node 8. This ε value is allocated to all the unsatisfied nodes of the network which completes the capacity of node 8 and node 4 because of which the neighboring unsatisfied nodes are removed from the set V_{active}. Thus the new V_{active}={10}. As the set V_{active} is not empty, algorithm continues and similar to the above step, the smallest allocation ε is obtained as 0.45/1=0.45, where the remaining capacity of node 7=0.45 and the active nodes present=1. This ε is allocated to the nodes present in the V_{active}. At this point the capacity of the node 7, node 2 and node 5 are completed and hence the unsatisfied neighbors of these nodes should be removed from the unsatisfied set, making V_{active} empty and hence the algorithm terminates at this point, and the final topological allocation resulted by Algorithm 3 is S={0.13, 0.1, 0.16, 0.01, 0.16, 0.16, 0.1, 0.02, 0.16, 0.61}. In the Table 3.1 the summary of the iterations of the algorithm is shown.

#	ε	**Capacity Vector**	V_{active}
1	0.01	{0.94,0.94,0.95,0.91,0.94,0.95,0.94,0.91,0.96,0.96}	{1,2,3,5,6,7,8,9,10}
2	0.01	{0.89,0.89,0.91,0.83,0.89,0.91,0.89,0.83,0.93,0.92}	{1,2,3,5,6,7,9,10}
3	0.08	{0.57,0.57,0.67,0.27,0.57,0.67,0.57,0.27,0.77,0.6}	{1,3,5,6,9,10}
4	0.03	{0.45,0.51,0.58,0.12,0.51,0.58,0.51,0.12,0.71,0.54}	{3,5,6,7,9,10}
5	0.03	{0.36,0.45,0.52,0,0.45,0.52,0.45,0,0.68,0.48}	{10}
6	0.45	{0.36,0,0.52,0,0,0.52,0,0,0.68,0.03}	ϕ

Table 3.1: Step-by-step execution process of topology 2.5 using Algorithm 3

3.5 CW Calculation

In this section, we provide details of the conversion function i.e., Algorithm 4 which is called from Algorithm 1. Using Algorithm 4, we convert the persistence value of each node into a corresponding CW value. Inputs to this function are the set of nodes V, the minimum and maximum sizes of the contention window CW_{min} (32), and CW_{max} (1024), respectively, and the vector S containing the topological persistence as output from Algorithm 3. The output is CW, a vector that contains the cw_i value of each node based on its topological persistence value s_i. As persistence is defined as the amount of time a node is permitted to transmit. Therefore a node with persistence P should transmit more often than a node with persistence $P/2$. This leads to inverse relation between persistence and waiting time (contention window size).

A node may attempt transmission if it has a packet to transmit when its backoff timer has expired. Hence in [27], the persistence probability p_i of a node i is expressed in terms of cw_i as :

$$p_i = \frac{2}{1 + cw_i} \tag{3.1}$$

Equation 3.1 is the persistence probability function used in IEEE 802.11 DCF, which is obtained by modeling the stochastic backoff time counter process and channel allocation process as Markov chain. The entire proof is presented in [5] [6]. Using this equation 3.1, the CW values based on a node's topological persistence is calculated using

$$cw_i = min\{\frac{2 * CW_{min}}{s_i} - 1, CW_{max}\} \tag{3.2}$$

To restrict the CW values between the range CW_{min} and CW_{max}, we multiply the constant CW_{min} by 2 in the numerator in the equation 3.2. We restrict the CW value such that it never exceeds CW_{max}.

3.6 Channel Access with New Backoff

Once the calculated CW values are distributed to all the nodes, a node follows Algorithm 5 for transmitting a packet. This algorithm is called whenever a node has a packet to transmit.

In Algorithm 5, we present the details of channel access by a node i using its new cw_i value that is calculated using its topological persistence s_i. The channel access strategy is similar to that of IEEE 802.11 channel access.

When a node i has a packet to transmit, it first senses the channel for DIFS time. If the channel is found idle, then the node waits for extra time that is a random number of slots between 0 and cw_i time interval and rechecks for the channel status. This extra wait time takes into account how persistent the node should be considering its topology

Algorithm 4 CalculateCW

function CALCULATE_CW(V, CW_{min}, CW_{max}, S) ▷ //Converting topological persistence into contention window.

Inputs:

- V = Set of nodes.
- CW_{min} = Minimum Contention Window size (32).
- CW_{max} = Maximum Contention Window size (1024).
- S = A vector of topological persistences S=(s_1,s_2,...,s_n).

Result:

- CW vector where $CW = (cw_1, cw_2, \ldots, cw_n)$

for every node i from 1 to $|V|$ **do**
// Node's persistence is taken and converted to contention window size using the equation 3.2.
$persistence \leftarrow s_i$
$cw_i \leftarrow \frac{2*CW_{min}}{persistence} - 1$

if $cw_i > CW_{max}$ **then**
$cw_i \Leftarrow CW_{max}$
end if
if $persistence$==1 **then**
$cw_i \Leftarrow CW_{min}$
end if
end for
return CW
// The CW vector contains CW size of all nodes in the network
end function

and load. If the channel is detected as free then the data packet is sent. If channel is busy, then the transmission is deferred by calling the backoff handler without changing the cw_i value of the node. This is important, to retain the node's persistence.

If two or more nodes with same persistence try to transmit by sensing the channel at the same time, then it might result in collision because all the contending nodes sense the channel for same DIFS period, detect the channel as idle and every one start transmission. Hence to overcome such scenarios we make a node wait for a random number of slots between 0 and cw_i before acquiring the channel. The cw_i value that is calculated using the topological persistence will not change with the success or failure of a packet transmission. A change in a node's persistence only occurs with a change in topology (movement of nodes) and/or demand.

The two differences between Algorithm 5 when compared with IEEE 802.11 style of channel access are:

1. Using extra random time slots to wait before accessing the channel, and
2. The cw_i value change is independent of collision or successful transmission of a packet. A change occurs only with a change in persistence which is caused by change in the topology or the demand.

Algorithm 5 Channel_Access_With_New Backoff

---

procedure CHANNEL_ACCESS
▷ // Channel access with new backoff strategy
    while *any node i has a packet to transmit* **do**
    // get the iˆth value in the CW vector.
        $sensing\_timer \leftarrow DIFS + (Random\ number\ of\ slots\ in\ the\ interval\ [0, cw_i]) * slottime$;
        sense the channel to see it is free
        if *channel* is idle **then**
            Node *i* waits until the *sensing_timer* expires
            //This confirms that the channel is free and ensures the scheduling when two or more nodes try to sense the channel at same time.
            if *channel* continues to be free after *sensing_timer* time **then**
                Node *i* gains the channel for transmitting the packet
            else if *channel* is not idle **then**
                Backoff_Handler(cw_i)
            end if
        else
            Backoff_Handler(cw_i)
        end if
    end while
end procedure

Algorithm 6 is the backoff handler function which works identically to IEEE 802.11 backoff handler. A random number of slots are selected in the interval 0, $cw_i - 1$ of the node and wait until that slot time expires. After the expiration of this waiting time, node will again try to gain the channel using Algorithm 5, if it has a packet to transmit.

Algorithm 6 Backoff Handler

procedure BACKOFF_HANDLER(cw_i)
    $Timer \leftarrow (Random\ number\ of\ slots\ in\ the\ interval\ [0, cw_i]) * slottime$;
    while $Timer > 0$ **do**
        wait;
        $Timer \leftarrow Timer - 1$
    end while
    if $Timer == 0$ **then**
        return;
    end if
end procedure

3.7 Summary

In this chapter we present the algorithms that we use in implementing the new backoff strategy using topological persistence for wireless networks. We started the discussion by explaining the role of the central node. We then describe the calculation of topological persistence using the centralized approach, and provide the details of converting the topological persistence values into the CW size. The final part introduces how a node accesses channel using the new backoff strategy that takes into account both topology and demand.

In the next chapter, we provide the details of the experimental setup and the experimental results of using the new backoff strategy that we implemented. The performance evaluation compares the new backoff algorithm to IEEE 802.11 using BEB in single-hop and multi-hop scenarios which include both the cases of static and mobile networks along with the changes in load.

Chapter 4

Performance Evaluation

4.1 Overview

In this chapter, we explain the simulation setup and the performance results of the new backoff strategy in several network scenarios. We start the discussion by providing the details of *network simulator* `NS-2 2.34`, the setup parameters that were used, and the evaluation metrics we considered. We then provide the results of all the single-hop scenarios, then multi-hop scenarios, and finally MANETs.

4.2 Network Simulator & Parameter Setup

The network simulator NS-2 is a discrete event simulator that can simulate network protocols, network scenarios, and using this software, new protocols can also be implemented and evaluated. The simulation study is carried out on the network simulator NS-2 version 2.34 [1]. In all the different network scenarios which we discuss in the next sections, the simulation parameters we used are the same unless explicitly specified.

We use a data channel rate of 1 Mbps, and each data packet is of size 256 bytes. We assume that the transmission range of a wireless node (250 m) is different from its carrier sensing range (550 m). Each flow in the network is saturated, that is, a transmitter always has a packet to transmit so that the channel is used at its full capacity. Traffic is generated by the `NS-2` *constant bit rate (CBR)* generators and the transport layer support is given by UDP. Table 4.1 lists all the simulation parameters that we used in the experiments.

Using the software *Design Expert* [24], values of the parameters CW_{min}, CW_{max} and packet size are chosen by conducting a screening experiment and then performing an ANOVA analysis on the responses of average throughput and average delay.

Parameter	Value
Simulation Area	1500m $\times$ 300 m
Transmission Range	250 m
Carrier Sensing Range	550 m
Simulation Time	300 sec
Packet Size	256 bytes
CBR rate	1Mbps
Traffic Type	UDP
CW_{min}	32
CW_{max}	1024
Number of Iterations	10
Antenna	Omni-directional
Channel data rate	1 Mbps
Routing Protocol	AODV
MAX Retry count	7

Table 4.1: Simulation Parameters.

Evaluation Metrics

In all the scenarios, we evaluate the performance of our new backoff strategy by comparing the responses of average throughput, average end-to-end packet delay, and delay variance with that of the IEEE 802.11 using the *BEB* protocol. Throughput is defined as the amount of data received from the transmitted packets in a given time, expressed in *bytes/bits per second* (bps). Average throughput per flow f is defined as the average of the throughput obtained at the receiver of the flow f for 10 iterations . Overall average throughput in a given network is the average of the average throughput on all the flows in the entire network. End-to-end delay is defined as the time taken for a packet to reach the destination from a source. End-to-end delay per flow is defined as the time taken for transferring the packets of flow f that are received at the receiver end. Delay is measured in terms of *milliseconds (msec)*. Average end-to-end packet delay per flow f is defined as the average of the packet delay over the flow f for 10 iterations. Overall average delay in the network is the average of the average delay on all the flows in the entire network. *Delay variance* measured in square of the milliseconds $(msec)^2$, is the variance in the end-to-end delay between the received packets in a flow with any lost packets being ignored [11].

4.3 Static Scenarios

First, we present the performance results of single-hop scenarios.

Exposed Terminal

Consider a topology of 4 nodes as shown in Figure 4.1 where receiver B in the Flow 1 can sense transmitter C of Flow 2. Transmitters A and C are not in transmission range or carrier sensing range. In IEEE 802.11, the Flow 1 throughput is very low because both RTS packets from A and C collide at B which cannot respond with a CTS to A. On the other hand, the throughput on Flow 2 is high because transmitter C can sense the CTS packet from D. Here B is exposed to the CD transmission which makes C aware of the Flow 1, and node A not aware of the Flow 2. This scenario is the classic exposed terminal scenario and is an example of an information asymmetry. We choose this scenario because we know IEEE 802.11 can create exposed terminals resulting in unequal throughput. Our interest is to see if our new backoff strategy yields better response.

Using the new backoff strategy with topological persistence, the persistence of nodes A and C, and the *CW* size we obtained are shown in Table 4.2. We can observe that the persistence of nodes A and C are the same. This means both the nodes are permitted to transmit with equal frequency.

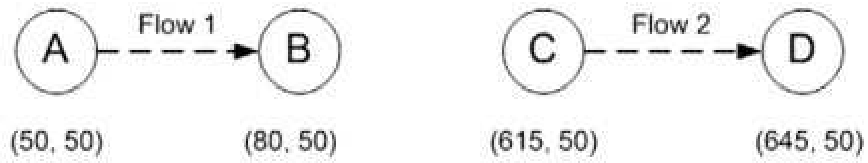

Figure 4.1: Exposed Terminal scenario.

The results of the exposed terminal scenario when both A and C transmit by using IEEE 802.11 with BEB, and by using new backoff strategy are shown in Figures 4.2, and 4.3. In the Figure 4.2, evaluation of the two protocols is done by comparing the average throughput per flow and in the Figure 4.3, the evaluation with respect to average delay per flow is shown.

Transmitter	**Persistence**	**CW size**
A	0.46	137
C	0.46	137

Table 4.2: Persistences and contention window sizes for nodes A and C using Algorithm 3 and Algorithm 4.

In Figure 4.2, we see that by using the new backoff strategy, the Flow 2 improved its throughput by approximately 100% compared to BEB and at the same time Flow 1 has gained the throughput and became about equal to that of the Flow 2, as their persistences are same. The error bars shown indicate the standard deviation, and we can notice that the value of standard deviation is about 97.6% lower for the new backoff strategy when compared with that of the IEEE 802.11 using BEB. The overall average throughput gain by using the new backoff is about 4 times that of the overall average throughput by using the IEEE 802.11 with BEB.

In the Figure 4.3, we observe that the new protocol reduced the average end-to-end delay of Flow 2 from 101 msec to 3.45 msec and the average packet delay of Flow 1 increased to 5.82 msec from 0.519 msec. Overall average end-to-end delay of this network reduced from 51.1 msec to 4.8 msec. The results of the delay variance in the exposed terminal scenario are shown in the Table 4.4. The delay variance by using the new backoff reduced from 5112.73 to 46.85 $(msec)^2$ which indicates a significant performance gain. In the Table 4.3 we tabulate the fairness results using Jain's fairness index (JFI) [14] which is calculated by considering the throughput on each flow. JFI is the fraction of square of the overall average throughput in the network to n times the sum of squares of the average throughput per flow, where n indicate the number of flows in the network, i.e., JFI=$\frac{(\sum_{i=1}^{n} x_i)^2}{n \times \sum_{i=1}^{n} x_i^2}$ where x_i indicate the average throughput over flow i. We can observe over a 40% of increase in fairness with the new protocol.

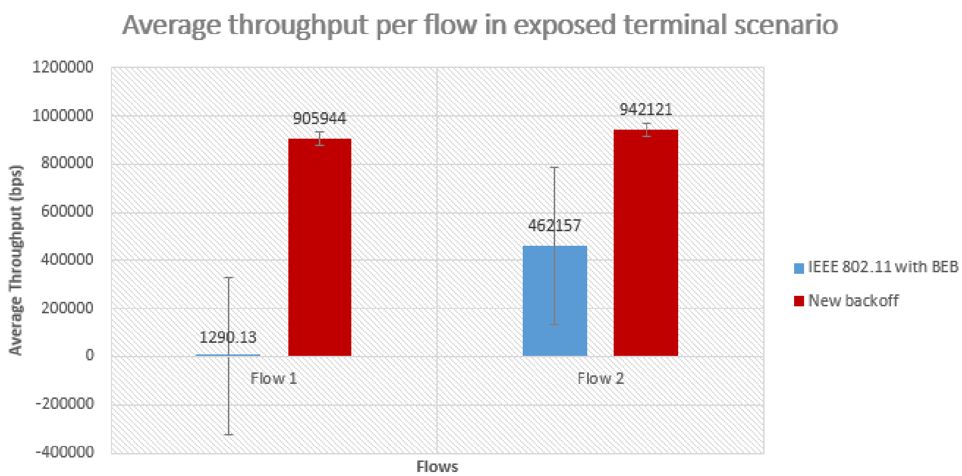

Figure 4.2: Average throughput of flows 1 and 2 in the exposed terminal scenario.

Protocol	**JFI value**
IEEE 802.11 with BEB	0.50
New backoff	0.99

Table 4.3: JFI in the exposed terminal scenario.

In the new protocol the contention window size cw_i at each node i corresponds to its topological persistence value s_i. As the persistence is influencing the backoff, a

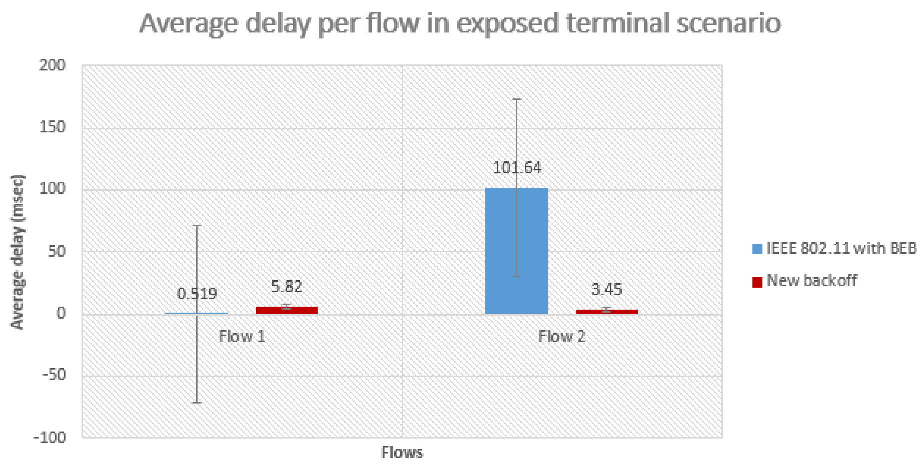

Figure 4.3: Average delay of flows 1 and 2 in the exposed terminal scenario.

Protocol	**Delay Variance** $(msec)^2$
IEEE 802.11 with BEB	5112.73
New backoff	46.85

Table 4.4: Delay variance of the two backoff schemes in the exposed terminal scenario.

node transmits according to its persistence. This way of transmitting after finding the persistence based on topology and demand reduces the information asymmetry. In the exposed terminal scenario, as the nodes A, C have same persistence values, they compete for the channel with about the same frequency. The calculated cw_i value of both the nodes A, and C is the same and will remain the same irrespective of the packet transmission status. As the cw_A, cw_C of the nodes are equal and do not change, node A also gets the same chance to transmit which results in the higher performance gain.

Thus in this scenario we conclude that the new backoff outperforms the IEEE 802.11 with BEB.

Flow in the Middle

Consider the network scenario as shown in Figure 4.4 from [10]. Here, we have three flows: from A to B, C to D, and E to F, respectively. We can observe that the node C is in carrier sensing range of both A and E, similarly node D is in carrier sensing range of node B and F. This scenario is an example of classic of unfair scenario when BEB is used for collision resolution. Hence we would like to consider this scenario to see whether the new backoff can improve the network performance.

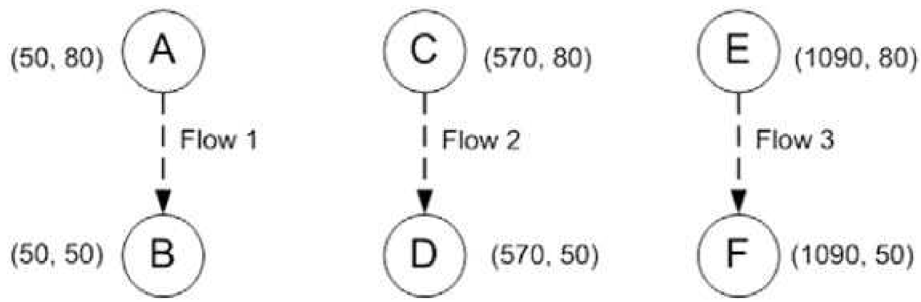

Figure 4.4: Flow in the middle scenario.

With the new protocol, the persistences and the contention window sizes of nodes A, C, E as calculated by Algorithm 3 and by Algorithm 4 are shown in the Table 4.5. Similar to the exposed terminal scenario, we can observe that persistences of nodes A, C, E are the same which gives equal opportunity for all the nodes to transmit, improving the fairness. Evaluation results of the new backoff protocol and IEEE 802.11 protocol are shown in the Figures 4.5 and 4.6.

Using the IEEE 802.11, Figure 4.5 shows that we observe good throughput over Flow 1 and Flow 3, whereas the Flow 2 throughput is very low. This is because node C senses the transmission from A and E. When A or E transmissions are successful, the CW size of node A or E is reset to CW_{min} but node C increases its CW size reaching CW_{max} in no time, resulting in C not being able to acquire the channel. Similarly if node C acquires the channel, both the nodes A and E cannot send packets and increase their CW. This fluctuation of the CW, reduces the overall network performance.

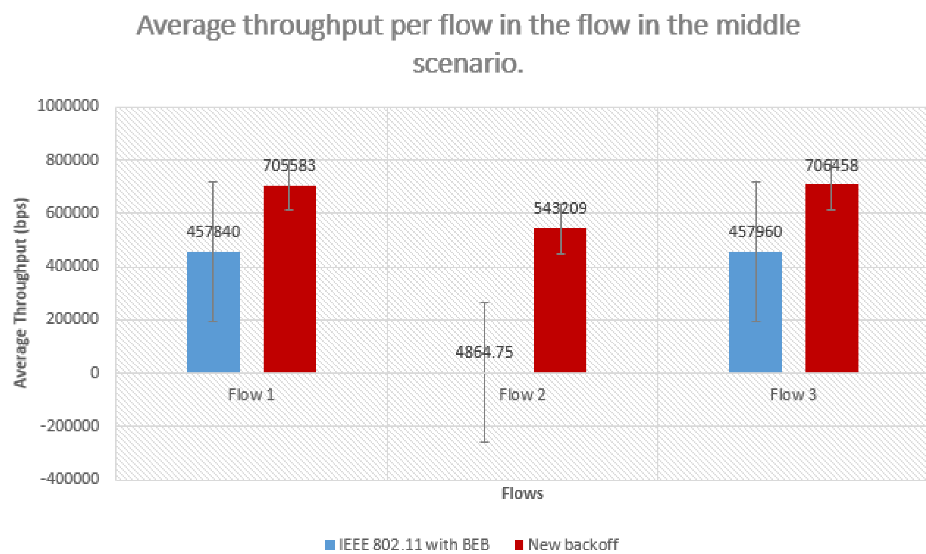

Figure 4.5: Average throughput for flows 1-3 for the flow in the middle scenario.

From the Figure 4.5, we observe that the throughput on the Flow 2 (C to D) has increased by nearly 95% by using the new protocol. The overall average throughput of the network increased from 306 Kb to 651.75 Kb which is more than double the overall average throughput of IEEE 802.11. We even notice the standard deviation is about 64% lower for the new backoff strategy when compared with that of the IEEE 802.11 using BEB.

In the Figure 4.6, we see that the average delay of Flow 2 (C to D) increased slightly from 95 msec to 101 msec by using the new backoff. The overall end-to-end average delay in the network by using IEEE 802.11 is 99.63 msec whereas using the new backoff strategy it is 100.5 msec. Hence the overall end-to-end average delay of the network is almost same in both the protocols. The delay variance results are shown in Table 4.7. We observe that the delay variance in the new backoff for this scenario is 0.21 $(msec)^2$ whereas for IEEE

802.11 with BEB it is 11.43 $(msec)^2$. Table 4.6 shows Jain's fairness index, for the flow in the middle scenario. Again, we observe that the fairness index increased by more than 40% than that of the IEEE 802.11.

Transmitter	**Persistence**	**CW size**
A	0.31	208
C	0.31	208
E	0.31	208

Table 4.5: Persistence and contention window size for transmitters A, C, and E in the flow in the middle scenario.

Protocol	**JFI**
IEEE 802.11 with BEB	0.67
New backoff	0.98

Table 4.6: JFI values of each backoff strategy for the flow in the middle scenario.

The performance gain we noticed in the results is because of the new backoff strategy where the contention window (CW) size has been calculated by implicitly considering topology and demand of the neighboring nodes as factors. This information sharing of all nodes about contention reduces the information asymmetry.

Protocol	**Delay Variance** $(msec)^2$
IEEE 802.11 with BEB	11.43
New backoff	0.21

Table 4.7: Delay variance of each backoff strategy for theflow in the middle scenario.

All the transmitters (A, C, E) have same persistence and CW values which indicate that every node has an equal chance in accessing the channel. Moreover the nodes are stationary, which means their CW values which are calculated once do not change. So node C cannot increase the CW size even if it cannot access the channel. This gives a chance for the node C to transmit an almost equal number of times as that of nodes A and

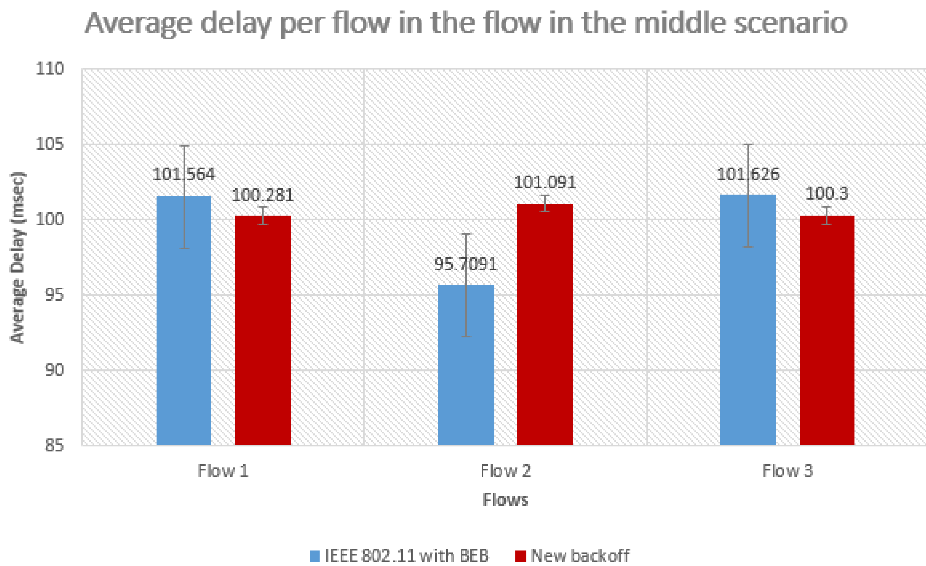

Figure 4.6: Average delay of flows 1-3 for the flow in the middle scenario.

E, due to which we observe the high increase in the network performance especially with the overall average throughput and delay variance.

Star Topology

A star topology such as the one shown in Figure 4.7 is considered for evaluating the performance of new backoff strategy. In this topology, all the nodes A, B, C, and D in the network try to transmit data to the base node or central node E. We assume all the nodes are stationary. Star topology, using IEEE 802.11 with BEB, keeps high channel access fairness rates but suffers from low performance due to high collisions occurring at the central node. We would like to check whether our new backoff improves the performance by reducing the collisions in star topology.

Using the new backoff strategy, the persistence and the *CW* size of each transmitter in the star topology is shown in the Table 4.8. A comparison of the results of both the protocols with respect to the average throughput and average delay per flow are shown in

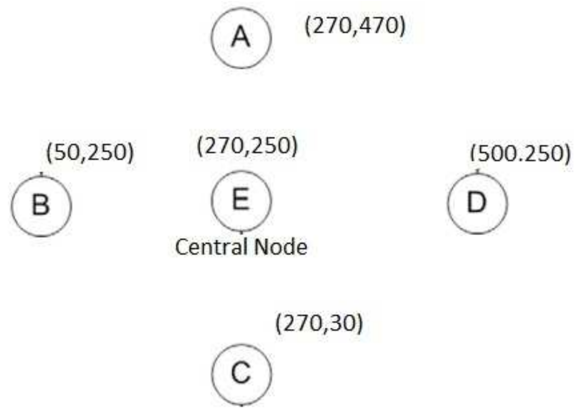

Figure 4.7: Star Topology.

Figures 4.8 and 4.9, respectively.

Transmitter	Persistence	CW size
A	0.25	258
B	0.25	258
C	0.25	258
D	0.25	258

Table 4.8: Persistence and contention window size for transmitters A, B, C, and D in the star topology

When all nodes transmit the data to the central node, using IEEE 802.11 with BEB, the overall average throughput in Figure 4.8 is very low because of the heavy packet collisions that occur at the central node. Whereas in the new protocol, we observe that the overall average throughput obtained is about 3.65 times higher than that in IEEE 802.11 with BEB.

In the Figure 4.9 , we observe that the end-to-end packet delay on each flow as well as for the overall network is about same in both IEEE 802.11 with BEB and new backoff using topological persistence. The new backoff improved the channel access of all nodes and reduced the collisions due to which throughput is high. Even though more packets were able to get through, the delay for each packet is same and so the average delay per

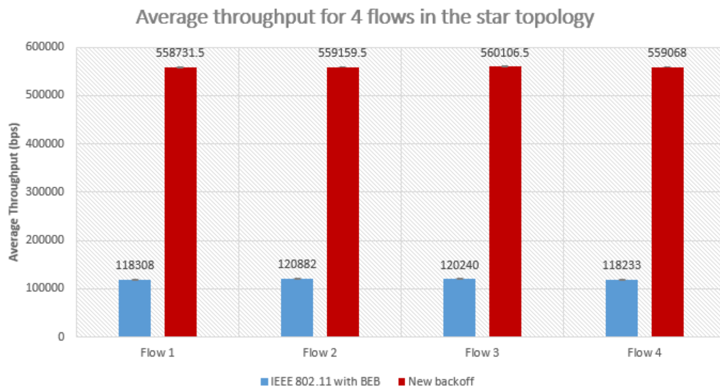

Figure 4.8: Average throughput for flows 1-4 of the star topology.

flow is same. From the Table 4.9, the delay variance using both the protocols is shown. We observe that using the new backoff strategy, delay variance is twice more than that of the IEEE 802.11 using BEB. We tabulate the JFI values for the two backoff strategies in Table 4.10. We observe the fairness index in both the backoff strategies is the same.

Protocol	**Delay Variance** $(msec)^2$
IEEE 802.11 with BEB	0.01
New backoff	0.02

Table 4.9: Delay variance for each backoff strategy in the *star topology* scenario.

Protocol	**JFI**
IEEE 802.11 with BEB	0.99
New backoff	0.99

Table 4.10: JFI values of each backoff strategy for the star topology.

In IEEE 802.11 with BEB, a node access the channel greedily. As a result, many collisions occur at the central node E which doubles the *CW* of the transmitter due to which waiting time increases. In the new protocol, the persistence and the *CW* values of all the transmitters (A, B, C, and D) are same. Hence all the nodes transmit data using their persistence, an almost equal number of times. Due to this more equal opportunity, removal of the greedy nature in accessing the channel, and making the *CW* independent of

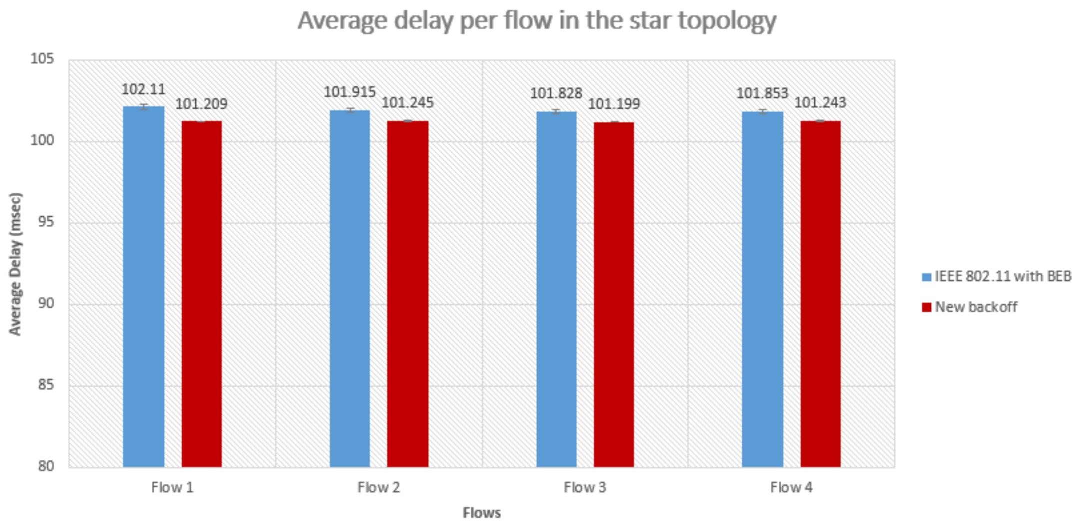

Figure 4.9: Average delay for flows 1-4 of the star topology.

packet transmission status by making it dependent only on the topology and demand, the collisions at the central node are reduced, improving the overall average throughput.

Random Single-hop Networks with High Load

We distributed 50 nodes randomly in the 1500m $\times$ 300m area. We then picked 25 distinct transmitters and receivers randomly which are in one hop distance.

Simulation is carried with this setup and the evaluation results are shown in the Figures 4.10 and 4.11. The persistence values of most of the transmitters in this scenario are very low because of the high density, which means all these nodes get the same CW size which is equal to CW_{max}.

In the Figure 4.10, we observe the overall throughput using the new backoff strategy is about six times more than that of the IEEE 802.11 with BEB. From the Figure 4.11, the overall average end-to-end packet delay of the random network with 25 transmitters in the IEEE 802.11 with BEB we obtained is 79.01 msec whereas the overall average end-to-end packet delay in the new backoff is 106.13 msec. In Table 4.11, the delay variance results

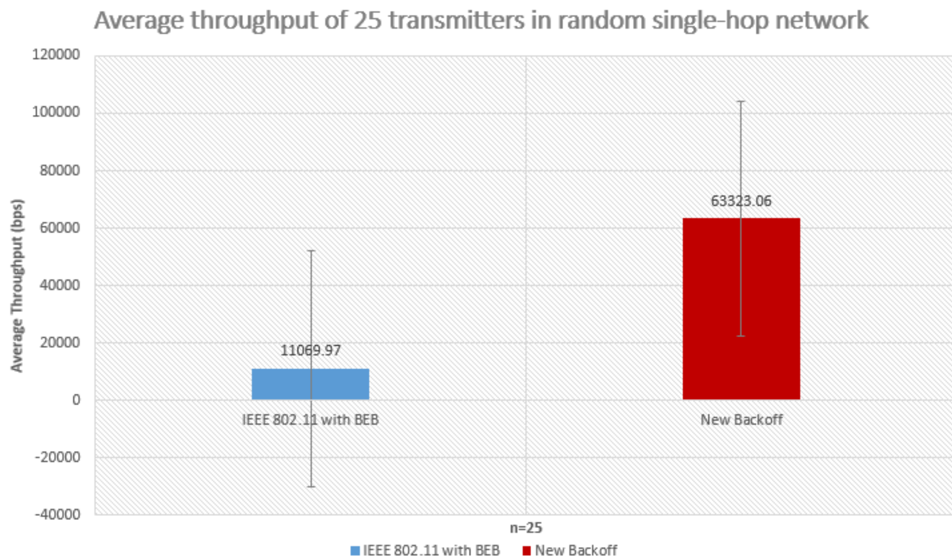

Figure 4.10: Overall average throughput in the random single-hop network with 25 flows.

are shown. Even though the average delay with new backoff is about 20% more than IEEE 802.11, from the Table 4.11, we observe that the delay variance in new backoff strategy is 186.9 $(msec)^2$ which is much less when compared with the delay variance of the IEEE 802.11 with BEB whose value is 349.3 $(msec)^2$. Thus by using the new backoff we observe a significant improvement in the overall network performance.

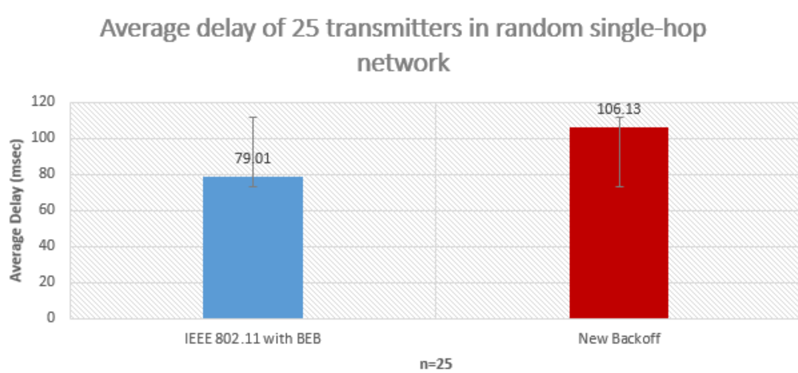

Figure 4.11: Overall average delay in the random single-hop network with 25 flows.

In IEEE 802.11 with BEB, a node tries to access the channel as soon as it has a packet to transmit. If the channel is detected to be busy in this scenario of heavy load, the node backoff and quickly the CW value reaches CW_{max}, resulting in the reduction of the overall performance. In the new backoff strategy, collisions are reduced because the channel is accessed using the node's persistence, and CW is independent of the packet transmission status. Moreover the transmitters which might be suffering with the unfairness issue in IEEE 802.11 with BEB will now get the chance to transmit according to their persistence.

IEEE 802.11 achieves less throughput with large packet drop rate, due to which the delay is very low. In the new backoff strategy, higher throughput is obtained which means more packets are getting through and hence the delay is higher.

Protocol	**Delay Variance** $(msec)^2$
IEEE 802.11 with BEB	349.3
New backoff	186.9

Table 4.11: Delay variance for each backoff strategy in the random single-hop network with 25 flows.

4.4 Static Multi-hop Scenarios

Here we describe the details of the multi-hop scenarios that we considered. All the nodes of the network in these scenarios are stationary.

Small Multi-hop Network

Consider a topology with 7 nodes as shown in the Figure 4.12. The node A is transmitting to the node F, and node B is transmitting to the node G. Nodes C, D, and E are the forwarding nodes. In multi-hop networks, intermediate nodes play an important role in the overall network performance. In the Figure 4.12, because the nodes C, D, E are bottlenecks IEEE 802.11 drops a lot of packets reducing the overall network performance. We choose this scenario to see how new backoff strategy handles the situation of bottleneck nodes in the multi-hop scenarios. Persistence values of the nodes are shown in the Table 4.12.

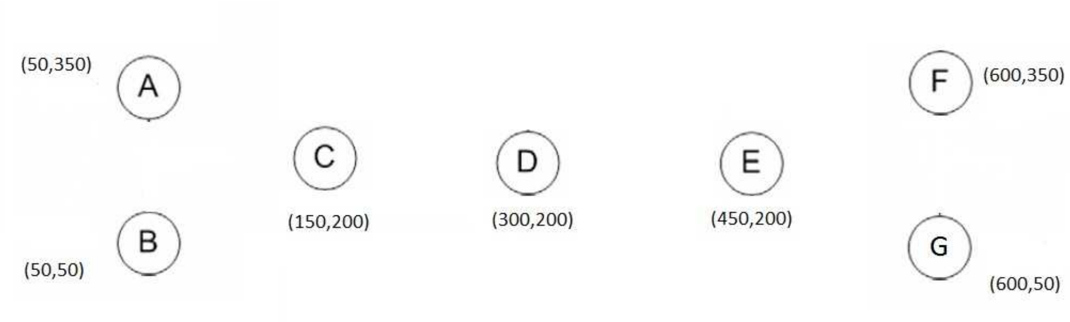

Figure 4.12: Topology for the small multi-hop scenario .

Transmitter	**Persistence**	**CW size**
A	0.13	497
B	0.13	497
C	0.24	269
D	0.24	269
E	0.24	269

Table 4.12: Persistence and contention widow size of nodes A, B, C, D, E in the small multi-hop network.

The simulation results for this scenario using our protocol and the IEEE 802.11 with BEB are presented in the Figures 4.13 and 4.14. In the Table 4.13 the delay variance of the both protocols is presented. In multi-hop scenarios, if the persistence of intermediate nodes is low, the network performance will also be very low. In this scenario, the arrival rate of the intermediate nodes C, D, and E is double the arrival rate of A, and B so that we can see the persistence is also doubled for these nodes.

In the Figure 4.13, we observe that with our protocol, the overall average throughput is about four times more than the overall average throughput in IEEE 802.11 with BEB. From the Figure 4.14, the overall average end-to-end delay by using the new backoff is about 4% lower than the overall average delay by using IEEE 802.11 with BEB. From the Table 4.13, the delay variance of new protocol is 0.03 $(msec)^2$ and delay variance of IEEE 802.11 is 0.66 $(msec)^2$ which indicate an approximate 95% of reduction in delay variance.

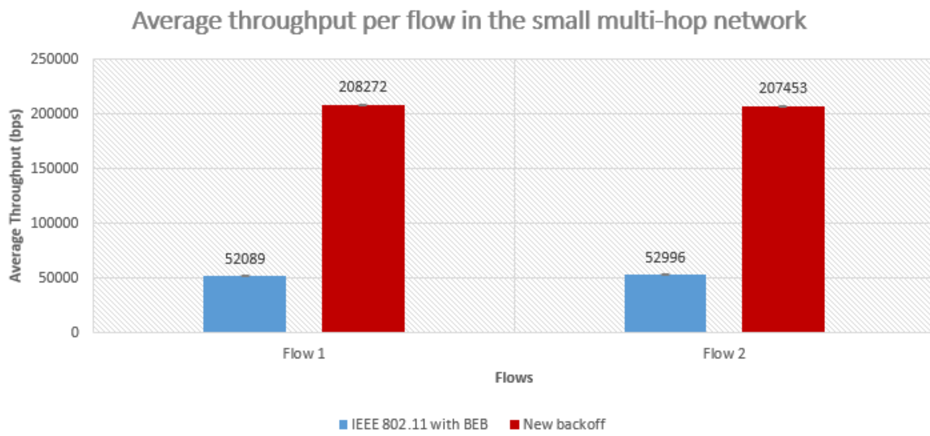

Figure 4.13: Average throughput for flows 1 and 2 of the small multi-hop network.

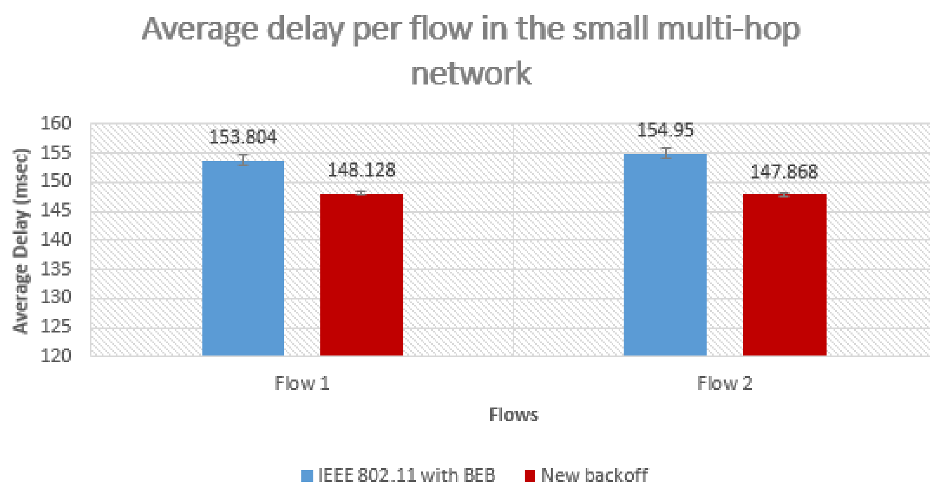

Figure 4.14: Average delay for flows 1 and 2 of the small multi-hop network.

Protocol	**Delay Variance** $(msec)^2$
IEEE 802.11 with BEB	0.66
New backoff	0.03

Table 4.13: Delay variance for each backoff strategy in the small multi-hop network.

In IEEE 802.11 with BEB, many collisions tend to occur at the intermediate node C. As a result both the nodes A and B increase their CW value reducing the channel access rate thereby resulting in lower throughput. As the new backoff strategy removes the dependency of the *CW* on the packet transmission status, *CW* values never change for A and B and so the overall network performance in terms of average throughput and average delay is improved.

Random Multi-hop Network with Increasing Load

In this experiment, to simulate a multi-hop scenario we first randomly distribute 50 nodes in the simulation area and then select the transmitters and receivers at random such that no two transmitters or receivers are same. We then increase the load in the network by increasing the number of transmitters from 5 to 25 in multiples of 5.

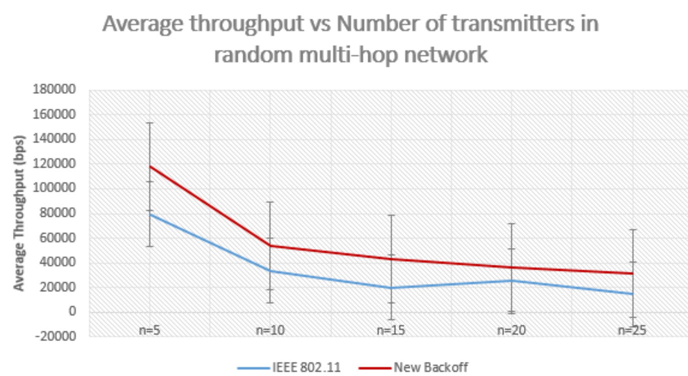

Figure 4.15: Overall average throughput vs number of transmitters in the random multi-hop scenario.

In the Figure 4.15, we present the overall average throughput of the network achieved by the IEEE 802.11 with BEB and the new backoff scheme. We can see the overall average throughput in both the protocols is reducing with the increase in the load. When load is very low, we observe the average throughput using new backoff is almost 35% more than the overall throughput using IEEE 802.11 with BEB. As the load is increased and made to reach maximum (25 transmitters), the overall average throughput of IEEE 802.11 reduced to a low value of 11 Kb whereas using the new protocol, the overall throughput is 31 Kb, which is almost three times higher. Thus the new backoff strategy outperforms IEEE

802.11 in the multi-hop scenario especially in the higher loads, in terms of the average throughput.

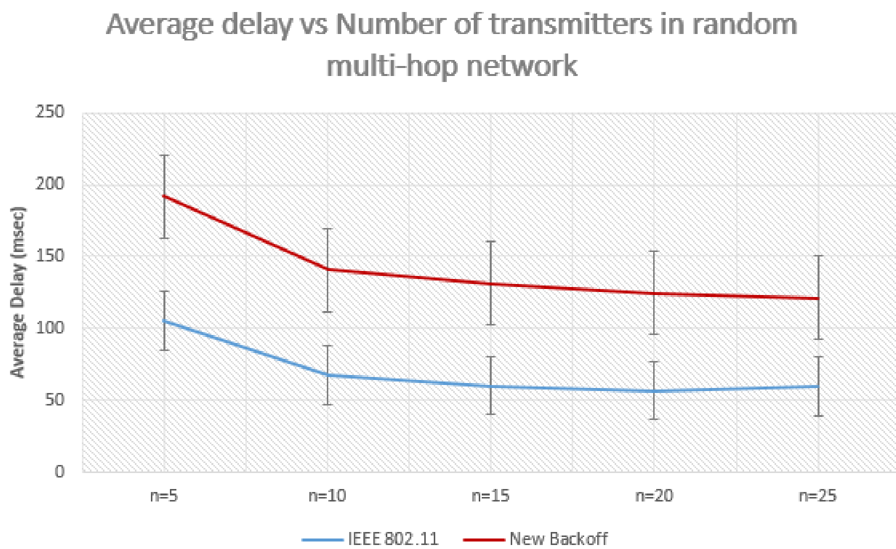

Figure 4.16: Overall average delay vs number of transmitters in the random multi-hop scenario.

Figure 4.16 presents the average delay in the network for IEEE 802.11 with BEB and the new backoff strategy. We observe the average end-to-end packet delay using new backoff strategy is almost double the average delay using IEEE 802.11. IEEE 802.11 achieves much less throughput especially in the heavy load. Each packets that gets through has a very low delay but most of the packets are dropped. Whereas in the new backoff strategy, higher throughput is obtained as more packets are getting transmitted and hence we observe the delay to be higher. The new backoff achieves an overall delay less than 200 msec.

The Table 4.14 the delay variance results of the two protocols with increasing load are presented. We observe that the delay variance in lower load is about 30% higher for the new backoff strategy than for the IEEE 802.11. But when the load is increasing, the delay variance of the new backoff strategy is much lower than the delay variance of the

IEEE 802.11 using BEB. When the load is maximum with 25 transmitters, delay variance for new backoff is 1013.87 $(msec)^2$ whereas the same for IEEE 802.11 with BEB is 1482.6 $(msec)^2$ which indicate about a 32% lower delay variance with new backoff strategy. Due to heavy packet drop in IEEE 802.11 a much less throughput is obtained especially in the heavy loads resulting in the overall delay and delay variance being low but in the new backoff strategy, the number of packets dropped is less and so we see a higher throughput and higher delay resulting in the higher delay variance.

Number Of Transmitters	**Delay Variance $(msec)^2$ in IEEE 802.11**	**Delay Variance $(msec)^2$ in New Backoff**
5	2387.7	3441.0
10	1976.0	1721.9
15	2027.9	1334.0
20	1446.3	1509.8
25	1482.6	1013.8

Table 4.14: Delay variance in both backoff strategies for different load in random multi-hop network.

Thus from the results of the random multi-hop network with increasing load, we observe that the overall average throughput of IEEE 802.11 is reduced with the increasing load because of more collisions and packet drops. Whereas the new backoff strategy maintained high overall throughput with the increasing loads, by reducing the packet drops.

4.5 MANET Scenario

MANET with varying load

In this section, a mobile ad hoc network is considered for the evaluation of our protocol. To simulate the MANET, 50 nodes are randomly placed in a 1500m $\times$ 300m area. In the experiment, we randomly choose n transmitters and n receivers, where n is from 5 to 25 in the multiples of 5. We now use the random way-point mobility model steady state to generate the random movement of all the nodes [3]. Speed of nodes is selected randomly from 5 m/sec to 25m/sec. Along with the mobility we even varied the CBR rate of each flow randomly between 100 Kbps to 1Mbps to include the evaluation of the new backoff strategy in the varying demand scenario.

As we are using the centralized approach for the calculations of the topological persistence, and the CW size, we call the algorithm for calculation at a regular interval of 10 msec. If the calculation changes the persistence of any node, the change will be automatically reflected in the node's CW. With this network setup, we start the simulation study using both the new backoff strategy and the IEEE 802.11 with BEB.

In Table 4.15, we show the statistics (min, max) of the persistence values of nodes in the MANET with 25 flows moving at an average speed of 5 m/sec. We observe that the persistence values are low because of the high density of nodes in the network.

Minimum Persistence	**Maximum Persistence**	**At Simulation Time (sec)**
0.0119	0.0385	50.0
0.0118	0.0329	100.0
0.0134	0.0283	150.0
0.0167	0.0201	200.0
0.02	0.02	250.0
0.02	0.02	300.0

Table 4.15: Min and Max persistence of nodes in the MANET with 25 flows at different simulation time.

In the Figure 4.17, we present the overall average throughput of the network for each n between the IEEE 802.11 with BEB and the new backoff. We can see the overall average throughput is reducing with an increase in the load in both the protocols. When the load is very low, the average throughput using new backoff strategy is almost equal with that of the IEEE 802.11 with BEB. But as the load increases the average throughput

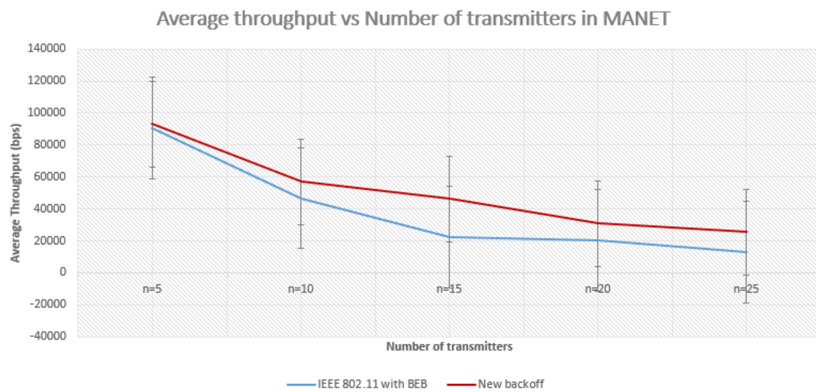

Figure 4.17: Average throughput for a MANET as a function of the number of transmitters.

by using new backoff strategy is around 30% higher than the average throughput in IEEE 802.11 with BEB.

In the Figure 4.18, we present the average end-to-end delay results of both the protocols. The average delay of the new protocol is twice the average delay of the IEEE 802.11. In IEEE 802.11, the number of packets that dropped are high and so a few packets are transmitted successfully resulting in lower throughput and lower delay. But in the new backoff strategy, number of packets being transmitted are more and so the throughput is increased resulting in higher delays.

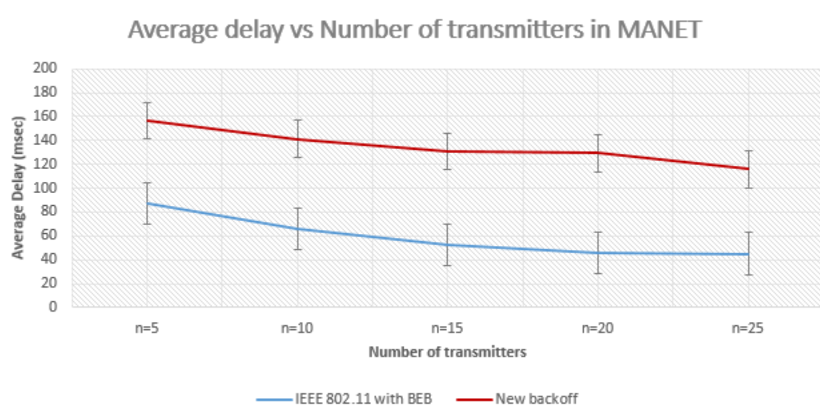

Figure 4.18: Average delay for a MANET as a function of the number of transmitters.

The Table 4.16 shows the delay variance results of the two protocols with increasing load in MANET. When the load is low, the delay variance in new backoff is higher. But when load increases, we observe a huge improvement in the delay variance of the new backoff. Especially when the load is maximum with 25 transmitters, delay variance for new backoff is 553.1 $msec^2$ whereas the same for IEEE 802.11 with BEB is 1214.9 $msec^2$.

Number Of Transmitters	**Delay variance** $(msec)^2$ **in IEEE 802.11**	**Delay variance** $(msec)^2$ **in New Backoff**
5	1631.41	2898.78
10	1562.12	927.43
15	1296.77	1022.14
20	1357.90	1163.67
25	1214.99	553.10

Table 4.16: Delay variance between IEEE 802.11 with BEB and new backoff for increasing number of transmitters in a MANET.

In Figure 4.19, we present the results that show the distribution of the average throughput with the average speed of the nodes in the MANET. We vary speed from 5 m/sec to 25 m/sec in the multiples of 5 and collect the overall average throughput results for different number of flows. As the average speed increases, the overall throughput of network decreases. When the node movement is slow, the sender receiver range changes slowly so the communication will persist for more time resulting in higher throughput. But when the speed of the node is increased the sender receiver range fluctuates quickly making the nodes out of range resulting in the decrease of overall throughput.

Thus from the results, the new backoff strategy maintained high overall throughput with the increasing loads, and fluctuating speeds and demands, by reducing the collisions and packet drops.

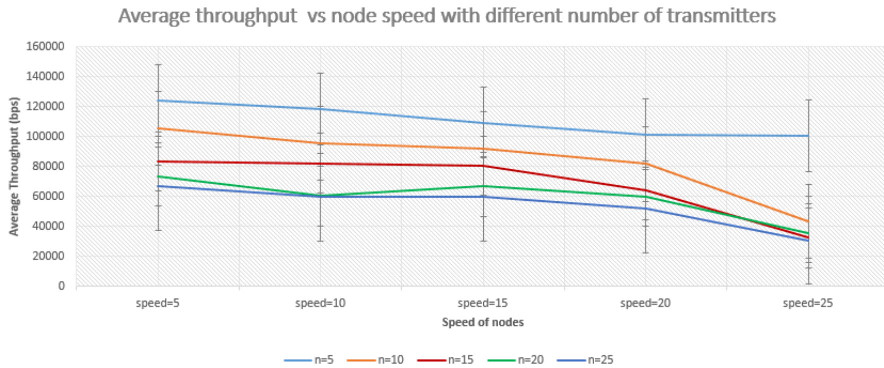

Figure 4.19: Average throughput for a MANET as a function of the speed of movement of nodes and number of transmitters.

4.6 Summary

In this chapter we provided the evaluation results of the implemented new backoff strategy that uses topological persistence in wireless networks. We first evaluated in single-hop network scenarios including the exposed terminal, flow in the middle, star topology and random single-hop network scenarios. We then evaluated static multi-hop networks, and random multi-hop network with increasing load. Finally, we evaluated the performance in a mobile ad hoc network MANETs by varying the load, speed and the arrival rate. From the analysis of results, we conclude that the new backoff strategy improves the network performance especially in the single-hop networks, in the multi-hop networks with high load and also in the MANETs with varying loads. The next chapter provides a summary of the contribution of this thesis and discusses the ideas for future work.

Chapter 5

Conclusion & Future Work

5.1 Overview

In this chapter we summarize the contribution of this thesis. We then propose some future work for the research.

5.2 Conclusion

The aim of this work is to implement a new backoff strategy in wireless networks that uses the topological persistence. In the IEEE 802.11, a node greedily accesses the channel irrespective of the load or topology. This may result in collisions, short term unfairness, low network performance in heavily loaded networks and in MANETs. Many changes were proposed in IEEE 802.11 protocols to improve its performance by revising the backoff strategy like MILD [4], LMILD [12], SBA [13], EIED [23], that ignored topology and/or the demand. We first calculate node persistence considering the topology and demand, and then use these persistence values to devise a new backoff strategy. Because persistence is lexicographic max-min fair, this may address some of the performance issues of IEEE 802.11.

We calculated the topological persistence using a centralized approach. We then converted the calculated persistence values into contention window (CW) sizes to develop a new backoff algorithm. In this new backoff scheme, the contention window (CW) sizes of all the nodes does not change with the packet transmission status. The contention window (CW) size change only with a change in the persistence value of a node caused by a change in its neighborhood or its demand. As the contention window is aware of the topology and the demand, packet collisions can be reduced.

Our study shows that this new backoff strategy outperforms the BEB scheme in the IEEE 802.11. In the scenarios such as exposed terminal, flow in the middle, and in star topology where the traffic load is minimum, the overall average throughput of the new backoff strategy is about 100% higher than that of the BEB scheme in IEEE 802.11. Similarly, using the new backoff strategy the fairness index (JFI) is increased by about 40% when compared with that of IEEE 802.11 using BEB scheme. Especially in the single-hop networks and in the multi-hop network with high load, the performance of the new backoff scheme with topological persistence is about 50% higher than that of the performance of the BEB scheme. In MANETs, the new backoff strategy using topological persistence outperformed the IEEE 802.11 using BEB in terms of overall average throughput and delay variance especially in the higher loads. By using the new backoff strategy, the overall network performance in terms of throughput is approximately double the overall throughput of IEEE 802.11.

5.3 Future Work

Our research of using the topological persistence in the contention based MAC to devise a new backoff strategy can be improved further. Some improvements suggested are as follows.

In our research, to get the right parameter values, we conducted a small screening experiment on 3 parameters and assumed values for the rest of simulation parameters based on the simulation study. There are many more parameters that can be considered to determine the minimum and maximum *CW* sizes.

We used the centralized approach to calculate the topological persistence and contention window size with the assumption that the network is stable during the calculation. As a result, in mobile networks, the calculation is simply repeated periodically. This does not capture any change in topology or load that happen at a higher rate. Thus, the distributed approach can be implemented for the calculation of the topological persistence. Moreover the simulation study is conducted by considering only the UDP traffic generated by the CBR application ignoring the case of TCP traffic. Thus we leave the interesting study of this new backoff strategy with the TCP traffic as future work.

REFERENCES

[1] The Network Simulator - ns-2. [Online; accessed May-2012].

[2] G. Anastasi, M. Conti, and E. Gregori. IEEE 802.11 ad hoc networks: protocols, performance and open issues. *Mobile Ad hoc networking*, pages 69–116, 2004.

[3] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *Mobile Computing, IEEE Transactions on*, 2(3):257–269, 2003.

[4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LAN's. *SIGCOMM*, pages 212–225, 1994.

[5] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[6] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans. In *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, volume 2, pages 392–396. IEEE, 1996.

[7] C. Chaudet, D. Dhoutaut, and I. Lassous. Performance issues with IEEE 802.11 in ad hoc networking. *IEEE Communications Magazine*, 43(7):110–116, 2005.

[8] H. S. Chhaya and S. Gupta. Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol. *Wireless networks*, 3(3):217–234, Aug. 1997.

[9] M. S. Corson, J. P. Macker, and G. H. Cirincione. Internet-based mobile ad hoc networking. *IEEE Internet Computing*, 3(4):63–70, July 1999.

[10] M. Cui and V. R. Syrotiuk. Time-space backoff for fair node throughput in wireless networks using power control. *Ad Hoc Networks*, 8(7):767–777, Sept. 2010.

[11] C. Demichelis and P. Chimento. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). RFC 3393 (Proposed Standard), November 2002.

[12] J. Deng, P. K. Varshney, and Z. J. Haas. A new backoff algorithm for the IEEE 802.11 distributed coordination function. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation (CNDS '04)*, San Diego, CA, USA, January 18-21 2004.

[13] Z. J. Haas, S. Member, and J. Deng. On optimizing the backoff interval for random access schemes. *IEEE Transactions on Communications*, 51:2081–2090, 2003.

[14] R. Jain, D.-M. Chiu, and W. R. Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation, 1984.

[15] P. Karn. MACA-a new channel access method for packet radio. In *ARRL/CRRL Amateur radio 9th computer networking conference*, volume 140, pages 134–140, 1990.

[16] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part I–carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, 1975.

[17] M. Krunz, A. Muqattash, and S.-J. Lee. Transmission power control in wireless ad hoc networks: challenges, solutions and open issues. *IEEE Network*, 18(5):8–14, 2004.

[18] J. Lutz, C. J. Colbourn, and V. R. Syrotiuk. Apples and oranges: comparing schedule- and contention-based medium access control. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, MSWIM '10, pages 319–326, New York, NY, USA, 2010. ACM.

[19] J. Lutz, C. J. Colbourn, and V. R. Syrotiuk. Topological persistence for medium access control. *IEEE Transactions on Mobile Computing*, 99(PrePrints), 2012.

[20] J. Lutz, C. J. Colbourn, and V. R. Syrotiuk. ATLAS: Adaptive topology-and load-aware scheduling. *arXiv preprint arXiv:1305.4897*, 2013.

[21] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[22] R. Rozovsky and P. R. Kumar. SEEDEX: A MAC protocol for ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, pages 67–75, New York, NY, USA, 2001. ACM.

[23] N.-O. Song, B.-J. Kwak, J. Song, and M. Miller. Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff

algorithm. In *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, volume 4, pages 2775–2778. IEEE, 2003.

[24] I. Stat-Ease. *Design Expert software, Educational Version 7.0.3*. Wiley Publishing, 2007.

[25] A. S. Tanenbaum and D. J. Wetherall. *Computer networks*. Pearson Higher Ed, 2012.

[26] Z. Tang and J. Garcia-Luna-Aceves. A protocol for topology-dependent transmission scheduling in wireless networks. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1333–1337 vol.3, 1999.

[27] R. Wang, J. Zhang, and X. Zou. Performance analysis and optimization of IEEE 802.11 DCF with constant contention window. In *Proceedings of ISECS International Colloquium on Computing, Communication, Control, and Management, 2008. CCCM'08.*, volume 2, pages 407–411. IEEE, 2008.

[28] X. Wang and K. Kar. Distributed algorithms for max-min fair rate allocation in ALOHA networks. In *Proceedings of the 42nd Annual Allerton Conference*, 2004.

[29] C. Zhu and M. S. Corson. A five-phase reservation protocol (FPRP) for mobile ad hoc networks. *Wireless networks*, 7(4):371–384, Sept. 2001.

[30] E. Ziouva and T. Antonakopoulos. CSMA/CA performance under high traffic conditions: throughput and delay analysis. *Computer Communications*, 25(3):313–321, Feb. 2002.