Modeling the Interactions Between MAC and Higher Layer: A Systematic Approach to Generate High Level Scenarios from MAC Layer Scenarios

Shamim Begum
Department of Electrical Engineering
University of Southern California
Email: shamim.begum3@gmail.com
Ahmed Helmy
Department of Computer and Information Science and Engineering
University of Florida, Gainesville
Email: helmy@cise.ufl.edu
and
Sandeep Gupta
Department of Electrical Engineering
University of Southern California
Email: sandeep@usc.edu

We propose a new framework for worst case performance evaluation of MAC protocols for wireless ad hoc networks. Given a protocol, its performance metrics and a network topology, our framework first generates MAC scenarios which achieve poor performance at MAC-level. In order to evaluate the impact of these MAC scenarios on the end performance, we model the interactions between MAC interface and the MAC layer using a state transition graph and generate high level scenarios using enumeration techniques. These high level scenarios can be simulated and compared with heuristics developed by others to identify high level scenarios that are expected to lead to the worst case end performance.

In order to demonstrate its usefulness, we use our framework to evaluate the worst case performance of IEEE 802.11 DCF protocol by generating a library of MAC and high level scenarios. We simulate the high level scenarios to demonstrate that the scenarios we generate exhibit the worst performance among all the scenarios, including those generated by using heuristics recently proposed by other researchers.

Categories and Subject Descriptors: C.2.2 [**Computer Communication Networks**]: Network Protocols—*Protocol Verification*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Finite state machine models, IEEE 802.11 performance, medium access control, modeling for performance, worst case performance

1. INTRODUCTION

In the OSI reference model, medium access is the function of the layer-2 sub-layer called the Medium Access Control (MAC) layer. Sharing a limited communication bandwidth efficiently among all nodes in the network is the main objective of wireless MAC protocols. A 2004 survey [Jurdak 2004] presents 34 MAC protocols for

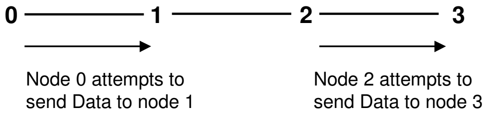


Fig. 1. A scenario leading to worst case unfairness in four node chain topology (Topology I).

wireless ad hoc networks ranging from industry standards (e.g., IEEE 802.11 [WG 1997]) to research proposals (e.g., MACAW [Bharghavan 1994]). Several studies analyze these protocols using analytical models (e.g., [Bianchi 2000, Gupta 2004, Abdrabou 2008, Bensaou 2000, Nandagopal 2000] for IEEE 802.11) and experience or intuition (e.g., [Li 2001, Barrett 2002, Pathma 2006] for IEEE 802.11). Simulation and testbed implementations are the main evaluation tools available today for which the *scenarios* are either manually generated (e.g., Bianchi [2000]) for average case analysis, or are based on intuition (e.g., [Li 2001, Gupta 2004, Abdrabou 2008, Bensaou 2000, Nandagopal 2000, Barrett 2002, Pathma 2006]). These evaluation approaches do not use any systematic approach for scenario generation. Often times, actual deployment of a protocol reveals many problems that cause unexpectedly poor performance.

The scenarios used in these works for analysis are either *MAC scenarios* analyzing the behavior at the MAC layer or *high level scenarios* capturing the effect of protocol behavior at the end level. However, high level scenarios are used for simulations and actual evaluation of performance. There has been no work that bridges the gap between these two levels of abstractions by generating scenarios at one level and transforming these to obtain scenarios to be used at the other level. The goal of this work is to augment other evaluation approaches with a library of meaningful high level scenarios. These high level scenarios are generated by transforming MAC scenarios generated automatically using generic algorithms for better evaluation of the protocols beyond average case and towards the worst case. In this paper, we refer to the effect of MAC performance (e.g., MAC throughput) on the end (user) application as *end performance.*

A wireless ad hoc network is required where a fixed communication infrastructure, wired or wireless, does not exist or has been destroyed. In recent years, ad hoc networks have been very useful in coordination of large scale emergencies, crisis response and military applications. Evaluating the entire operational region of the underlying communication protocols is essential, especially for such mission critical applications. Therefore, it is important to evaluate the worst case performance of protocols for successful completion of protocol evaluations. Figure 1 presents a scenario in which nodes 0 and 2 attempt to access channel to send data to nodes 1 and 3, respectively at the same time in a four node chain topology. Among destination nodes 1 and 3, node 1 can hear transmissions from both transmitters while node 3 can only hear transmissions from node 2. For this reason, transmission

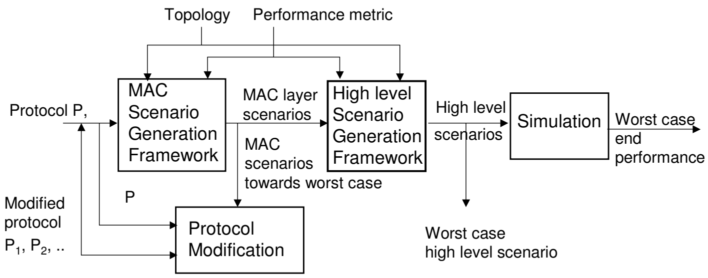


Fig. 2. An overview of our performance evaluation framework.

from node 0 fails (at receiver 1) and transmission from node 2 succeeds (at receiver 3). Because of binary exponential backoff, node 0 fails and node 2 succeeds in successive attempts to transmit data. This scenario is widely used and has been first identified in [Bensaou 2000] for analysis of (un)fairness of IEEE 802.11. Using our semi-automatic framework, we generate the same scenario and show that the scenario causes *zero throughput* for node 0 and maximum throughput for node 2, and thus exhibits the worst case (un)fairness both at the MAC level and the end application level for a longer time scale. Analysis of the above starvation scenario has motivated us to evaluate the worst case performance of wireless MAC protocols.

Figure 2 presents an overview of our complete performance evaluation framework. It has three main components: (1) MAC scenario generation framework, (2) high level scenario generation framework, and (3) protocol modifications for performance improvement. First, given a topology, a protocol and its performance metrics, the MAC scenario generation framework generates all MAC scenarios that exhibit poor MAC performance. Second, given all these MAC scenarios, the high level scenario generation framework generates high level scenarios that exhibit (near) the worst case end performance. These high level scenarios are simulated in a simulation environment to estimate the worst case performance of the given protocol. Our constructive semi-automatic scenario generation approach gives us insights that we use to modify the protocol in order to improve its worst case performance. Finally, these modified versions of the protocol are evaluated using our performance evaluation framework to demonstrate the effectiveness of all modifications.

The main focus of this paper is on our high level scenario generation framework. Given a performance metric, a network topology, and a set of MAC scenarios, we use this framework to generate high level scenarios that lead to the worst case performance for the given metric and for the given topology. We develop a finite state machine (FSM) model of the interface between the MAC layer and the higher layer. To increase the probability of successful transmission in CSMA/CA protocols, data at MAC layer is often fragmented into smaller data units known as *MAC data fragments*. We refer to MAC data fragment as *data fragment* in this paper. We define a sequence of MAC (control) messages that attempts to transmit a data fragment as a *MAC cycle*. A MAC cycle is defined as being *successful* if the MAC attempt to transmit the data fragment is successful. The cycle is defined as *unsuccessful* if the corresponding attempt is unsuccessful and requires a retransmission. A successful MAC cycle is followed by an attempt to transmit the next data fragment. We use this model to abstract MAC interface at each wireless node in a given network topology. We then use enumerations to generate sequences of MAC cycles. As we model the interface between higher layer and MAC, these sequences of MAC cycles represent the high level scenarios that appear at the MAC interface. Thus, given a MAC scenario at the interface, we use enumerations to generate *all* high level scenarios that can lead to the MAC scenario. In this step, we use heuristics to prioritize search alternatives and generate the high level scenarios that deem more likely to lead to the worst case end performance.

Our framework is applicable to all CSMA/CA based wireless MAC protocols. We have applied our methods to MACA [Karn 1990], MACAW [Bharghavan 1994] and IEEE 802.11 [WG 1997]. For each of the protocols, we use our MAC framework to generate a library of MAC scenarios and then use our high level framework to generate a library of high level scenarios from these MAC scenarios to invoke the (near) worst case end performance of the given metric. We only present the results of IEEE 802.11 protocol because of its widespread use and richer sets of semantics among other protocols in its class. Our analysis shows that IEEE 802.11 and MACAW show similar worst case performance in terms of throughput. However, the worst case throughput of MACA is much worse as there is no acknowledgment procedure in the protocol. IEEE 802.11 and MACA both use binary exponential backoff, and hence exhibit similar worst case (un)fairness.

The summary of simulation results for the worst case performance evaluation of IEEE 802.11 DCF protocol and its modification is as follows.

(1) The scenarios that we generate subsume the set of all scenarios used in related literature, namely, [Li 2001, Gupta 2004, Bensaou 2000, Pathma 2006, Xu 2002], for performance evaluation of the protocol.

(2) In all versions of the protocol, the worst case scenario generated by our performance evaluation framework shows the worst performance among all scenarios that we simulate. We simulate all protocol versions, several network load and network saturation conditions. The reductions in network throughput achieved by our worst case scenarios compared to the scenarios that are typically used (e.g., [Li 2001, Pathma 2006, Zhai 2006]) for performance evaluation of IEEE 802.11 DCF protocol are up to 34% and 36% in a network near and beyond the saturation point, respectively. The baseline scenarios have been used for evaluation of maximum throughput capacity [Li 2001] of the protocol at these saturation points.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 presents a brief overview of our MAC scenario generation framework. Section 4 presents details of our high level scenario generation framework. Sections 5 presents the results of the case study for IEEE 802.11 DCF protocol and demonstrates its effectiveness in evaluating the worst case performance of the protocol. Section 6 presents an analysis of our framework in evaluating (near) worst case performance. Section 7 concludes with brief outline of future research plans.

2. RELATED WORK

Many studies analyze the performance of IEEE 802.11 in ad hoc networks in general. To the best of our knowledge, no previous approach systematically generates scenarios that expose worst case performance - which is what we enable.

Bianchi [2000] analyzes the saturation throughput of IEEE 802.11 distributed coordinated function (DCF) in a single cell case using analytical model based on Markov chain. Bianchi's model is extended in [Gupta 2004] to multihop cases to analyze goodput of IEEE 802.11 DCF using simulation of arbitrary scenarios in ring and mesh topologies. Analytical models are used in [Abdrabou 2008] and [Shah 2005] to evaluate saturation throughput of IEEE 802.11 DCF in single clusters.

Ideal throughput capacity of a long chain of nodes in isolation is 1/4 of the total channel bandwidth [Li 2001]. However, the chain capacity that 802.11 MAC *achieves* with a greedy sender has been shown to be about 1/7 of the bandwidth in a near saturation condition, because of the fact that the nodes early in the chain starve the later nodes in a multihop case [Li 2001]. We refer to the $N-1$ hop scenario used in [Li 2001] as the *baseline scenario.* Our simulation results show that the reduction in network throughput achieved by our worst case scenario from the baseline scenario can be up to 34-36% in saturated neworks.

A number of studies in the literature analyze the fairness of 802.11 based ad hoc networks using different analytical methods to capture the definition of the fairness index. It has been shown in [Bensaou 2000] that the protocol exhibits a short-term unfairness in a chain of four nodes because of the fact that the binary exponential backoff (BEB) favors the last succeeding node. Our framework generates the same scenario as a worst case fairness scenario in a four node chain. We also identify long-term unfairness using our high level scenario generation framework. The unfairness of the protocol both in terms of short-term and long-term has been shown in previous work [Nandagopal 2000, Barrett 2002].

3. OVERVIEW OF OUR MAC SCENARIO GENERATOR

Inputs to the MAC scenario generation framework are: (1) the protocol model, (2) a network topology, and (3) a performance metric. The framework uses two main algorithms: (1) wanted_states generation, and (2) test scenario generation, to generate all MAC layer scenarios in which a performance penalty is achieved under the given metric for the given topology. In this paper, these MAC scenarios are used to generate high level scenarios.

Given a protocol performance metric, it first transforms the performance metric in a form that reflects our objective. For example, given MAC throughput, it transforms it into a set of *wanted_states* that is a set containing states that reduce throughput (i.e., a penalty of throughput) for a given topology. The wanted_states generation algorithm transforms the performance objective into wanted_states. Second, given a set of wanted_states and a topology, the test generation algorithm generates details of all MAC scenarios that lead to the performance penalty for the given protocol performance metric. As the high level scenarios are based on MAC scenarios, we present a brief overview of our MAC layer models and simple examples in Sections 3.2 and 3.3, respectively. Section 3.4 presents the limitations of this framework to be directly used for end performance evaluation. For details of the algorithms, see Chapter 2 of doctoral dissertation [Begum 2009a].

3.1 Assumptions

Our MAC layer model is based on the following assumptions. (1) We assume that wireless nodes do not fail or run out of battery power. Therefore, transmitters never stop in the middle of their transmissions. (2) We do not model message loss explicitly, however, loss due to collision and drop is implicitly modeled. (3) We assume the network to be static. We could have relaxed this assumption by allowing the neighborhood to change over time as the semantics of time allows the change of neighborhood at the granularity of message transmission (or reception). However, for typical automobile speeds for a mobile node, the movement during the period of a message transmission is not significant. (4) When a receiver is receiving transmissions from two nodes and is unable to receive the signal from either nodes, the phenomenon is known as *collision*. When the receiver is able to cleanly receive the signal from the closer transmitter, the phenomenon is called *capture*. Our finite state machines are deterministic and model only *collision*, not *capture*. (5) We only model virtual carrier sense mechanisms. (6) The carrier sensing range is twice the transmission range.

3.2 Overview of Our MAC Layer Models

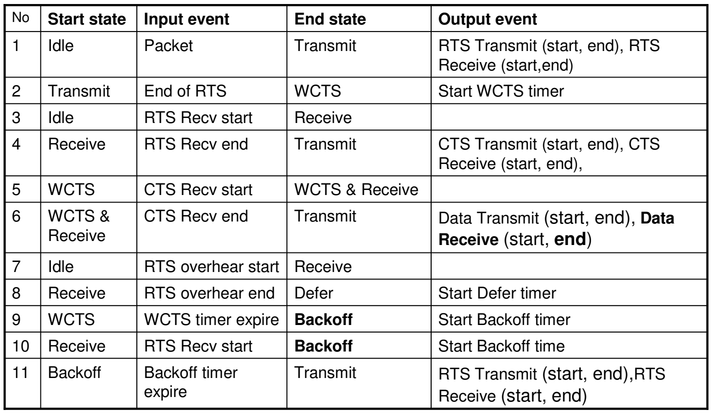
For purposes of illustration, we consider a simple 3-way handshake based CSMA/CA protocol P and a 4-node chain topology in which our objective is to generate MAC scenarios which exhibit poor performance in MAC throughput and fairness. MAC throughput is defined as the fraction of time the channel is used to successfully transmit payload bits [Bianchi 2000]. Let ρ be the amount of time a receiver spends in successfully receiving payload (data) and γ be the total time the transmitter spends since the first transmission opportunity to transmit the data, then we define *MAC throughput* as:

$$\text{MAC throughput} = \frac{\rho}{\gamma}. \tag{1}$$

In our model, ρ denotes the time it takes for an intended receiver to successfully receive MAC data packet. γ denotes the entire time a transmitter spends from the start of the first RTS (Request-To-Send) transmission to the end of successful reception of acknowledgment. Note that if an acknowledgment is lost requiring a retransmission, the time it takes for successful reception of retransmitted data and acknowledgment is also included in γ. A simple definition of *fairness* of a protocol is its ability to allocate the channel bandwidth equitably [Barret 2002]. We define a scenario to exhibit fairness if all nodes initiating handshakes get equal opportunity to transmit data. We define a scenario to exhibit *unfairness* if the handshake is unsuccessful for at least one of the nodes that initiate handshake in the scenario and one or more of the nodes are successful. Index of unfairness is defined as:

$$\text{Index of unfairness} = \frac{\text{Number of nodes unsuccessful in handshake}}{\text{Total number of nodes that initiate handshake}}. \tag{2}$$

Topology is modeled in terms of transmission range of each node in the network. Transmission range of node i is a set G_i containing the nodes who receive its transmission and can decode it. While its carrier sensing range is a set g_i containing



No	**Start state**	**Input event**	**End state**	**Output event**
1	Idle	Packet	Transmit	RTS Transmit (start, end), RTS Receive (start,end)
2	Transmit	End of RTS	WCTS	Start WCTS timer
3	Idle	RTS Recv start	Receive	
4	Receive	RTS Recv end	Transmit	CTS Transmit (start, end), CTS Receive (start, end),
5	WCTS	CTS Recv start	WCTS & Receive	
6	WCTS & Receive	CTS Recv end	Transmit	Data Transmit (start, end), **Data Receive** (start, **end**)
7	Idle	RTS overhear start	Receive	
8	Receive	RTS overhear end	Defer	Start Defer timer
9	WCTS	WCTS timer expire	**Backoff**	Start Backoff timer
10	Receive	RTS Recv start	**Backoff**	Start Backoff time
11	Backoff	Backoff timer expire	Transmit	RTS Transmit (start, end),RTS Receive (start, end)

Fig. 3. Transition table of a CSMA/CA protocol P.

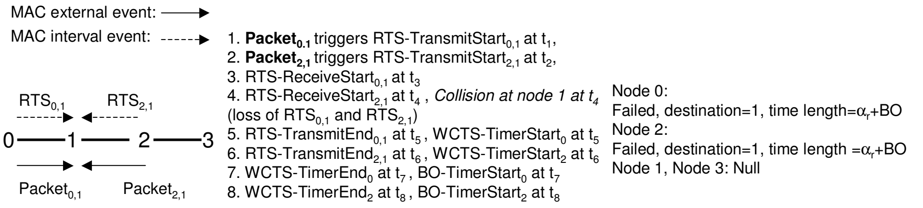
nodes who receive its transmission, but cannot decode it. Figure 1 presents a wireless network of 4 nodes where $G_0 = \{1\}$, $G_1 = \{0, 2\}$, $G_2 = \{1, 3\}$, and $G_3 = \{2\}$, and $g_0 = \{1, 2\}$, $g_1 = \{0, 2, 3\}$, $g_2 = \{0, 1, 3\}$, and $g_3 = \{1, 2\}$.

Figure 3 presents a *transition table* of protocol P representing a loss-less channel. Each row of the transition table represents a state transition of a node running the protocol. Each output event of a transition is associated with a variable representing the delay of the output event from the state transition. The protocol works as follows: upon receiving a *Packet* from higher layer for node j, node i transmits an *RTS* (Request-To-Send) destined for node j, schedules a *WCTS* (Wait-for-CTS) timer to wait for the *CTS* (Clear-To-Send) from j. Upon receiving the *RTS* destined for it, node j transmits a *CTS* (Clear-To-Send) to node i. On overhearing the *RTS* from i to j, a node k defers access to the channel by scheduling a timer. If node i receives the *CTS* from j before its *WCTS* timer expires, it goes to *Transmit* state to transmit the data, otherwise, it goes to *Backoff* (BO) state after which it retransmits the *RTS*. Row 1 of the table presented in Figure 3 states the following transition. When a node in *Idle* state receives a *Packet* from higher layer, it changes to *Transmit* state and triggers *RTS Transmit* and *RTS Receive* events. Among the events in Figure 3, the events triggered by an entity other than MAC layer are defined as *MAC external events* (e.g., *Packet*). The events triggered by MAC layer messages (e.g., RTS) are *MAC internal events* (e.g., RTS transmit start event).

A *MAC scenario* is composed of (1) network node states, (2) MAC events, and (3) time relations between the state transitions and MAC events. In our abstraction of MAC scenarios, a MAC layer event (e.g., RTS transmit start) represents an atomic action that causes state transitions of nodes in the network.

3.3 Examples of Generated MAC Scenarios

Using our MAC scenario generation framework, we automatically generate a set of MAC layer scenarios in a given topology. Let us input a four node chain topology (Figure 1), protocol P (Figure 3), and throughput as performance metric (Equation



1. **$Packet_{0,1}$** triggers RTS-TransmitStart$_{0,1}$ at t_1,
2. **$Packet_{2,1}$** triggers RTS-TransmitStart$_{2,1}$ at t_2,
3. RTS-ReceiveStart$_{0,1}$ at t_3
4. RTS-ReceiveStart$_{2,1}$ at t_4 , *Collision at node 1 at t_4* (loss of $RTS_{0,1}$ and $RTS_{2,1}$)
5. RTS-TransmitEnd$_{0,1}$ at t_5 , WCTS-TimerStart$_0$ at t_5
6. RTS-TransmitEnd$_{2,1}$ at t_6 , WCTS-TimerStart$_2$ at t_6
7. WCTS-TimerEnd$_0$ at t_7 , BO-TimerStart$_0$ at t_7
8. WCTS-TimerEnd$_2$ at t_8 , BO-TimerStart$_2$ at t_8

Node 0:
Failed, destination=1, time length=α_r+BO
Node 2:
Failed, destination=1, time length =α_r+BO
Node 1, Node 3: Null

(a) MAC scenario M_1. (b) Sequence of MAC events. (c) Summary of transmission attempts.

Fig. 4. A MAC scenario M_1 leading to a penalty of throughput.

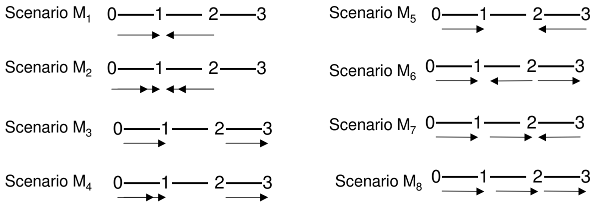


Fig. 5. A library of MAC scenarios generated in four node chain topology.

1) to our MAC scenario generator. Figure 4 presents a MAC scenario M_1 as generated automatically by our framework. Figure 4(a) shows a graphical representation of M_1. An arrow with a solid line represents a MAC external event ($Packet_{i,j}$) and that with a dashed line represents a MAC internal event. In this scenario, simultaneous transmissions of $RTS_{0,1}$ and $RTS_{2,1}$ cause a collision at node 1. Figure 4(b) presents a detailed sequence of MAC events that leads to the target collision at node 1. In this scenario, node 0 and node 2 each receives a MAC data fragment to transmit to node 1 as indicated by the external event $Packet_{0,1}$ and $Packet_{2,1}$, respectively. These $Packet$ events trigger RTS transmission start events at times t_1 and t_2, and RTS receive start events at t_3 and t_4, respectively as shown in the figure. The later receive start event leads to a collision at node 1 at time t_4. RTS transmission end events are triggered at t_5 and t_6 at which the transmitting nodes 0 and 2 start WCTS timers. These timers expire at t_7 and t_8 respectively, at which BO (Backoff) timers are started. Note that the MAC scenarios generated contain time relations of all events and state transitions.

Figure 5 presents the library of MAC scenarios automatically generated by our framework. In these scenarios, the arrows represent the direction of MAC external event. A scenario with double arrows represents a retransmission. MAC scenario M_1 represents two RTS events, one triggered from node 0 to node 1 and another from node 2 to node 3. Scenario M_2 represents two retransmitted RTS events, one triggered by node 0 and the other by node 2. Time relations of these events are extracted from the description of the scenario generated by our algorithms.

3.4 Limitations of MAC Scenarios

The strength of the framework is that it automatically generates many MAC scenarios. The scenarios show time relations of MAC events that we can use to take a step further. However, there are two major limitations of MAC scenarios. First, a MAC scenario represents a time scale which ranges from the order of micro-seconds to the order of mili-seconds. Our analysis of MAC scenarios show that a scenario showing the worst MAC performance does not show the (near) worst case end performance among the scenarios that we generate and simulate. Second, MAC scenarios only represent a network scenario within one hop neighborhood. These two limitations have motivated us to develop a framework that models the interface of MAC to its higher layer and generates high level scenarios systematically from the automatically generated MAC scenarios.

4. A FRAMEWORK FOR HIGH LEVEL SCENARIO GENERATION TO INVOKE THE NEAR WORST CASE END PERFORMANCE

The main objective of high level scenario generation is to evaluate the impact of a given MAC scenario at a larger time scale as well as in a larger network topology beyond one hop neighborhood. Figure 6 presents an overview of our high level scenario generation framework. The inputs to the high level scenario generation framework are: (1) the above MAC layer scenarios, (2) the network topology, and (3) the performance metric. The framework has three main components: (1) Identification of MAC cycles, (2) enumeration algorithms, and (3) comparison of high level scenarios. Given a MAC scenario, we first identify the constituent MAC cycles of the scenario. We also identify whether each of these MAC cycles was a success or a failure. These are input to the enumeration algorithms. We enumerate possible sequences of MAC cycles on the given topology. These sequences of MAC cycles represent high level scenarios and are the outputs of the framework. Finally, the high level scenarios are compared to determine the scenario that leads to the worst performance. In this step, we use heuristics, proposed by others or developed by us, to guide our search to generate scenarios that are deemed more likely to lead to the worst case. We refer to MAC cycles as MC in the rest of this paper.

4.1 Main Concepts for Generating High Level Scenarios

In a layered architecture, each layer represents an entity that provides some services to a layer above or beneath it. As we model MAC states of a network over a period of time using a MAC scenario, we take an approach that summarizes a MAC scenario into services the respective MAC entities provide to a higher layer. Let us consider the MAC scenario M_1 in Figure 4. In this scenario, simultaneous transmissions of $RTS_{0,1}$ and $RTS_{2,1}$ messages cause a collision that leads to a failure for both transmission attempts. We can summarize the MAC scenario from the viewpoint of an entity that receives a service from MAC layer, for example, an interface to higher layer, as presented in Figure 4(c), as follows: transmission attempts made by nodes 0 and 2, both destined for node 1, have failed. The time spent by each attempt

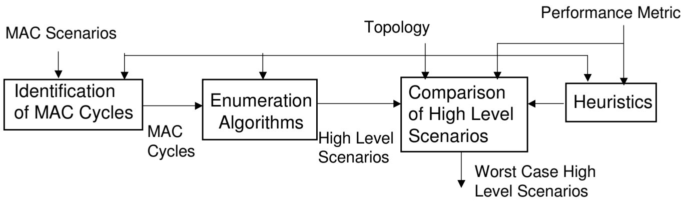


Fig. 6. An overview of our high level scenario generation framework.

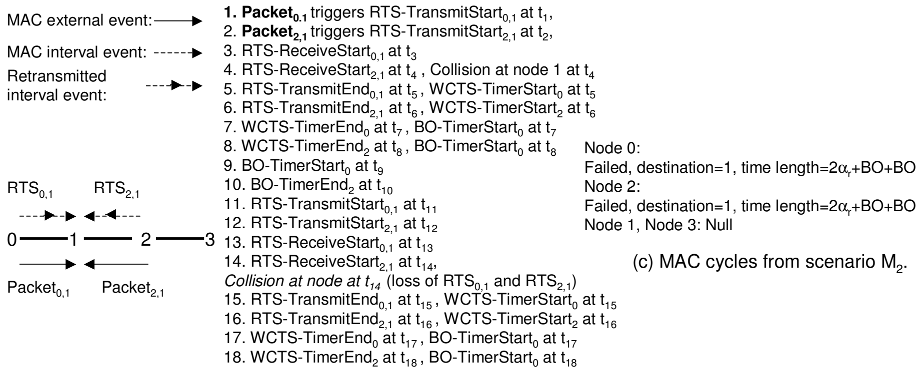
is given by the relation that is extracted from the MAC scenario. In this scenario the time lengths are given by expression $(\alpha_r + BO)$ such that α_r denotes the time to transmit an RTS and BO denotes the backoff interval. Finally, the summary describes that nodes 1 and 3 have not attempted to transmit in this scenario.

Given a network topology and a MAC scenario, we present the services provided by all nodes in the network as a summary of transmission attempts made by all nodes in the network. Let us denote each transmission attempt the MAC of a node makes as a *MAC cycle (MC)*. An MC is defined by: (1) a source, (2) a destination, (3) status, and (4) time length. The *source* of an MC represents the node that has attempted to transmit a data fragment at the MAC layer. *Destination* of the cycle represents the destination of data fragment. *Status* represents whether the transmission attempt made by the source is a success or a failure. Note that if a node does not attempt to transmit in a given MAC scenario, its status is represented as NULL. *Time length* of an MC represents the period of time for which the attempt has lasted. MAC scenario M_1 (see Figure 4(c)) consists of two MCs: failed cycles initiated by nodes 0 and 2.

As we generate MAC scenarios that exhibit poor performance (e.g., throughput is reduced), all our MAC scenarios are composed of at least one failed cycle initiated by at least one node. Figure 7 presents MAC scenario M_2 in which simultaneous transmissions of retransmitted $RTS_{0,1}$ and retransmitted $RTS_{2,1}$ cause a collision at node 1. These RTSs are retransmitted because of an earlier collision. Figure 7(b) shows the detailed sequence of MAC events extracted from the output of MAC generator. Note that RTS transmit start events at times t_{11} and t_{12} are triggered by expiration of backoff timers at nodes 0 and 2 respectively at times t_9 and t_{10} (i.e., the RTSs are retransmitted RTSs). Figure 7(c) presents the summary of transmission attempts. Time lengths of the failed MCs are given by expression $(2\alpha_r + BO + BO)$. Note that we can compare time lengths of two or more failed MCs in order to compare the throughput reductions achieved by the respective high level scenarios. For example, throughput reduction achieved by high level scenarios generated from MAC scenarios M_1 and M_2 can be compared using expressions in Figure 4(c) and 7(c).

4.2 Assumptions

We assume all flows to be continuous supplies of data at specific rates. Therefore, flows never stop during the period we consider for high level scenario generation.



1. **Packet$_{0.1}$** triggers RTS-TransmitStart$_{0,1}$ at t_1,
2. **Packet$_{2,1}$** triggers RTS-TransmitStart$_{2,1}$ at t_2,
3. RTS-ReceiveStart$_{0,1}$ at t_3
4. RTS-ReceiveStart$_{2,1}$ at t_4 , Collision at node 1 at t_4
5. RTS-TransmitEnd$_{0,1}$ at t_5 , WCTS-TimerStart$_0$ at t_5
6. RTS-TransmitEnd$_{2,1}$ at t_6 , WCTS-TimerStart$_2$ at t_6
7. WCTS-TimerEnd$_0$ at t_7 , BO-TimerStart$_0$ at t_7
8. WCTS-TimerEnd$_2$ at t_8 , BO-TimerStart$_0$ at t_8
9. BO-TimerStart$_0$ at t_9
10. BO-TimerEnd$_2$ at t_{10}
11. RTS-TransmitStart$_{0,1}$ at t_{11}
12. RTS-TransmitStart$_{2,1}$ at t_{12}
13. RTS-ReceiveStart$_{0,1}$ at t_{13}
14. RTS-ReceiveStart$_{2,1}$ at t_{14},
Collision at node at t_{14} (loss of RTS$_{0,1}$ and RTS$_{2,1}$)
15. RTS-TransmitEnd$_{0,1}$ at t_{15} , WCTS-TimerStart$_0$ at t_{15}
16. RTS-TransmitEnd$_{2,1}$ at t_{16} , WCTS-TimerStart$_2$ at t_{16}
17. WCTS-TimerEnd$_0$ at t_{17} , BO-TimerStart$_0$ at t_{17}
18. WCTS-TimerEnd$_2$ at t_{18} , BO-TimerStart$_0$ at t_{18}

Node 0:
Failed, destination=1, time length=$2\alpha_r$+BO+BO
Node 2:
Failed, destination=1, time length=$2\alpha_r$+BO+BO
Node 1, Node 3: Null

(a) MAC scenario M_2. (b) Sequence of MAC events. (c) MAC cycles from scenario M_2.

Fig. 7. MAC scenario M_2 and its constituent MAC cycles.

The parameters of a flow are: (1) source, (2) destination, (3) data rate, and (4) start time. Source and destination of a flow are determined by our scenario generation algorithms. The start time of a flow is determined by the sequences of MCs representing the high level scenario. We assume a flow to have a fixed rate during the period the sequences of MCs are generated. In other words, once started, a flow never changes the rate at which it generates data. In our scenario generation, we qualitatively denote two levels of data rates instead of assuming absolute data rates. When MC generation rate of a node is much lower compared to the rate at which the node can serve the MC, we denote the node as being *under-saturated*. When MC generation rate approaches the MC service rate, we denote the node as being *saturated*. As we assume all flows to have the same data rate in a scenario, the above definition can be generalized as *under-saturated network* and *saturated network*, respectively. Finally and most importantly, in our scenario generation, we assume nodes to be capable of generating at most one flow at a time. This assumption is necessary in order to maintain finiteness of the search space. Relaxing this assumption would lead to an infinite search space.

4.3 Basic Models for High Level Scenario Generation

4.3.1 *MAC cycle as a model of events.* MAC cycle is the atomic event in our high level abstraction of scenarios. We denote a MAC cycle using notation $MC_{x,y}$ that represents a transmission attempt initiated by node x destined for node y. An MC is specified by source, destination, status and time length.

4.3.2 *Model of state of a wireless node.* We model the state transition of a wireless node using a non-deterministic state transition diagram. Let us denote $Data^f$ as the f^{th} data fragment at the MAC layer. Figure 8 presents the state transition diagram of a wireless node i. The external event of the transition diagram

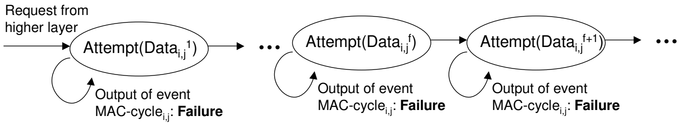


Fig. 8. State transition diagram of a wireless node i's MAC interface.

is a request from higher layer to transmit a number of data fragments to some destination node j. Let us denote the state of node i in which it attempts to transmit data fragment $Data_{i,j}^{f}$ to node j as $Attempt(Data_{i,j}^{f})$. In order to transmit the data fragment $Data_{i,j}^{f}$, the node transits its state to $Attempt(Data_{i,j}^{f})$ in which it triggers the event $MAC\text{-}cycle_{i,j}$. If the event $MAC\text{-}cycle_{i,j}$ is a failure, it goes back to state $Attempt(Data_{i,j}^{f})$. Otherwise, the node changes its state to $Attempt(Data_{i,j}^{f+1})$ in which it makes an attempt using MC to deliver fragment $Data_{i,j}^{f+1}$.

4.3.3 *Model of a high level scenario.* A high level scenario is defined by a set of high level flow descriptions. Let $Flow_{x,y}$ denote a high level flow from node x to node y. Given a high level scenario with more than one flows, the time relations between these flows are also defined in the scenario. A flow represents a continuous supply of data from a source node to a destination node over a period of time. For that period of time, this supply of data from node x to node y can be represented using a sequence of MCs from node x to node y. As we consider a general model of high level scenario, flow from node x can be destined for any node in the network. Note that MCs are single hop (as they are based on MAC scenarios). Algorithms to generate multihop MCs are presented in Section 4.4.

4.3.4 *Performance metrices.* Given a sequence of MCs, we define delivery ratio of $\text{Flow}_{i,j}$ ($\text{dR}_{i,j}$) and delivery ratio obtained by a scenario (dR_s) using Equations 3 and 4, respectively.

$$\text{dR}_{i,j} = \frac{\text{Total number of successful } \text{MC}_{i,j}\text{s}}{\text{Total number of } \text{MC}_{i,j}\text{s with end of } \text{Data}_{i,j} \text{ transmission}}. \tag{3}$$

$$\text{dR}_s = \frac{\sum_{i,j} \text{Total number of successful } \text{MC}_{i,j}\text{s}}{\sum_{i,j} \text{Total number of } \text{MC}_{i,j}\text{s with end of } \text{Data}_{i,j} \text{ trans.}}. \tag{4}$$

Note that $MC_{i,j}$ is successful when the transmission attempt to transmit data fragment from node i to node j does not require a retransmission. Given a sequence of MCs, we define end throughput (eTh) achieved by the high level scenario using Equation 5. In this equation, $\text{Data}_{i,j}^{f}$ denotes data fragment f from node i to node j.

$$\text{eTh} = \frac{\sum_j \sum_f \text{Total time recvr. } j \text{ spends to recv. } \text{Data}_{i,j}^{f} \text{ succ.}}{\sum_i \sum_f \text{Total time spent by txr. } i \text{ to succ. trans. } Data_{i,j}^{f} \text{ since 1st } \text{RTS}_{i,j}} \tag{5}$$

Note that throughput equation considers all flows (subscripts i and j of $\sum_i$ and $\sum_j$, respectively) and all data fragments (subscript f of $\sum_f$) in a given sequence of MCs. For a specific data fragment in a given flow, the numerator considers reception of Data only once in the case when it is retransmitted. However, the denominator considers the entire time the transmitter spends since the first attempt to access the channel to transmit the fragment till the end of successful ACK (Acknowledgment) reception for that fragment. In other words, it considers all failed MCs that may precede a successful MC.

Given a sequence of MCs, we define the index of unfairness ($\mathrm{iUF}_{i,j}$) between two receivers i and j using Equation 6.

$$\mathrm{iUF}_{i,j} = \frac{\sum_x \text{Total time rcvr. } i \text{ spends in succ. recvng. } \mathrm{Data}_{x,i}}{\sum_y \text{Total time rcvr. } j \text{ spends in succ. recvng. } \mathrm{Data}_{y,j}}. \qquad (6)$$

Note that receivers i and j may receive data from multiple sources (subscripts x and y of $\sum_x$ and $\sum_y$, respectively). A pair of flows exhibits perfect fairness if the receivers of these flows spend the same amount of time in receiving data successfully from the respective transmitters. Therefore, an index of 1 represents a perfectly fair scenario. We consider index of unfairness between all pairs of flows in a given sequence of MCs. We define *a scenario to exhibit the worst case unfairness* if there exists a pair of flows in the scenario that gives the worst (minimum) index of unfairness among all pairs of flows of all scenarios that we generate.

4.4 High Level Scenario Generation Algorithm

Given a MAC scenario, our high level scenario generation algorithm (see Figure 6) first identifies its constituent MCs. It then determines if there is a dependency between MCs generated by different nodes. In a given network topology, a node that can independently generate MAC cycles (MCs) injects flows to some destination nodes in the topology. A node that depends on other nodes in generating MCs forwards flows towards the final destination of multihop flows. Node y is defined as being *dependent on node x in generating MCs* if node y forwards data it receives from x to a node z. Therefore, given a MAC scenario, we can generate $Flow_{x,z}$ as a part of high level scenario if MAC cycles $MC_{x,y}$ and $MC_{y,z}$ exist in the MAC scenario. The algorithm enumerates all possible dependencies and generates all sequences of MCs, each representing a high level scenario. Finally, the high level scenarios are compared (time length of all MCs in a scenario) to determine the scenario leading to the worst performance. In this step, we also determine the high level scenario that leads to the best case end performance. The three main procedural components are as follows. For details, see Chapter 3 of the doctoral dissertation [Begum 2009a].

4.4.1 *Identification of MAC cycles.* Given a MAC scenario and a node x, existence of MAC external event triggered by x (e.g., $\mathrm{Packet}_{x,y}$ to any other node y), identifies a MAC cycle. If there is no such event triggered by x, the procedure returns a NULL for node x. It also returns the status as *success* if the corresponding end of handshake (e.g., successful reception of $\mathrm{ACK}_{y,x}$) exists in the MAC scenario. Otherwise, it returns the status as *failed.*

4.4.2 *Enumeration of sequences of MAC cycles.* Given initial MCs of all nodes, the algorithm enumerates possible sequences of MCs using following rules.

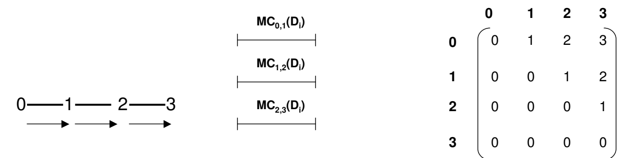
(1) A node can generate an MC independent of other nodes in the topology. This allows us to generate a high level scenario which results from a direct mapping of the corresponding MAC scenario. For example, if there exists a MAC layer RTS event from node x to node y, we generate a high level flow from node x to node y. Such an intuitive notion of mapping MAC scenarios to high level scenarios has been adopted in related works (e.g., [Gupta 2004, Bensaou 2000, Nandagopal 2000, Pathma 2006, Xu 2002]) which have analyzed behavior of IEEE 802.11 at MAC layer and simulated using high level scenarios.

(2) Two nodes x and y can generate dependent MCs such that generation of an MC of node y depends on generation of a successful MC by node x, for example, if the initial MC of node x has node y as a destination. This rule allows us to generate scenarios in which the high level scenarios have flows that travel two hops. We extend this rule to hop = 3 cases as follows. Three nodes x, y and z can generate dependent MCs such that generation of an MC of node y depends on successful MC generation by node x if the initial MC of node x has node y as destination and generation of MC of node z depends on successful MC generation by node y if the initial MC of node y has node z as destination. This rule is extended to as many hops as possible for given topology and the initial MCs. Note that such a dependency of MAC cycles allows us to generate multihop high level scenarios from MAC layer scenarios.

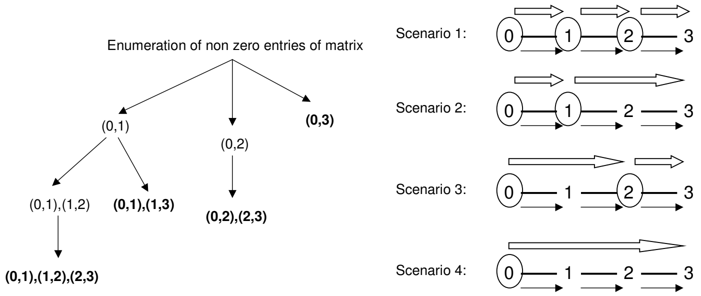
Enumeration algorithm uses procedure *construct-dependency-matrix* that builds a matrix according to the dependency of MC generation among various nodes of the topology. Let us denote $Dep_{i,j}$ as j^{th} column of i^{th} row of the dependency matrix. If there exists no dependency between nodes i and j, $Dep_{i,j}$ is set to 0. If there exist an m-hop dependency between nodes i and j, $Dep_{i,j}$ is set to m. The input MAC cycles, the network topology and the dependency matrix are input to procedure *enumerate-sequences* that enumerates all possible sequences of MCs. The procedure first generates scenarios which are the results of direct mapping. It then generates all multihop scenarios for each non-zero entries of the matrix. It finally generates multihop scenarios by taking the cross product between all possible combinations of non-zero entries of the matrix.

4.4.3 *Comparison of high level scenarios.* Finally all high level scenarios are compared to evaluate the scenario leading to the worst case end performance. In this step, we use topology to determine if a given sequence of MC is successful or not, depending on the number of simultaneous attempts of MAC cycle in a neighborhood of nodes.

4.4.4 *Examples of generated high level scenarios.* Figure 9(a) and 9(b) respectively present a MAC scenario (M_8) and its initial MCs. Note that each node is capable of generating MCs (except the rightmost node). Figure 9(c) presents the corresponding dependency matrix. A non-zero entry of the matrix at row i and column j represents a dependency between the nodes i and j in the initial MCs. For example, node 0 is capable of one, two and three hop dependency in generating MCs destined for nodes 1, 2, and 3, respectively. Figure 9(d) shows the possible



(a) MAC scenario M_8. (b) Initial MAC cycles. (c) 4x4 Dependency matrix.



(d) Enumeration of high level scenarios. (e) Four high level scenarios.

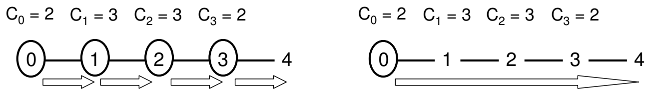
Fig. 9. High level scenario generation from MAC scenario M_8.

assignments of sources (independent nodes) and destinations of flows enumerated from the matrix as follows: a) (0,1),(1,2),(2,3), b) (0,1),(1,3), c) (0,2),(2,3), and d) (0,3). Figure 9(e) presents these four high level scenarios. In *Scenario 1*, each node inserts a flow in the network independently. *Scenario 2* and *Scenario 3* each has two flows: $Flow_{0,1}$ and $Flow_{1,3}$, and $Flow_{0,2}$ and $Flow_{2,3}$, respectively. *Scenario 4* is an $N-1$ hop scenario in which flows from one end is forwarded by the intermediate nodes to the other end of the chain.

4.5 Evaluating the Extreme Performance

Our objective is to generate the high level scenario that leads to the worst case end performance. The number of MAC scenarios that lead to poor MAC performance is quite large. For a given topology, the numbers of dependencies are also large resulting in a large number of high level scenarios from a given MAC scenario. For this reason, we use heuristics to prioritize our search and choose a scenario that deems more likely to lead to the worst case. The heuristics (Sections 4.5.4 and 4.5.5) are based on the properties of MAC cycles (Section 4.5.1), network topology (Section 4.5.2) and transmission capabilities of nodes (Section 4.5.3). We compare scenarios qualitatively and identify the scenario showing the extreme performance.

We consider two types of networks based on network traffic: (1) fully loaded network, and (2) half-loaded network. A network is fully loaded when all nodes have data to transmit. A network is half-loaded when half of all nodes have data to transmit. For each of these networks, we consider three network saturation conditions: (1) under-saturated network, (2) saturated network, and (3) a network beyond saturation. We present heuristics for fully loaded network in this paper. For half loaded network, see Chapter 3 of dissertation [Begum 2009a]. We assume



(a) S_1: Single hop scenario. (b) S_2: N-1 hop scenario.

Fig. 10. Single hop and N-1 hop scenarios in fully loaded chain topologies.

the *time* for which sequences of MCs are generated to be the same for all scenarios we generate for a given topology and a performance metric.

4.5.1 *Properties of MAC cycles.* (a) Given a failed MC, the longer the time length, the higher the reduction of end throughput by the MC. (b) Given a sequence of MCs, total time period of all failed MCs represents the total reduction of throughput by the high level scenario. (c) Given a sequence of MCs, the higher the number of failed MCs, the lower the number of successful MCs. (d) Given a sequence of MCs, the higher the number of total MCs, the higher the number of transmission attempts.

4.5.2 *Properties of network topology.* (a) Let us denote C_x as the number of competitors of a node x, i.e., the number of nodes with which it must contend for channel access in a given network topology. Let us take a five node chain topology shown in Figure 10. In this topology, C_0, C_1, C_2 and C_3 denote the numbers of competitors of nodes 0, 1, 2 and 3. In this topology, $C_0 = 2$ (nodes 1 and 2), $C_1 = 3$ (nodes 0, 2, 3), $C_2 = 3$ and $C_3 = 2$. (b) Simultaneous access of channel by two nodes that are within a hop distance of two or less, leads a failure for one of the attempts. From MC property, such a failure leads to a backoff for the node that fails. (c) Simultaneous access of channel by nodes that are within a hop distance of three or more may lead to success for both of the attempts. From MC property, such successful attempts may lead to failures for ongoing transmissions causing backoff and wasted bandwidth. If such a failure occurs, the throughput reduction is higher compared to the reduction caused by simultaneous transmission attempts by nodes within a hop distance of 2. In this paper, we refer to the backoff state of a node in which the node has one or more data fragments in its queue as *backlogged.*

4.5.3 *Steady state transmission capabilities.* Note that an *independent* node can generate MCs on its own as a source of a flow. A *dependent* node depends on other nodes to provide data fragments for generating MCs. Let us consider two scenarios with the minimum and the maximum number of independent nodes possible in a given topology. Figure 10 presents these scenarios. In this figure, a circled node denotes an independent node, i.e., a source of a flow. A node without a circle represents a dependent node, i.e., a forwarding node towards the destination of a flow. Figure 10(a) presents single hop scenario S_1 with N-1 independent nodes. Figure 10(b) presents (N-1) hop scenario S_2 with only one independent node.

Let us assume in S_1, four flows $Flow_{0,1}$, $Flow_{1,2}$, $Flow_{2,3}$, and $Flow_{3,4}$ start at the same time. Note that the numbers of competitors of nodes 0 and 3 are 2, and those of nodes 1 and 2 are 3. Therefore, initially in S_1, nodes 0 and 3 are more likely to win the contention and start to transmit data. Node 2 is likely to be the third node to access channel to transmit its data as node 1, when it gets access, transmits ACK for the data it receives from node 0. Therefore, in steady state, nodes 1 and 2 have backlog of data. Moreover, the backlog is higher at node 1 compared to node 2. The same sequences of MCs are observed in scenario S_2 in steady state. However, from this state of backlog, scenarios S_1 and S_2 take two different paths. Note that binary exponential backoff algorithms are biased against the node that looses a contention [Bensaou 2000]. Accordingly, from the backlog condition, node 2 gets access to the channel before node 1. In S_2, backlog of node 2 is cleared at some point as node 2 cannot generate the MC on its own. Instead, node 2 waits for node 1 (which is backlogged and cannot transmit when 2 is transmitting), giving node 1 the access to the channel to transmit its data. In other words, dependency among nodes in multihop chain topologies helps clear backlogs of data that occur at various stages of a scenario. In S_1, however, node 2 always has data to transmit as it is an independent source of flow. Therefore, backlog of node 1 is not cleared in the steady state. Thus in steady state, transmission capabilities of independent nodes depend on the rate at which the flow is generating data. Transmission capability of a dependent node is restricted by transmission capability of its upstream node in steady state. Therefore in steady state, transmission capabilities of independent nodes are higher than the transmission capabilities of dependent nodes.

4.5.4 *Heuristic 1 for delivery ratio.* We obtain delivery ratio using Equation 4 given a sequence of MCs. Given a set of high level scenarios, the scenario with maximum transmissions of data as well as maximum loss of data must show the worst case delivery ratio. On the other hand, the scenario with minimum transmissions of data as well as the minimum loss must show the best case delivery ratio. We identify the scenarios that we expect to achieve the maximum and the minimum values of denominator of Equation 4 and heuristically predict these scenarios respectively as the worst case and the best case. Then we argue that among these two scenarios, the numerator also shows the criteria needed for the extreme performance.

As the transmission capability of an independent node is higher than the transmission capability of a dependent node, the scenario with maximum number (i.e., $N-1$) of independent nodes is expected to achieve the maximum value of denominator of Equation 4. Accordingly, we expect a single hop scenario (e.g., S_1 in Figure 10(a)) to exhibit the worst case delivery ratio. Similarly, an $N-1$ hop scenario (e.g., S_2 in Figure 10(b)) is expected to exhibit the best case delivery ratio. Now, let us examine the numerator of Equation 4 to compare the number of losses these scenarios achieve. Note that the higher the number of failed MCs, the lower the value of the numerator reducing delivery ratio. In S_1, the number of simultaneous attempts is expected to be higher in steady state as all nodes have their own data to transmit. In S_2, dependency of nodes reduces the number of simultaneous attempts in steady state. The other components of losses, for example, losses due to interference, are expected to be similar in steady state for the two scenario. Therefore, the number of failed MCs are higher in S_1 resulting in a lower value of numerator of Equation 4. Therefore, *a single hop scenario is likely to achieve the worst case delivery ratio. An N-1 hop scenario is likely to achieve the best case delivery ratio.*

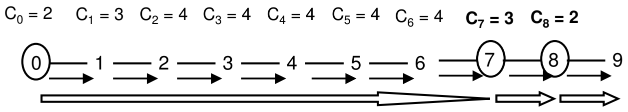


Fig. 11. Placement of independent nodes in a ten node chain for worst network throughput.

4.5.5 *Heuristic 4 for network throughput.* Figure 11 presents a MAC scenario in a 10-node chain topology that leads to a number of high level scenarios based on dependency of node. In this figure, C_0, C_1,.., C_8 are the numbers of competitors of the respective node in steady state when all nodes. Our objective is to identify the worst case scenario in saturated network. Note that in steady state, all nodes are expected to have their queues full. We have shown that different scenarios evolve differently from the steady state based on dependency of nodes in the scenario and eventually lead to different throughput values [Begum 2009a].

We formulate the problem of identifying the worst case scenario as a problem of assigning dependency of nodes that we expect to show the worst case throughput. We must place the first source at node 0. Note that hop = 3 scenario is expected to achieve the worst case throughput compared to hop = 1 or hop = 2 scenarios [Begum 2009a]. Therefore, the next source can be placed at any node starting from node 3. Thus, there are six choices of next sources. These choices are nodes 3, 4, 5, 6, 7 and 8. Now, we must choose the next independent node such that it takes the longest time to clear the backlogs created at various nodes in this chain topology. Note that in a $N-1$ hop scenario, backlogs are cleared earlier because of dependency. It also shows to exhibit a monotonically decreasing transmission capability along the chain. Let us assign node 7 as the next independent source as it has the minimum competitors among above choices. Selecting node 7 as an independent source allows node 3 or node 4 to transmit simultaneously in steady state. Therefore, backlogs at succeeding nodes (nodes 5 and 6, for example) are higher compared to backlogs at node 3 or node 4. Similarly, backlogs at node 1 and 2 are higher compared to node 3 or node 4. Such an asymmetry in backlogs, especially higher backlogs in downstream nodes compared to the upstream nodes in multihop chain causes the scenario to take longer to clear it backlogs. On the other hand, an independent source at the edge of the chain with minimum number of competitors causes the scenario to arrive at backlog periodically. Therefore, *(N-3) hop scenarios are expected to achieve the worst case throughput among the scenarios in which independent sources are non uniformly distributed over the chain topology.*

5. CASE STUDY: PERFORMANCE OF IEEE 802.11 DCF PROTOCOL

IEEE 802.11 [WG 1997] is based on single channel multiple access schemes. The channel access is based on both physical and virtual carrier sense. We consider the protocol mechanisms in DCF which is designed for ad hoc networks and is based on CSMA/CA. Each node maintains a *Network Allocation Vector* (NAV) timer to monitor the channel status. When node *0* of Topology I wants to transmit data to node *1*, it senses the channel status. If the channel is idle for a predefined period

Table I. High level scenarios in a fully loaded seven node chain.

Scenario Id	Flow description
Sys 10	(0,6)
Sys 19	(0,5),(5,6)
Sys 21	(0,4),(4,6)
Sys 20	(0,4),(4,5),(5,6)
Sys 13	(0,3),3,6)
Sys 22	(0,3),(3,5),(5,6)
Sys 23	(0,3),(3,4),(4,6)
Sys 24	(0,3),(3,4),(4,5),(5,6)
Sys 25	(0,2),(2,6)
Sys 26	(0,2),(2,5),(5,6)
Sys 27	(0,2),(2,3),(3,6)
Sys 11	(0,2),(2,4), (4,6)
Sys 28	(0,2),(2,4),(4,5),(5,6)
Sys 29	(0,2),(2,3),(3,4),(4,6)
Sys 30	(0,2),(2,3),(3,4),(4,5),(5,6)
Sys 31	(0,1),(1,6)
Sys 32	(0,1),(1,5),(5,6)
Sys 33	(0,1),(1,4),(4,6)
Sys 34	(0,1),(1,3),(3,6)
Sys 8	(0,1),(1,2),(2,3),(3,4),(4,5),(5,6)

(DIFS), it sends RTS to *1* indicating the period T for which it expects to reserve the channel if RTS/CTS is successful. Node *0* schedules a WCTS timer to limit the period of time to wait for CTS. When node *1* receives RTS, if its NAV timer is not running, it replies with a CTS packet in the next frame slot. Otherwise, it discards the RTS packet. While sending CTS, node *1* schedules a WData (Wait-for-Data) timer to limit the period of time to wait for data. All other nodes in the range of transmitter and receiver update their NAV with the value given in the RTS/CTS packets. During the period when a node has NAV running, it defers access to the channel. When node *0* receives CTS before its WCTS timer expires, it transmits data while scheduling WACK (Wait-for-ACK) timer to wait for ACK. If ACK from node *1* does not arrive within this period, node *0* backs off and re-contends. Upon reception of data from node *0*, node *1* sends an ACK back to node *0*.

5.1 High Level Scenarios towards Extreme Performance

Table I presents a set of all high level scenarios generated by our framework in a fully loaded seven node chain topology. The column titled *Scenario Id* represents the unique Id of a scenario. The column titled *Flow description* represents the description of flows in the scenario. For example, scenario *Sys 10* contains one flow: a flow from node 0 to node 6. Scenario *Sys 8* contains six flows: each node sends data to its right side neighbor in the chain. The summary of results using our framework is as follows. (1) *Sys 8* and *Sys 10* are expected to show the worst and the best case delivery ratio, respectively. (2) *Sys 20* and *Sys 10* are expected to show the worst case network throughput in fully loaded chains in saturated and under-saturated conditions, respectively. (3) In all network conditions, *Sys 8* is expected

Table II. Simulation parameters in ns-2 simulation.

Parameter	**Setting**
Transmission range	250 meters
Carrier sensing range	550 meters
Node distance	250 meters
Routing	Static routing (NOAH)
IEEE 802.11 data rate	2 Mbps
CBR packet size	1500 bytes
CBR packet intervals	0.057, 0.029, 0.012, 0.006, 0.005 sec.

to achieve the best network throughput among all scenarios in fully loaded network as the scenario contains the highest number of sources.

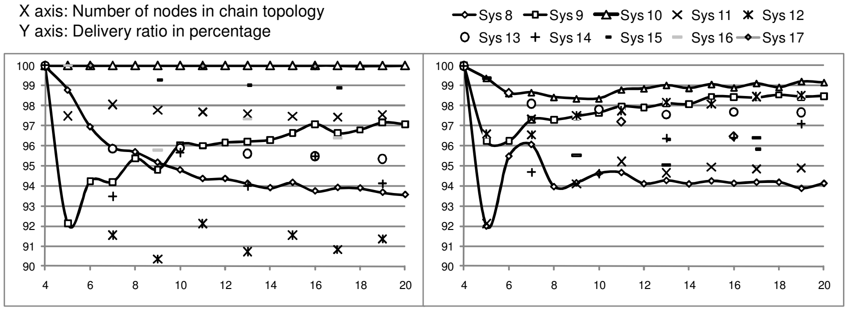
5.2 Simulation Results

We simulate our high level scenarios using ns-2 simulation [Breslau 2000] in order to evaluate the end performance achieved by these scenarios. Table II presents simulation parameters used in our experiments. To minimize the effect of routing we use static routing in all our simulations. We use the protocol in its 2 Mbps data rate mode. We use CBR data rates of 0.21 Mbps, 0.42 Mbps, 1 Mbps, 2 Mbps and 2.4 Mbps to respectively represent network conditions below, at and beyond saturation point. Note that the network is at the saturation point when CBR data rate is 2 Mbps. We run simulations for throughput and fairness experiments for 50 seconds and 300 seconds, respectively. In this section, we report simulation results of delivery ratio and network throughput for fully loaded chain topologies. For others, see Chapter 3 of dissertation [Begum 2009a].

5.2.1 *Delivery ratio.* Figure 12(a), 12(b) and 12(c) present delivery ratio of scenarios in fully loaded chain topologies at CBR rates of 0.2, 2 and 2.4 Mbps which represent networks before, near and beyond the saturation point, respectively. Simulation results of other data rates are consistent with these three cases (see [Begum 2009a]). X-axis of these graphs represents the number of nodes in chain. Y-axis of these graphs represent the delivery ratio in percentage. Note that at all data rates, *Sys 10* ($N-1$ hop scenario) achieves the maximum delivery ratio in most cases. *Sys 8* (single hop scenario) achieves the minimum delivery ratio in most cases where data rate is near and beyond saturation point.

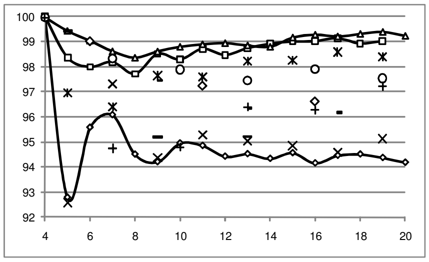
We have performed detailed packet level analysis of these scenarios based on simulation results to validate the properties we identified in Section 4.5. The analysis of packet level statistics over period of five seconds and 50 seconds simulations match our properties. See Chapter 3 of the dissertation [Begum 2009a] for the detailed discussion. In summary, backlogs at node 1 in steady state causes the single hop scenario to cause higher losses, reducing numerator of delivery ratio equation (Equation 4). Independence of single hop scenario increases the total transmissions increasing the denominator of the equation. In $N-1$ hop scenario, dependency of nodes reduces the number of transmissions reducing the denominator.

5.2.2 *Network throughput.* Table III presents a summary of simulated network throughput we achieve in a 13 node chain topology. In this topology, we simulate



(a) CBR rate = 0.2 Mbps.

(b) CBR rate = 2.0 Mbps.



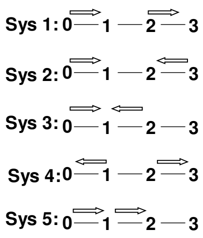
(c) CBR rate = 2.4 Mbps.

Fig. 12. Delivery ratio of scenarios in fully loaded chain topologies.

Table III. Summary of simulated network throughput in 13 node chain.

	CBR1Mbps	**CBR2Mbps**	**CBR2.4Mbps**
Baseline scen Id	Sys 10	Sys 10	Sys 10
Baseline net thpt (Kbps)	2367.12	2515.68	2521.44
Worst thpt scen Id	Sys 20	Sys 20	Sys 20
Worst net thpt (Kbps)	1735.28	1765.66	1786.16
Best thpt scen Id	Sys 8	Sys 8	Sys 8
Best net thpt (Kbps)	4493.31	4660.42	4744.4
%Reduction w.r.t. Baseline	26.7	29.82	29.17
%Reduction w.r.t. Best	61.39	62.12	62.36

32 scenarios at five CBR data rates. For each data rate, we calculate the worst and the best case throughput. Scenario *Sys 10* represents our baseline scenario, in which flow from one end of a chain is sent to the other end of the chain. This scenario has been used for performance evaluation of IEEE 802.11 in multihop ad hoc networks (e.g., in [Li 2001, Zhai 2006, Pathma 2006]). These works have used this scenario to evaluate the maximum throughput capacity of chain topologies, especially when the network operates in saturation region. Table III presents a summary of simulation results at three CBR rates representing a network near and beyond saturation for a fair comparison with related work. Note that scenarios *Sys 20* and *Sys 8* respectively show the worst case and the best case network throughputs for these data rates. The last two rows represent percentage reduction in network through-



Data rate (Mbps)	**Sys 1**	**Sys 2**	**Sys 3**	**Sys 4**	**Sys 5**
0.2	0.61	1	1	1	1
0.42	0.61	1	1	1	1
1	**0.23**	0.92	0.95	1	0.95
2	**0.07**	0.93	0.9	0.97	0.99
2.4	**0.06**	0.96	0.97	0.99	0.96

(a) High level scenarios. (b) Index of unfairness.

Fig. 13. High level scenarios in a half loaded seven node chain.

put achieved by the worst case scenario (*Sys 20*) compared to the baseline scenario (*Sys 10*) and the best case scenario (*Sys 8*), respectively. Note that our worst case scenario achieves a reduction by up to 30% compared to the baseline scenario. Our worst case scenario achieves up to 62% reduction in throughput compared to the best case scenario. Simulation results also show that the higher the size of the chain, the higher the reduction in throughput by our worst case scenarios compared to the baseline scenario *Sys 10* and the best case scenario.

These results demonstrate that we are able to generate scenarios which show significant reductions in throughput compared to the scenarios typically used for performance evaluation approaches. Our scenario generation algorithm is able to semi-automatically generate a library of scenarios including all the ones used for evaluation in related work. Wireless ad hoc networks are used for emergency response and mission critical applications in which all wireless nodes may be deployed with similar capabilities to achieve the best performance of the network. In such cases, scenarios like *Sys 10* ($N-1$ hop scenario), *Sys 20* ($N-3$ hop scenario) and *Sys 8* (single hop scenario) are equally likely to occur. We show that *Sys 20* achieves reductions of 30% and 62% compared to *Sys 10* and *Sys 8*, respectively.

5.2.3 *Fairness.* In half loaded network topologies, we generate scenarios in which a performance penalty is achieved with respect to per node fairness. Figure 13.(a) presents a set of high level scenarios for a half loaded four node chain topology. *Sys 1*, one of our scenarios is reported as the scenario leading to the worst case (un)fairness. Figure 13.(b) presents index of unfairness achieved by scenarios *Sys 1* thru *Sys 5* at various data rates. An index of 1 represents a perfectly fair scenario. A lower value of the index indicates more unfairness of the scenario towards the flow receiving the minimum throughput. In all data rates, *Sys 1* shows the most unfairness. The unfairness of *Sys 1* increases with the data rate. *Sys 1* has been used for analysis of unfairness of the protocol in [Bensaou 2000]. Our results augment the previous work by showing that the scenario exhibits the worst case (un)fairness in chain topologies among scenarios with similar load. We simulate scenario *Sys 1* for longer chains ($N = 6,..., 20$) to evaluate the extent to which unfairness of the protocol exists. Simulation results show that the unfairness towards a single flow degrades as the chain gets longer, however, the unfairness on average improves.

6. ANALYSIS OF THE FRAMEWORK

Given a set of MAC scenarios, a network topology and a protocol performance metric, we generate high level scenarios. Therefore, the completeness of our high level scenario generation algorithms depends on the completeness of our MAC scenario generation algorithms. We have previously proved the completeness of our MAC scenarios generation algorithms in [Begum 2009]. We present these proofs along with our basic algorithms in our online supplementary document. In the document, Section 2 presents an overview of our MAC scenario generation algorithms. Sections 3 and 4 respectively present our search and implication algorithms and proofs. Section 5 of the document completes the proof of completeness (MAC).

Our high level scenario generation algorithm assumes a node to be capable of generating at most one flow at a time. Under this assumption, we use high level scenario generation algorithms to generate high level scenarios. We use heuristics to prioritize search alternatives in order to generate the worst case scenario within realistic run time. In Section 6.1, we show that our algorithms are complete without and with the heuristics. Section 6.2 presents an analysis of complexity of our MAC and high level scenario generation algorithms. Section 6.3 discusses the performance guarantee provided by our framework.

6.1 Completeness of High Level Scenario Generation Algorithms

Given a MAC scenario, our high level scenario generation algorithm uses two main procedures: (1) *construct-dependency-matrix* and (2) *enumerate-sequences*. See Appendix III.B and Appendix III.C of dissertation [Begum 2009a] for details of these procedures, respectively. For a given network topology of N nodes, the minimum and the maximum number of possible hops are 1 and $N-1$, respectively. Therefore, generating all dependencies inclusively from 1 to $N-1$ ensures that we generate all possible dependencies in the given N node topology. The (non-zero) values of dependency denoted by $Dep_{i,j}$, j^{th} column of i^{th} row of the NxN dependency matrix represents all non-zero hop source-destination pairs in the topology. From the matrix, we take cross product between all non-zero entries. The cross product ensures the generation of all possible sequences of flows (high level scenarios). Therefore, our algorithm is complete and we are guaranteed to generate all possible high level scenario from the given MAC scenario, if sufficient run time is allowed.

Note that we use heuristics to prioritize search alternatives and guide our search to generate the worst case scenario within a realistic run time. We have proposed four heuristics (See Chapter 3 [Begum 2009a] for all heuristics). These heuristics are based on properties of network topology and network nodes. In generating high level scenarios, we use these heuristics to choose among alternative choices. We do not eliminate any choice in this step. Therefore, if sufficient run time is allowed, we are guaranteed to generate all scenarios when using these heuristics.

6.2 Complexity of Our Algorithms

Let N be the number of nodes in the given network topology. Let R be the number of rows in the transition table. Let E and S respectively denote the number of protocol events and number of network node states. Based on the main procedures presented in Appendix II of dissertation [Begum 2009a], the worst case complexity of MAC scenario generation algorithm is $O((N^2E + NS)^4R^2)$.

The worst case complexities of procedure *construct-dependency-matrix* and procedure *enumerate-sequences* are $O(N^3)$ and $O(N^4)$, respectively. Therefore, the worst case complexity of our high level scenario generation algorithm is $O(N^4)$.

6.3 Performance Guarantee Provided by Our Framework

As discussed above, our algorithms generate all high level scenarios given a metric and a network topology. These scenarios are output by our framework as sequences of MAC cycles. We compare time length of MAC cycles of these scenarios to determine the scenario that leads to the worst case performance.

Based on the proven completeness of our search algorithms (also see [Begum 2009]), if sufficient run time is allowed, our algorithms are guaranteed to generate the *true* worst case scenarios. For practical considerations, in our study, we have generated *all* high level scenarios in 10-node and 13-node chain topology using IEEE 802.11 DCF protocol. Then we have compared all these scenarios to identify the scenario that exhibits *true* worst case performance. Finally, we have generated the worst case scenario using our heuristics. For these two topologies, the worst case scenarios generated using our heuristics are the same as the *true* worst case scenarios identified by enumeration. Thus for small topologies, our framework evaluates the *true* worst case performance. For larger topologies, we have shown that our algorithms generate high level scenarios leading to near worst case in the following manner. For larger topologies, the scenarios that we generate subsume the set of *all* scenarios used in related literature, namely, [Li 2001, Gupta 2004, Bensaou 2000, Pathma 2006, Xu 2002], for performance evaluation of the IEEE 802.11 protocol. The near worst case scenario generated by our framework shows the worst performance in simulation among all scenarios we generate (and simulate), for all topologies and all versions of the IEEE 802.11 protocol.

7. SUMMARY AND FUTURE WORK

We propose a new framework for worst case performance evaluation of MAC protocols for wireless ad hoc network. Given a topology, a protocol and its performance metrics, our framework first automatically generates MAC scenarios which exhibit poor MAC performance. In order to evaluate the impact of these MAC scenarios on the end performance, we model the interactions between MAC interface and the MAC layer using a state transition graph and generate high level scenarios using enumeration techniques. We use heuristics to identify high level scenarios that are expected to lead to the worst case end performance.

In order to demonstrate the usefulness, we use our framework to evaluate the worst case performance of IEEE 802.11 DCF protocol by generating library of MAC and high level scenarios. We simulate the high level scenarios to demonstrate that the scenarios we generate expose the worst performance among all the scenarios that we have simulated. Our constructive scenario generation approach gives us insights to identify causes of the worst case performance. We use these insights to derive modified versions of the protocol to improve the worst case performance. Analysis using our performance evaluation framework of all modified versions of the protocol mostly matches with the respective simulation results. For example, we achieve throughput improvement by using our modification in most cases for which we expect improvements. For a fully loaded network, one of our modifications achieves an improvement of the worst case performance of the IEEE 802.11 DCF protocol by up to 89% (see Chapter 3 of dissertation [Begum 2009a]).

We plan to evaluate the worst case performance of IEEE 802.11 networks for applications like TCP. We plan to extend our framework to synthesize topologies that expose the worst case performance of a given MAC protocol.

REFERENCES

Abdrabou, A. and Zhuang, W. 2008. Service time Approximation in IEEE 802.11 Single-hop Ad Hoc Networks. *IEEE Transactions on Wireless Communications 7,* 1 (January).

Barrett, C. L., Marathe, M. V., Engelhart, D. C., and Sivasubramaniam, A. 2002. Analyzing the Short-term Fairness of IEEE 802.11 in Wireless Multi-hop Radio Networks. In *IEEE/ACM International Sysmposium on Modeling, Analysis, Simulation of Computer and Telecommunication Systems.* 137–144.

Begum, S. 2009. A Framework for Worst-Case Performance Evaluation of MAC Protocols for Wireless AdHoc Networks. Ph.D. thesis, University of Southern California, LA, California.

Begum, S., Helmy, A., and Gupta, S. 2009. Modeling and Test Generation for Worst-Case Performance Evaluation of MAC Protocols for Wireless AdHoc Networks. In *IEEE/ACM International Sysmposium on Modeling, Analysis, Simulation of Computer and Telecommunication Systems.* 205–214.

Bensaou, B., Wang, Y., and Kuo, C. C. 2000. Fair Medium Access in 802.11 based Wireless Ad-Hoc Networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing.* 99–106.

Bharghavan, V., Demers, A., Shenker, S., and Zhang, L. 1994. MACAW: A Media Access Protocol for Wireless LANs. In *Special Interest Group on Data Communications.* 212–225.

Bianchi, G. 2000. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas of Communication 18,* 3 (March), 535–547.

Breslau, L. et al. 2000. Advances in Network Simulation. *IEEE Computer 33,* 5 (May), 59–67.

Gupta, N. and Kumar, P. R. 2004. A Performance Analysis of the 802.11 Wireless Lan Medium Access Control. *Communications in Information and Systems 3,* 4 (September), 279–304.

Jurdak, R., Lopes, C., and Baldi, P. 2004. A Survey, Classification and Comparative Analysis of Medium Access Control Protocols for Ad Hoc Networks. *IEEE Communication Surveys and Tutorials 6,* 1, 2–16.

Karn, P. 1990. MACA - A New Channel Access Method for Packet Radio. In *9th ARRL/CRRL Computer Networking Conference.* 134–140.

Li, J., Blake, C., Couto, D., Lee, H., and Morris, R. 2001. Capacity of Ad Hoc Wireless Network. In *ACM International Conference on Mobile Computing and Networking.* 61–69.

Nandagopal, T., Kim, T. E., Gao, X., and Bharghavan, V. 2000. Achieving Mac layer fairness in Wireless Packet Networks. In *ACM International Conference on Mobile Computing and Networking.* 87–98.

Pathmasuntharam, J. S., Das, A., Mohapatra, P., and Gupta, A. K. 2006. A Flow Control Framework for Improving Throughput and Energy Efficiency in CSMA/CA based Wireless Multihop Networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WowMom).* 143–149.

Shah, S. H., Chen, K., and Nahrstedt, K. 2005. Dymanic Bandwidth Management in Single-Hop Ad Hoc Wireless Networks. In *Mobile Networks and Application.* 199–217.

WG, IEEE 802.11. 1997. *IEEE Standard 802.11-1997. Part 11: Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications.*

Xu, K., Gerla, M., and Bae, S. 2002. How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks? In *Global Telecommunications Conference, GLOBECOM.* 72–76.

Zhai, H. and Fang, Y. 2006. Distributed Flow Control and Medium Access in Multihop Ad Hoc Networks. *IEEE Transactions on Mobile Computing 5,* 11 (November), 1503–1514.