IEEE 802.11 MAC Simulator with DCF and Power-Saving Mode

Charles Song
Department of Computer Science
University of Maryland, College Park
csfalcon@excite.com

Table of Content

I. Introduction:

In recent years, wireless communications grew rapidly; this growth is especially evident in the realm of personal and business computing. With the development of wireless networks, computer users are no longer bounded by the length of their network cables. Within a wireless network, computer devices are able to move around to satisfy users' mobility needs. Students can take their computers to their favorite outdoor spot to surf the Internet. Research team members can bring their laptops together during a meeting to discuss and compare results. At locations where network cables are inefficient to reach, wireless network is the perfect replacement. However, the benefits of a wireless network do not come free. Even though wireless networks are more flexible for the users, they are also much more complex than the traditional wired networks.

One of the technologies that made this computing mobility possible is the IEEE 802.11. IEEE 802.11 is a standard protocol proposed for wireless networks; it includes a physical layer (PHY) and a medium access layer (MAC). This protocol controls and manages the stations on the network and their access to the wireless medium. In the heart of IEEE 802.11 is Distributed Coordination Function or DCF; IEEE 802.11 also provides a way for stations to conserve energy when they are operating in the power-saving mode. This simulator is created to run the IEEE 802.11 protocol in a simulated environment, the simulator helps to understand the behavior of this wireless technology. DCF along with power-saving mode is the main focus of this IEEE 802.11 simulator.

II. Background:

1. IEEE 802.11

There are two configurations for an IEEE 802.11 network: infrastructure and ad-hoc. In the infrastructure configuration, stations communicate with other stations through a special kind of nodes called access points or AP. To send a frame through the network, a station must first send the frame to the AP, and then the AP will forward the frame to the destination. AP's are sometimes connected to wired networks; this feature enables the AP's to act as a bridge between wired networks and wireless networks in different locations. In the ad-hoc configuration all communication between stations is done directly without help from the AP. In this configuration AP actually behaves just like regular stations on the network. Infrastructure mode has the advantage of centralized control for sending and receiving wireless frames, however, it poses more overhead than the ad-hoc mode.

IEEE 802.11 standard specifies a physical layer (PHY) and a medium access control layer (MAC). The PHY layer provides the ability for nodes on the network to transmit data, the working on this layer will not be the focus of this paper. The MAC layer functions to maintain order on the wireless medium. Both the distributed coordination function (DCF) and point coordination function (PCF) are used by the MAC layer to control stations' access to the wireless medium. Even though DCF and PCF can coexist in a network, DCF is more commonly used in implemented networks. Therefore DCF is the focus of this paper and is described in more detail in later parts.

Wireless networks are usually designed for mobile applications. In mobile

applications, battery power is an unavoidable issue that must be dealt with. IEEE 802.11 standards address the power management mechanism. In the infrastructure configuration, a station can go into power-saving mode to conserve energy. The AP keeps track of all the stations that are in power-saving mode and buffers frames addressed to these stations. These frames are kept until the stations request them to be sent or discarded if they are not requested for a certain period of time.

2. DCF

The fundamental medium access method of the DCF is carrier sense multiple access with collision avoidance or CSMA/CA; the goal of CSMA/CA is to minimize the possibility of two or more stations accessing the wireless medium at the same time. CSMA requires some mechanisms to detect if the medium is busy and a set of protocols to schedule the access to the medium. DCF has two ways to detect the status of the medium, virtual and physical carrier-sense functions. The virtual carrier-sense mechanism uses small frames call request-to-send (RTC) and clear-to-send (CTS) to announce the time and duration of future data exchanges. All the workstations within the reception range of these frames will learn about the medium reservation, therefore schedule their sending and receiving to avoid this time slot. The RTC and CTS is just one of the virtual carrier-sense mechanisms, the other mechanisms is the Duration/ID field built into every data frame. This field also provides the time and station ID that is reserving the medium. How the physical carrier-sense mechanism detects the status of the

wireless medium is beyond the scope of this paper. With the combination of virtual and physical carrier-sense functions, DCF is able to determine if the wireless medium is busy or not. To achieve collision avoidance, a random back off procedure is used. When a station detects the medium to be busy when it tries to send a frame, it will start a timer with a random time value. This procedure is run by every station; therefore it minimizes the possibility of two stations accessing the medium at the same time.

Basic access refers to the protocol a station follows to determine if it may access the medium. Here is a walk through of a transmission of a frame. When a station wants to send a frame, it first detects the status of the medium. If the medium is idle and continues to be idle for a period of time set in DIFS, then the station gains access to the medium and can start sending the pending frame. However, if the medium become busy during the time the station is monitoring the medium, a random back off procedure will start. A random back off time is chosen for the count down. The timer will decrease the back off time when the medium is idle. If the medium is busy during a station's back off procedure, the back off timer will be suspended. When many stations are competing for the medium, the station that chose the shortest back off time will gain access to the medium first.

DCF uses positive acknowledgment. When a frame is successfully received by the destination station, an ACK frame is sent back to the source station. If the ACK is not received within the allowable duration, the source station sends the frame again. After the source station sends out the data frame following the medium access procedure, it gives up control of the medium to get ready to received ACK from the receiving station. The

sending station waits for a period of time set in SIFS before it attempts to receive the ACK. If an ACK is not received in the mount of time set in ACKTimeout, the station will conclude the transmission has failed and competes for the medium again to resend the frame. The sending station can keep on retransmitting the failed frame for the maximum number of retry attempts, after the maximum is reached, the frame is dropped.

DCF also supports broadcast and multicast frames; these frames are only sent in the infrastructure mode. The station sending the broadcast and multicast frames follows the basic access procedure. Frames are directed to the AP. Once the frames are received by the AP, they are then injected into the network. Every station including the station originated the data frames will receive them, however the data frames will be filtered out by the originated station if the source address contained in the frame matches its address. For the broadcast and multicast frames, no RTS/CTS frames should be used to schedule the transmission, and no ACK should be sent when these frames are received.

3. Power Saving Mode

IEEE 802.11 has different power management modes in the infrastructure configuration. A station can be in one of two power management modes, active and power-saving. In the active mode, a station is fully powered. It can send and receive frames at any time. In the power-saving mode, station can be in one of two states, a sleep state and an awake state. Most of the time, a station in the power-saving mode remains in the sleep mode, it only get into the awake state to listen to management frames call beacons at a certain interval and receives frames from AP at its convenience. In this mode,

a station consumes very low power. When a station is in the power-saving mode, the AP buffers all the frames that are directed to that station. Embedded in the beacon frames that the AP sends out is TIM. TIM contains information about pending frames. By reading the TIM, a station in power saving mode can determine if there are data frames stored for it and decide if it wants to change to the awake state to receive the pending frames from the AP.

To enter the power-saving mode, a station must first inform the AP. A frame with a power saving request is sent from the station to the AP following the basic medium access procedure. A reply should be sent by the AP and received by the station before the station can enter the power saving mode. Once the exchange is successful, the station can operate with very little power consumption and AP buffers all the frames addressed to this station. However, if the request or the reply was not successfully exchanged, the station will remain in active mode and retransmit the request again to the AP.

Once the station is in the power saving mode, it periodically changes to the awake state and listens for beacon frames sent by the AP. This interval for listening to beacon frames is defined by the value in BeaconInterval. Contained in the beacon frames are information coded in a partial virtual bitmap called TIM. TIM indicates whether any frames directed to this station are pending in the AP. There are two different types of TIMs. AP sends out a TIM with every beacon, but every DTIMPeriod a DTIM is sent in a beacon instead of a TIM. DTIM signals that the AP is about to send out the buffered broadcast/multicast frames using normal frame transmission rules. The MoreData field

can be set in the frame to inform the station if there are more frames after the current transmissions. The station in the power-saving mode can not choose when to receive these broadcast and multicast frames, but it can choose to ignore these transmissions.

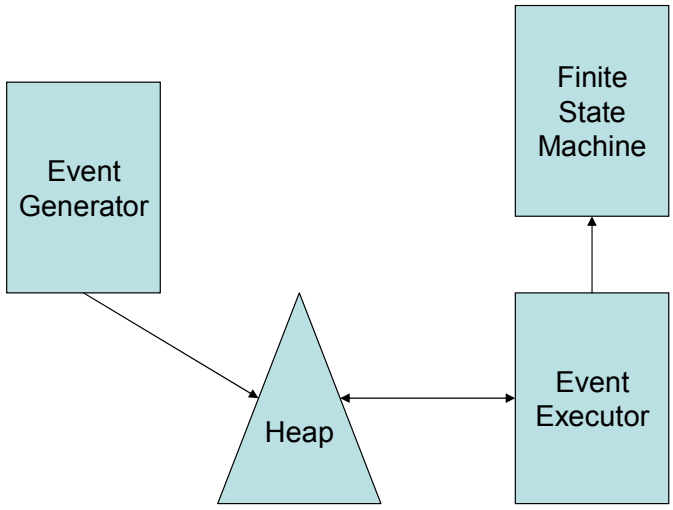
If in the TIM there is an indication of pending unicast frames, the station can choose to receive those frames at its convenience. To receive a unicast frame, the station sends out a PS-Poll to the AP, this signals that the station is ready to receive a frame. After the reception of the PS-Poll, the AP forwards the pending frame to the station. The MoreData field can be set in the data frame to indicate further pending frames buffered by the AP. After a successful transmission of data frames, the station can either go back to the doze state or choose to receive more frames by sending out another PS-Poll. If no more frames are buffered in the AP, then the station will go back to the doze state.

4. IEEE 802.11 Simulators

Other IEEE 802.11 simulators have used states of communicating machines to describe the function of the MAC layer. The IEEE 802.11 simulator described in this paper is based on the specification of the DCF protocol analyzed by Arunchandar Vasan and Raymond Miller of University of Maryland. In addition to the DCF protocol specified in their research paper, this emulator will simulate the power-saving mode that is specified by the IEEE 802.11 standards. Also, the sending and receiving of multicast frames in DCF are also simulated.

III. Design and Implementation

1. Overall Architecture



Simulator Overall Architecture

The simulator can be divided into four major components; an event generator, an event executor, a heap and a finite state machine. The event generator creates different events that can happen on a wireless network, events such as medium becomes busy, or data frame is ready to be send out. This newly created event is then saved in a heap structure. This structure is a shared data structure; it serves as a way for the event generator and the event executor to work together. The events are inserted and taken out from the heap. There are two different kinds of events, absolute events and relative events. Relative events builds the absolute events, for example, when a station wants to sent to

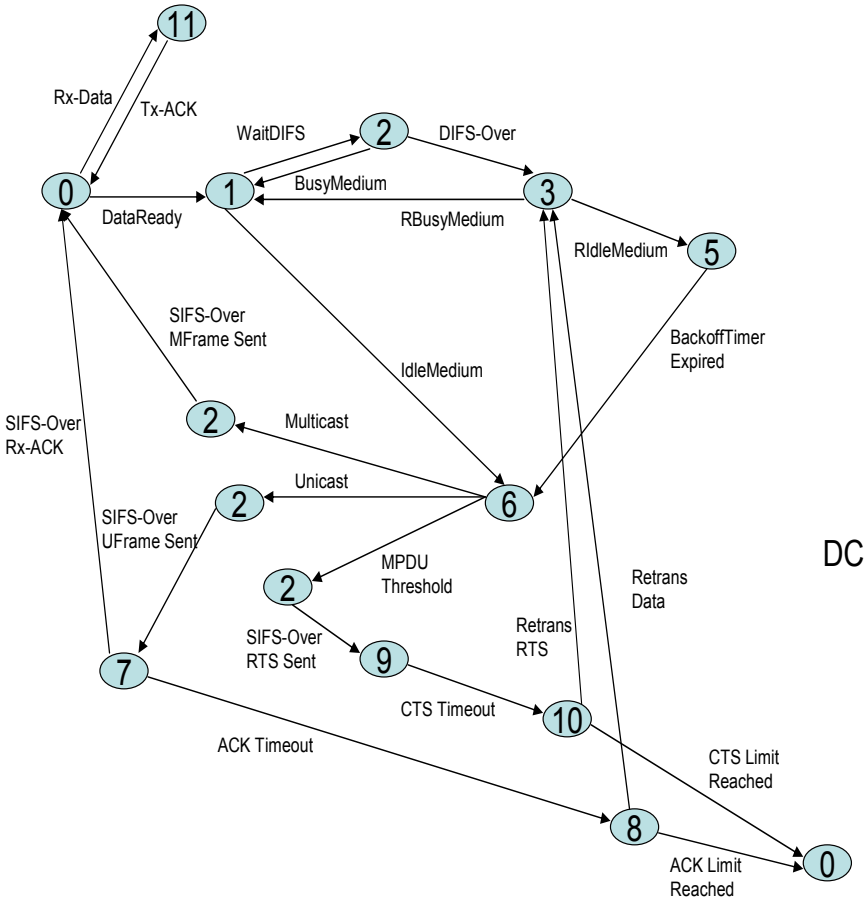
other station in the infrastructure configuration, the command to send would be an absolution event. But sending of the ACK back to the source would be a relative event. There is another way to distinguish between absolute and relative events; the absolute events are generated by the event generator and relative events are generated by the event executor. Event executor takes the absolution events from the heap and processes them. If a relative event needs to be generated, the event executor inserts a new event into the heap for later processing. The event executor processes the events with the help from the finite state machine module or FSM. The FSM is the simulator for each station in the network. Every station in the simulation will have its own instantiation of the FSM. The FSM changes states according to the events that are being executed by the event executor. In the FSM is where DCF and power-saving mode being implemented.

2. Finite-State-Machine Specification

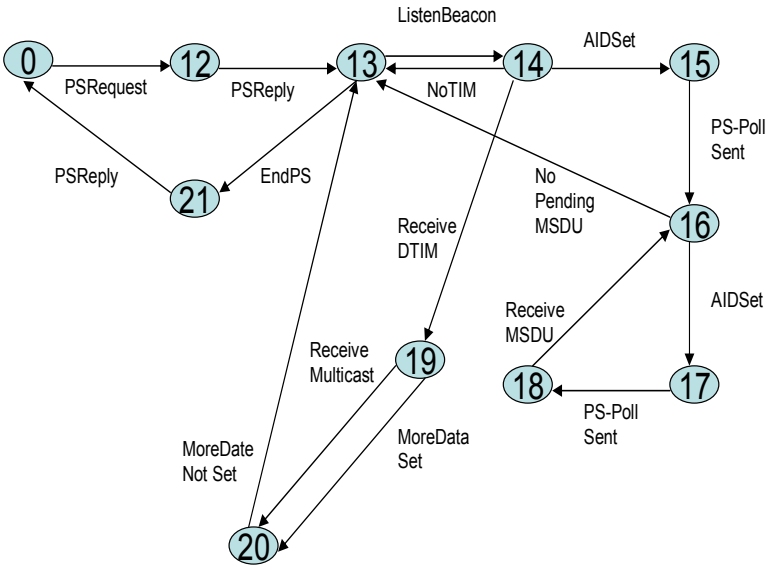
Transition	Predicate	Action
DataReady	HaveData = true	Backoff-Timer.val = rand()
IdleMedium	Medium = 0	HaveChannel = true
WaitDIFS	Medium != 0	DIFS-Timer.st = ACT
	Backoff = false	DIFS-Timer.val = STD
WaitEIFS	Retrans = true	EIFS-Timer.st = ACT
	Medium != 0	EIFS-Timer.val = STD
	Backoff = false	
BusyMedium	DIFS-Timer.st = ACT V	Backoff-Timer.val = rand()
	EIFS-Timer.st = ACT	
	Medium != 0	
DIFS-Over	DIFS-Timer.st = EXP	Backoff-Timer.val = STD
EIFS-Over	EIFS-Timer.st = EXP	Backoff-Timer.val = STD
RBusyMedium	Medium != 0	Backoff-Timer.st = INACT
RIdleMedium	Medium = 0	Backoff-Timer.st = ACT
Backoff-Timer Expired	Backoff-Timer.st = EXP	HaveChannel = true
Multicast	HaveChannel = true	SIFS-Timer.val = STD
	ToDs = 0	SIFS-Timer.st = ACT
Unicast	HaveChannel = true	SIFS-Timer.val = STD
	ToDs = 1	SIFS-Timer.st = ACT
MPDU Threshold	HaveChannel = true	SIFS-Timer.val = STD
	Length > Threshold	SIFS-Timer.st = ACT
SIFS-Over MFrame Sent	SIFS-Timer.st = EXP	HaveChannel = false

SIFS-Over UFrame Sent	SIFS-Timer.st = EXP	WaitACK = true
		SIFS-Timer.val = STD
		SIFS-Timer.st = ACT
		ACK-Timer.val = STD
		ACK-Timer.st = ACT
SIFS-Over RTS Sent	SIFS-Timer.st = EXP	WaitCTS = true
		SIFS-Timer.val = STD
		SIFS-Timer.st = ACT
		CTS-Timer.val = STD
		CTS-Timer.st = ACT
SIFS-Over ACK Rx	SIFS-Timer.st = EXP	HaveChannel = false
	WaitACK = true	ACK-Timer.st = INACT
SIFS-Over CTS Rx	SIFS-Timer.st = EXP	HaveChannel = false
	WaitCTS = true	CTS-Timer.st = INACT
ACK Timeout	ACK-Timer.st = EXP	ACKLimit++
	WaitACK = true	Retrans = true
CTS Timeout	ACK-Timer.st = EXP	CTSLimit++
	WaitCTS = true	Retrans = true
ACK Limit	ACKLimit >= ACKLIMIT	HaveChannel = false
CTS Limit	CTSLimit >= CTSLIMIT	HaveChannel = false

3. State Diagrams



DCF State Description



Power Saving Mode