

LABORATORY FOR COMPUTER COMMUNICATIONS AND APPLICATIONS
(EPFL-I&C-LCA)

HANDS-ON EXERCISES: IEEE 802.11b STANDARD

MARIO ČAGALJ JEAN-PIERRE HUBAUX

IMAD AAD

{mario.cagalj, jean-pierre.hubaux, imad.aad}@epfl.ch

November 9, 2004

1 Introduction: IEEE 802.11b standard

The scope of the IEEE 802.11 [4] standard is to provide specifications for wireless connectivity for fixed, portable and moving stations within a local area. It defines over-the-air protocols necessary to support networking in a local area. This standard provides MAC and physical layer functionality. The extension IEEE 802.11b (used in our hands-on exercises) gives accommodation of transmission rates of up to 11 Mbps and operates in the 2.4 GHz band. The IEEE 802.11 standard takes into account of power management, bandwidth, security and addressing, since these are the significant differences from wireless to wired LANs. The MAC layer specification of the IEEE 802.11 standard provides radio channel access control functions, such as addressing, access coordination etc.

The Distributed Coordination Function (DCF) is the primary access protocol for the automatic sharing of the wireless medium between stations and access points. This DCF uses a carrier sense multiple access/collision avoidance (CSMA/CA) protocol for sharing the wireless medium.

As shown in Fig. 1, the DCF delays frame transmissions right after the channel is sensed idle for DIFS (DCF InterFrame Spacing) time. It waits for an additional random time, *backoff time*, after which the frame is transmitted. The backoff time is bounded by the contention window size CW . This is applied to data frames in the basic scheme, and to RTS frames in the RTS/CTS scheme. The backoff time of each station is decreased as long as the channel is idle. When the channel is busy, backoff time is frozen. When backoff time reaches zero, the station transmits its frame. If the frame collides with another frame (or RTS), the sender times out waiting for the ACK (or the CTS) and computes a new random backoff time with a larger CW to retransmit the frame with lower collision probability. When a frame is successfully transmitted, the CW is reset to CW_{min} . The network allocation vector (NAV) of all other stations is set to the frame duration field value in RTS/CTS and DATA headers.

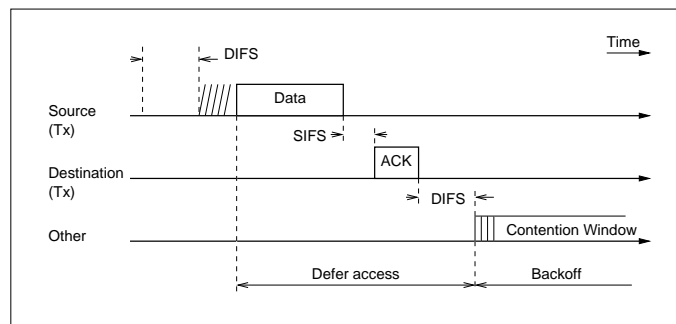


Figure 1: The distributed coordination function (DCF) of IEEE 802.11 operating in the basic mode.

Because of the possibility of partial network connectivity, wireless LAN protocols must

take into account the hidden terminal problem (this occurs when a station is able to receive frames from two different stations but these two stations can not hear each other). To solve this a virtual carrier sense mechanism through the exchange of control frames is used (Figure 2). These are the Request to Send (RTS) and the Clear to Send (CTS) frames. The RTS and CTS frames contain a duration field that defines the period of time that the medium is to be reserved to transmit the actual data frame and the returning ACK frame. All stations within the reception range of either the originating station (which transmits the RTS) or the destination station (which transmits the CTS) shall learn of the medium reservation. Thus a station can be unable to receive from the originating station, yet still know about the impending use of the medium to transmit a data frame. The RTS/CTS control frames should not be used for short data frames, since they would add traffic. According to IEEE 802.11b standard, the use of RTS/CTS mechanism is optional.

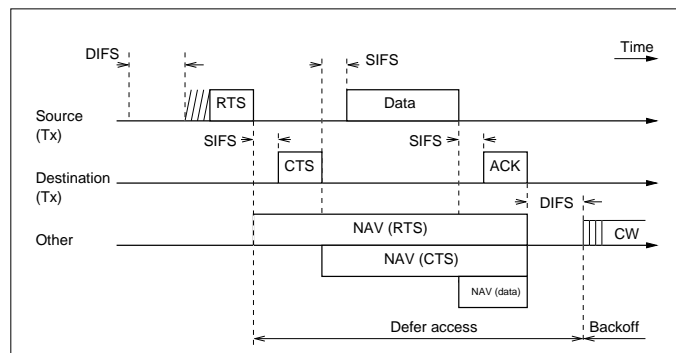


Figure 2: The distributed coordination function (DCF) of IEEE 802.11 operating in RTS/CTS mode.

IEEE 802.11 can be used in two different operating modes:

- (i) Infrastructure mode (all communication goes through an access point (AP)).
- (ii) Ad-Hoc mode (no AP is present; all communication is peer-to-peer).

2 Configuration of wireless client adapters (Infrastructure mode)

Our goal in this exercise is to set up an operational Wireless LAN (WLAN). For this, we will use Cisco Aironet 350 Series Client Adapters available in PCMCIA form. However, in order to plug in the wireless cards to available desktops, we use PCMCIA-to-PCI adapters. Recall that wireless client adapters connect a variety of devices to a wireless network either in ad hoc peer-to-peer mode or in infrastructure mode with access points (AP).

Please note that, once plugged in, the wireless adapters are ready to use, since the driver is already installed, as well as the client utilities *Aironet Client Utility (ACU)*. Before we proceed, let us just remind that each bench (BANC) comprises two machines, one labeled *Router* and the other one *Station*. Please note that in our exercises there is no difference between the two.

Do the following on both, the Station and Router.

TASK 1: Minimal configuration

1. Remove any Ethernet cabling
2. Plug in the client wireless adapter. Upon insertion you should get two beeps, confirming that the card is detected and the drivers are loaded successfully.

TASK 2: Basic settings

1. Log on to a machine and start an *xterm*
2. Assign an IP address to the inserted wireless adapter by typing the following:

```
#ifconfig xxx 192.168.100.abc netmask 255.255.255.0 up
```

where `xxx` is the name assigned to the wireless interface (e.g. `eth1` for Station; `eth2` for Router), and `192.168.100.abc` is the IP address assigned to the machine.¹ We use the following addressing plan to assign IP addresses to machines. All the machines use `192.168.100` as a common part of their IP address. Then we set:

`a` - to 1 in the case of the Router; to 2 in the case of the Station

`bc` - to the bench number (e.g. `06` in the case of BANC 6)

¹You can learn the identifier assigned to your wireless adapter by typing `ifconfig -a` at the command line with the adapter first unplugged and then plugged in.

3. In the case of the conventional wired networks, once we have interconnected two machines via a hub and assigned IP addresses to them as above, the machines can communicate with each other. However, this is not the case with the wireless IEEE 802.11 protocol. To see this, try to ping one machine from the other one. We will correct this in the task below.

TASK 3: Infrastructure mode

This mode is used to set up a connection to a wired network. This mode requires an Access Point to gain access to the wired part of the network. Note that in Infrastructure Mode, an adapter scans frequency channels to find an Access Point. Thus, we do not have to set the channel by ourselves.

1. Start *Aironet Client Utility* by typing:

```
#acu &
```

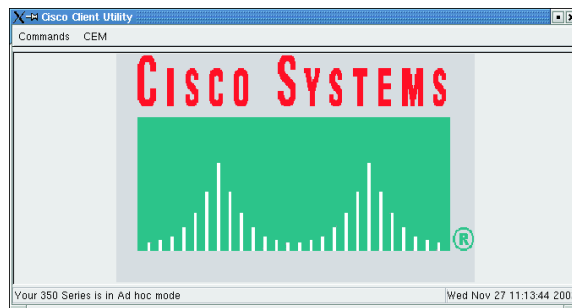


Figure 3: Aironet Client Utility (ACU)

2. In Commands menu of ACU select Edit Properties.

Fill out the text fields in the window that appears as follows:

Client Name - you can put whatever in this field, but in these exercises we use the following naming policy; *station-[bench No.]* for the Station and *router-[bench No.]* for the Router (e.g. in the case of bench 6, *station-6* and *router-6*).

SSID - Service Set ID (SSID) is a unique identifier that client devices use to associate with either AP or other client. This value **MUST** match the SSID of an access point (AP) that we want to communicate with. In our case, the SSID is *iew*. Do not forget that the SSID is case sensitive.

Network Type - here we check *Infrastructure*.

Current Profile - check *Use Enterprise Configuration* to use the above settings.

To save the settings, click the *Ok* button.

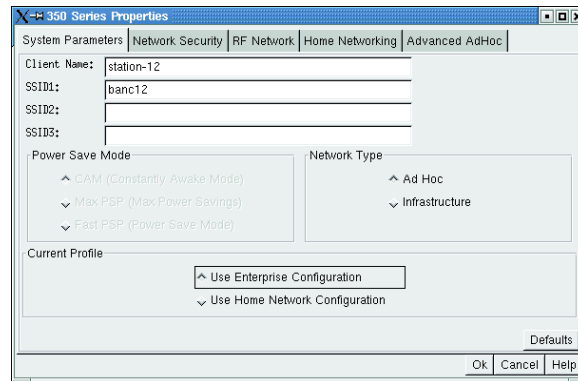


Figure 4: System Properties screen

2. Try to ping other machines in the room, as well as the access point (AP). The AP's radio and Ethernet ports can be accessed via IP address 192.168.100.254.

The station having IP address 192.168.100.200 is connected via the Ethernet cable to the AP's Ethernet port. Try to ping it. You should perform this test before proceeding, since this station will be used as an FTP server in the exercise to follow.

IMPORTANT: Start the `ethereal` tool to monitor packets exchanged between your station and the station having IP address 192.168.100.200. From the output produced by the `ethereal` tool retrieve the MAC address corresponding to the station with IP address 192.168.100.200. Note down this MAC address for exercises later on.

WARNING: In order to illustrate security aspects, these exercises describe some techniques aiming at subverting the normal behavior of a network. Please note that the usage of these techniques outside of the scope of the practical exercises is strictly prohibited.

3 Security issues I: passwords sniffing, DoS and MITM attacks

The goal of this exercise is to raise awareness about vulnerabilities of IEEE 802.11 protocol. All the vulnerabilities that we will present here are common to conventional wired networks. However, the properties of the radio channel makes an attacker more powerful and harder to detect. Thus, for example, the attacker, in order to sniff a IEEE 802.11 traffic, only has to be located somewhere within the communication range of possible victims (e.g. at a nearby parking lot).

In this exercise we will use a collection of tools for network auditing and penetration testing, called `dsniff` [1]. We will use only one tool out of that collection, namely: `arpspoof`. This tool is used to poison ARP cache of a machine.

Here we do not explore vulnerabilities of WEP (wired equivalent privacy) functionality (a form of data encryption used to scramble the data sent over the radio link). We just mention that tools for breaking the WEP encryption can already be found on the web (e.g. an open source tool `WEPCrack`).

Before we proceed, let us just mention that there are many real life events where the attacks to follow could be easily mounted. Thus, for example, during the course of many scientific conferences, the participants are generously offered with a *free* access to the Internet via IEEE 802.11b wireless LAN. However, usually no data encryption is used.

In this exercise, an attacking machine will be the laptop with the installed `dsniff` collection on it. The desktops available in the lab will play the role of either FTP client or FTP server (i.e. systems under attack).

TASK 1: Passwords sniffing

1. Make both machines of your bench work in the infrastructure mode. Set their SSID to `iew`.
2. Start the `ethereal` tool to monitor packets exchanged during the course of this exercise.
3. Set up an FTP session between your machine and the FTP server having IP address 192.168.100.200. However, when prompted for the user name type `anonymous`, while for the password type whatever you want (please make it human readable). Are you able to monitor the traffic (non-broadcast) of your neighbors (e.g., with `ethereal`)?

4. After you have initialized a session with the FTP server, you do not have to download anything for an attack to be successful. Thus, just terminate the session by typing:

```
ftp> e
```

5. Finally, you check whether your password has been captured by the attacking machine or not.

Basically, you have just experienced password sniffing attack. Even though you are not able to monitor the traffic from other machines in the room, the attacking machine is still able to sniff the traffic. The trick is that the wireless adapter used on the attacking machine is set up to work in the “monitoring” mode. Since the channel is not encrypted, it is straightforward for us to retrieve any interesting information, including your passwords.

In the following exercise we will perform the DoS attack, i.e., we will deny the access to our FTP server. What is interesting with this attack is that all that we need to successfully mount the DoS attack is the IP address of the FTP server, `arpspoof` tool and of course a wireless adapter. In addition, our attacking machine should be within the reception range of the machines under the attack (your own machines).

By using `arpspoof` we will force all the FTP clients to pass their traffic through the attacking machine. This is done by broadcasting fake ARP replay messages that bind different IP addresses (one of the FTP server) to the MAC address of the attacking machine. Finally, to mount the DoS attack the attacking machine simply drops all the packets arriving at it.

TASK 2: DoS attack

1. As in the Task 1, make both machines of your bench work in the infrastructure mode. Set their SSID to `iew`.
2. Start the `ethereal` tool to monitor packets exchanged during the course of this exercise.
3. Check the machines ARP caches occasionally by typing:

```
# arp -a
```

Compare the MAC address bound to IP address 192.168.100.200 with the MAC address of the FTP server 192.168.100.200 as retrieved in the very first exercise (Infrastructure mode). You can also observe fake ARP response packets through `ethereal`.

4. Try to set up an FTP session between your machine and the FTP server having IP address 192.168.100.200. Did you manage to get through?

4 How much do we get out of 11Mbps? (Infrastructure mode)

IEEE 802.11b adapters operates at the maximum data rate of 11Mbps. Supported Data Rates for the Aironet 350 include 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. The data rate of 11 Mbps sounds fairly good (~ 1 MBps). Note that this is the 802.11b raw data rate at the physical level of the network. In this exercise, we will measure the data rate at the network layer. Measurements will be performed on a real-life scenario (i.e. several stations compete for the available bandwidth).

For the following tasks, you will use a program `iptraf`, which allows us to monitor IP network statistics (e.g. data rates). To start this tool, open a new terminal (`xterm`) and type `iptraf` at the command line.

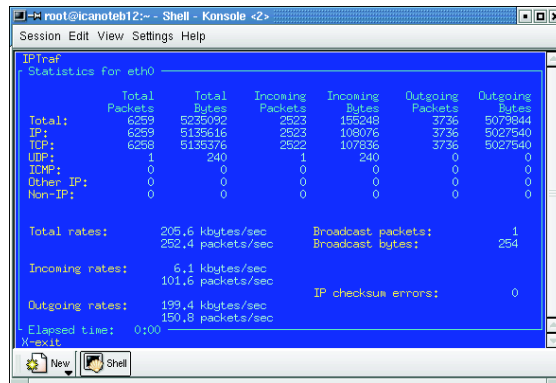


Figure 5: Iptraf screen

TASK 1: FTP server goes wired

In this task, an FTP server is connected to the AP's Ethernet port via an Ethernet cable. The station with IP address 192.168.100.200 plays the role of the FTP server. Do the following on both machines of your bench.

1. Configure each machine to work in the infrastructure mode. Set its SSID to `iew` as instructed in the previous exercise.
2. Set up an FTP session between your machine and the FTP server (192.168.100.200), which is connected via a wired link to the LAN, by typing:

```
#ftp 192.168.100.200
```

When prompted for the user name type `anonymous`, while for the password just press the *Enter* key.

- Before downloading `ws-test` file to the client, change the local directory the file `ws-test` will be stored to by typing:

```
ftp> lcd /home/ftp/pub
```

- Start the download of the file `ws-test` as shown below, and observe the data rate with `iptraf`. Memorize it for comparison later on.

```
ftp> get ws-test
```

- You can check a summary report on activities of the access point (the number of stations associated with AP, their IP and MAC addresses etc.) by typing the AP's address 192.168.100.254 in the *Address* field of a web browser (e.g. Netscape).

The screenshot shows a web browser window titled "AP350-5aaa23 Summary Status - Microsoft Internet Explorer provided by VIFonline... [Working Offline]". The address bar shows "http://192.168.100.254". The page content includes the Cisco logo and the following sections:

AP350-5aaa23 Summary Status
Cisco 350 Series AP 11.21

Home Map Network Associations Setup Logs Help 2002/11/27 10:05:16

Current Associations

Clients: 2 of 2	Repeaters: 0 of 0	Bridges: 0 of 0	APs: 1
-----------------	-------------------	-----------------	--------

Recent Events

Time	Severity	Description
2002/11/27 10:03:31	Info	Station [CAGALL-NTEK10009b77bb693] Associated
2002/11/27 10:03:31	Info	Station [CAGALL-NTEK10009b77bb693] Authenticated
2002/11/27 10:02:32	Info	Station [linug-mario10009b77bb693] Associated
2002/11/27 10:02:32	Info	Station [linug-mario10009b77bb693] Authenticated
2002/11/27 10:01:11	Info	Station [linug-mario10009b77bb693] Associated

Network Ports *Diagnostics*

Device	Status	Mb/s	IP Addr.	MAC Addr.
Ethernet	Up	100.0	192.168.100.254	0040965aaa23
AP Radio	Up	11.0	192.168.100.254	0040965aaa23

[Home][Map][Login][Network][Associations][Setup][Logs][Help]
Cisco 350 Series AP 11.21 © Copyright 2001 Cisco Systems, Inc. credits

Figure 6: Access Point summary table

How does the data rate obtained on your machine compare with the data rate obtained on other machines? How does it compare with the nominal data rate of 11Mbps?

Note that the number of the FTP clients in this exercise is at most 28. These are ones with which your machine shares the available bandwidth.

TASK 2: FTP server goes wireless

In this task, the FTP server is connected to the AP's radio port via a wireless adapter. The station with IP address 192.168.100.201 plays the role of the FTP server. Before

proceeding with this task please make sure that the FTP server is connected via the radio channel to the LAN.

Do the following on both machines of your bench.

1. Set up an FTP session between your machine and the FTP server (192.168.100.201), which is connected via the radio link to the LAN.
2. Start the download of the file `ws-test` as shown below, and observe the data rate with `iptraf`.
3. As before, you can check a summary report on activities of the access point (the number of stations associated with AP, their IP and MAC addresses etc.) by typing the AP's address 192.168.100.254 in the *Address* field of a web browser (e.g. Netscape).

How does the obtained data rate compare with the data rate obtained in the Task 1 above? How does it compares with the 11Mbps?

5 Ad Hoc mode

In the sequel we study the Ad Hoc mode. This mode is used to set up a small, temporary network between two or more computers.

TASK 1: Basic configuration

1. In Commands menu of ACU select Edit Properties. Fill out the text fields in the

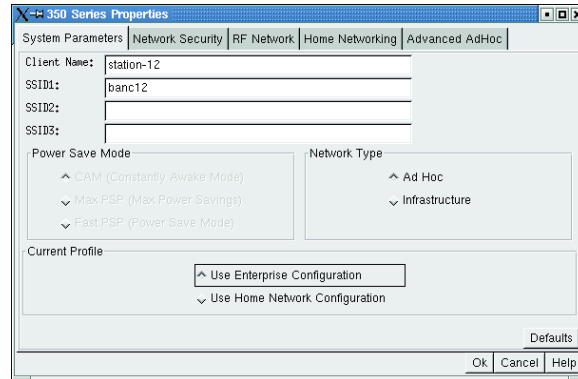


Figure 7: System Properties screen

window that appears as follows:

Client Name - the same values as for the Infrastructure mode.

SSID - Service Set ID (SSID) is a unique identifier that client devices use to associate with either AP or other client. This value **MUST** match the SSID of any other wireless client that you want to communicate with. In our exercises, we use *banc-[bench No.]* as the SSID on both, the Station and Router (e.g. for bench 6 we use *banc-6*). Note that the SSID is case sensitive.

Network Type - here we check *Ad Hoc*.

Current Profile - check *Use Enterprise Configuration* to use the above settings.

2. In the Ad Hoc mode, the frequency channel must match the channel used by the other Adapters you wish to communicate with. In the window Edit Properties, select the RF Network tab and set the channel your adapter will according to a *channel assignment plan*².

In the same window, set Transmit Power to 1mW to reduce the interference sensed by other adapters; and set Data Rate to 11Mbps.

²You may use any available channel except the channels 1 and 11. This is to prevent interference with the EPFL operational network (channels 1 and 11). For example, you can have the bench number determines the channel to be used (benches 1, 11 and 14 should use some other values).

NOTE: In order to avoid frustrating situations of not being able to change the channel already assigned to your adapter, try to turn off both the Station's and Router's radios before performing this task (Toggle radio on/off in the menu Commands). It worked with us.

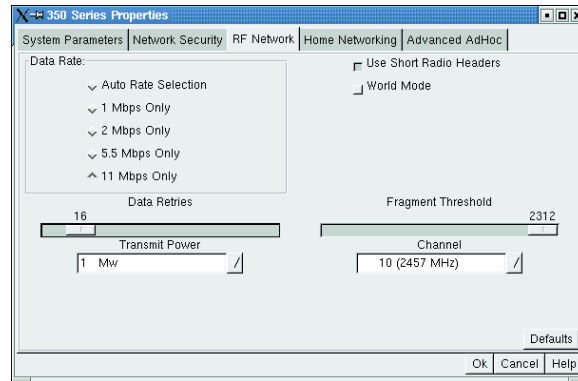


Figure 8: RF Network screen

To save the settings, click the *Ok* button.

3. Try to ping the Router from the Station and vice versa. Does it work now? If yes, congratulation, you have just set up an operational wireless ad hoc network.

IMPORTANT: In this exercise we will again extensively use `ws-test` file. If for any reason the size of file is not sufficient, you can generate a new one by running the following script in `/home/ftp/pub` directory:

```
#./file-gen
```

TASK 2: How much do we get here out of 11Mbps?

1. Set up an ad hoc network comprising the Station and Router as instructed in the previous exercise.
2. Next, set up an FTP session between the Station and Router. Since both of them run an FTP server daemon, it is irrelevant which of the two will play the role of the FTP server. In any case, we type at the client:

```
#ftp 192.168.100._
```

where `192.168.100._` is the IP address of the FTP server. When prompted for the user name type `anonymous`, while for the password just press the *Enter* key.

3. Before downloading `ws-test` file to the client, change the local directory the file `ws-test` will be stored to by typing:

```
ftp> lcd /home/ftp/pub
```

4. Start the download of the file `ws-test` as shown below, and observe the data rate with `iptraf`.

```
ftp> get ws-test
```

5. You may want to check the signal level at your adapter. For this, open the Status window in the Commands menu of ACU. The Signal level indicator is at the bottom of the window.



Figure 9: Status screen

What is the data rate obtained in this scenario? How does it compare with the nominal data rate of 11Mbps?

Note that the number of the FTP sources in this exercise is at most 14. Basically, these are ones with which the FTP server of your bench shares the available bandwidth.

TASK 3: Fairness issues

1. In this task you should coordinate your activities with the activities of your colleagues at a neighboring bench (located either on your left or your right). That is, you should select the same channel as the one of the neighboring bench.
2. In the RF Network window of both the Station and Router, set Transmit Power to 100mW make sure that you are heard by other stations in the room. The main reason for doing this is to achieve as symmetric topology as possible. Since we are studying fairness, we want each station to experience the same channel conditions.
3. Next, set up an FTP session between the Station and Router at each bench as in Task 2 above. However, here we want a symmetric topology. Thus, given a network of four lined up stations, either the two inner stations play the role of the FTP servers or the two outer.
4. Start to download the file `ws-test` simultaneously on both benches and observe the data rates with `iptraf` (possibly on both machines).

Is IEEE 802.11 fair, that is, is the available bandwidth shared in a fair manner between the FTP servers?

Do you have an idea what happens with the capacity of ad hoc networks that use IEEE 802.11 the *Distributed Coordination Function (DCF)* for sharing the radio channel, when the number of contending stations increases, given that all the stations use the same frequency channel?

6 Distributed Coordination Function (DCF): RTS/CTS mechanism

The basic medium access protocol is a DCF that allows sharing of the radio channel through the use of CSMA/CA (*carrier sense multiple access with collision avoidance*) and a random backoff time following a busy medium condition. Carrier sense (CS) is performed through physical and virtual mechanisms.

One means to achieve a virtual carrier sense mechanism is through the exchange of short (compared to data frames) RTS/CTS frames. Every RTS/CTS frame contains a *network allocation vector (NAV)* through which a sender and a designated receiver reserve the radio channel for the exchange of future traffic. Other stations, when receiving RTS or CTS frames, regard the radio channel busy for the period specified in NAV.

The RTS/CTS mechanism is primarily included in the IEEE 802.11 protocol to deal with the *hidden terminal problem*. Recall that this problem occurs when a station is able to receive frames from two different stations but the two stations cannot hear signals from each other. In this case a station may sense the radio channel as being idle even if the other one is transmitting, which results in a collision at the receiving station.

According to the IEEE 802.11 standard, the use of RTS/CTS mechanism is under control of the *RTSThreshold* attribute. This mechanism allows a station to be configured to use RTS/CTS either always, never, or only on frames longer than a specified length. When the transmitted packet is equal to or larger than the RTS threshold, an RTS/CTS mechanism is used.

TASK 1: **RTS Threshold** = *max_value*

1. In this task you should coordinate your activities with the activities of your colleagues at a neighboring bench (located either on your left or your right). This is to ensure that at least two servers compete for the available bandwidth.

Set up an ad hoc network comprising the Station and Router. Set the same frequency channel on both benches.

2. In the RF Network window of both the Station and Router, set Transmit Power to 100mW.

To save the settings, click the *Ok* button.

3. Do the following on the FTP servers of both benches. In the Commands menu of ACU, select Edit Properties and select the Advanced AdHoc tab. In the Advanced AdHoc window, set RTS Threshold to the maximum value (i.e. 2312).

To save the settings, click the *Ok* button.

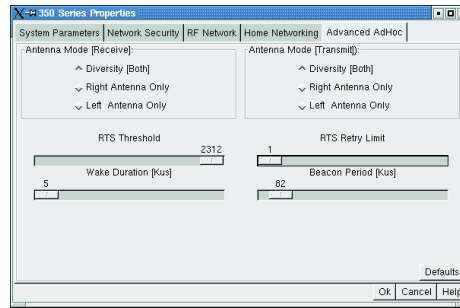


Figure 10: Advanced AdHoc screen

- Do the following on the FTP servers. In the Commands menu of ACU, select Statistics to open the Statistics window with the current statistics from the wireless adapter, including the number of RTS frames transmitted.

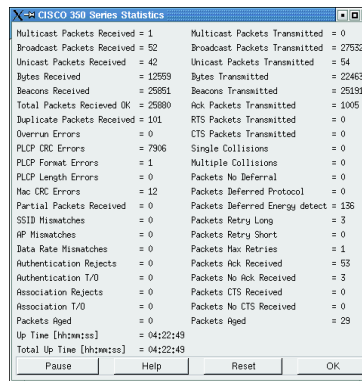


Figure 11: Statistics window

NOTE: You can also monitor the number of CTS packet transmitted by the FTP clients.

- Next, set up an FTP session between the Station and Router on both benches.
- Before starting to download the file `ws-test` simultaneously on both benches, press the Reset button in the Statistics window. Observe the data rate with `iptraf`.

Memorize observed values for the data rate and the number of RTS packet transmitted by the servers.

TASK 2: RTS Threshold = small.value³ (e.g. 12)

³Do not set this to 0.

1. Again, you should coordinate your activities with the activities of your colleagues at a neighboring bench (located either on your left or your right). Do the following on the FTP servers of both benches. In the Commands menu of ACU, select Edit Properties and select the Advanced AdHoc tab. In the Advanced AdHoc window, set RTS Threshold to some value close to 0, but not 0 (e.g. 12).

To save the settings, click the *Ok* button.

NOTE: You can also monitor the number of CTS packet transmitted by the FTP clients.

2. Do the following on the FTP servers. In the Commands menu of ACU, select Statistics to open the Statistics window with the current statistics from the wireless adapter, including the number of RTS frames transmitted.
3. Next, set up an FTP session between the Station and Router on both benches.
4. Before starting to download the file `ws-test` simultaneously on both benches, press the Reset button in the Statistics window. Observe the data rate with `iptraf`.

Compare the results obtained in the tasks 1 and 2. Pay particular attention to the difference in the number of RTS packets transmitted by the FTP servers.

How does the number of CTS packets transmitted by FTP clients compare with the number of RTS packets transmitted by the corresponding FTP servers?

7 Routing in mobile ad hoc networks: AODV routing protocol

The purpose of this exercise is to demonstrate the feasibility of multihop wireless networks. You will also learn a fundamental limitation of multihop wireless networks based on IEEE 802.11b protocol.

Routing algorithms aim at finding a path between a source and a destination station that are not necessarily within the reception range of each other. Existing routing protocols can be classified into two categories:

Proactive routing. Protocols in this category keep track of routes from a source to all destination in the network (even if a station will never use some of the routes). In this way, as soon as a route to a destination is needed, it can be selected in the routing table. The advantages of a proactive protocol are that communication experiences a minimal delay and routes are kept up to date. The disadvantages are the additional control traffic and that routes may break, as a result of mobility, before they are actually used or even that they will never be used at all, since no communication may be needed from a specific source to a destination.

Reactive routing. In contrast to proactive routing protocols, reactive (or on-demand) routing protocols find a path between the source and the destination only when the path is needed (i.e., if there are data to be exchanged between the source and the destination). An advantage of this approach is that the routing overhead is greatly reduced. A disadvantage is a possible large delay from the moment the route is needed (a packet is ready to be sent) until the time the route is actually acquired.

The AODV (Ad-Hoc On-Demand Distance Vector) routing protocol [5, 6] is a reactive routing protocol that uses some characteristics of proactive routing protocols. Routes are established on-demand, as they are needed. However, once established a route is maintained as long as it is needed.

In this exercise we will use the AODV implementation [2] developed at Uppsala University, Sweden. The release we will use is based on AODV draft version 11.

TASK 1: Minimal configuration

1. Coordinate your activities with the activities of your colleagues at a neighboring bench (located either on your left or your right) as you are expected to set up an ad hoc network of 4 stations.
2. Put your wireless adapter in ad hoc mode. Make sure that you use the same channel and SSID as your colleagues at the selected bench. Assign an IP address to your wireless adapter; use the naming convention as specified in Chapter 1.

3. Open a new terminal window and change your current directory:

```
#cd /usr/local/aodv-uu-0.6
```

4. Start the AODV process as follows:

```
#aodvd -R
```

5. To check that AODV is successfully run, try to ping other machines in the established ad hoc network.

Since in our workshop all the machines are within the reception range of each other (i.e., any pair of machines can communicate directly with each other), we will apply the following technique to simulate multihop communication.

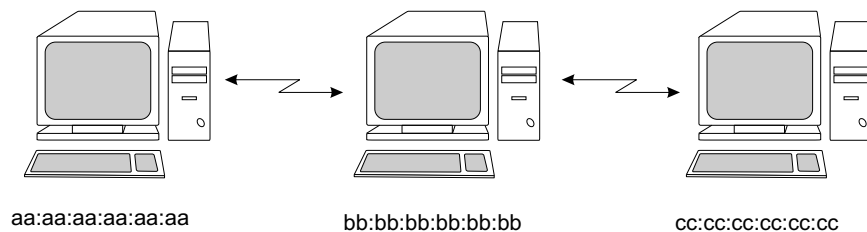


Figure 12: Multihop wireless network

Assume that we want to simulate a situation in which machines with MAC addresses `aa:aa:aa:aa:aa:aa` and `cc:cc:cc:cc:cc:cc` cannot transmit directly to each other (see the figure above). However, both machines can hear machine `bb:bb:bb:bb:bb:bb`. To achieve this we will use the `iptables` utility [3]. `Iptables` gives us the functionality of packet filtering. This is done on the network layer. The command to use to drop packets arriving from a specific machine is:

```
#iptables A INPUT -m mac --mac-source aa:aa:aa:aa:aa:aa j DROP
```

where `aa:aa:aa:aa:aa:aa` is the MAC address of the node of which messages should be dropped. To see the list of MAC addresses that are blocked, type: `iptables -L`, for help `iptables --help`. In our example, on machine `aa:aa:aa:aa:aa:aa`, we have to execute the following command:

```
#iptables A INPUT -m mac --mac-source cc:cc:cc:cc:cc:cc j DROP
```

whereas on machine `cc:cc:cc:cc:cc:cc` we should execute:

```
#iptables A INPUT -m mac --mac-source aa:aa:aa:aa:aa:aa j DROP
```

In this way, all the packets sent by machine `aa:aa:aa:aa:aa:aa` (`cc:cc:cc:cc:cc:cc`) will be dropped at the MAC layer on machine `cc:cc:cc:cc:cc:cc` (`aa:aa:aa:aa:aa:aa`). Since AODV operates at the network layer, we effectively simulate the situation in which the two machines cannot hear each other. As both machines `aa:aa:aa:aa:aa:aa` and `cc:cc:cc:cc:cc:cc` hear machine `bb:bb:bb:bb:bb:bb`, we can use this machine as a forwarding node.

TASK 2: 1,2 and 3-hop communication

1. Set up an FTP session between two arbitrary machines from your ad hoc network over a single hop (you do not need to use the `iptables` utility here). Download the `ws-test` file from the selected FTP server and observe the achieved throughput. If the size of `ws-test` file is not sufficient, you can generate a fresh one by running the following script in `/home/ftp/pub` directory:

```
#!/file-gen
```

Note down the observed throughput.

2. Set up an FTP session between two arbitrary machines from your ad hoc network over 2-hops (use the `iptables` utility as instructed in the example above). Convince yourself that the communication between the FTP server and the FTP client goes indeed over 2-hops. For this you may consider using the `traceroute` utility or you can simply disconnect the forwarding machine and check if there is still some traffic between the server and the client. Download the `ws-test` file from the selected FTP server and observe the achieved throughput. Note down the observed throughput.
3. Set up an FTP session between two arbitrary machines from your ad hoc network over 3-hops (make sure that this is indeed the case). Download `ws-test` file from the selected FTP server and observe the achieved throughput. Note down the observed throughput.

Compare the throughputs obtained from the above tests and try to make some conclusions on how the throughput (capacity) scales with the number of hops in IEEE 802.11b multihop networks.

IMPORTANT: Since only one station can transmit at a time on a common radio channel, it may seem that our “forced” multihop communication greatly underestimates the throughput achievable in real multihop scenarios (due to the space diversity; nodes that cannot hear each other can transmit simultaneously). However, we claim that the simulations we are using here match well real multihop scenarios. Can you say why?

TASK 3: Route re-establishment

1. Set up an FTP session between two arbitrary machines from your ad hoc network over 2-hops. Make both remaining machines forwarding nodes (simply start AODV daemon on them). Start to download `ws-test` file from the selected FTP server.
2. On the FTP client run:

```
#traceroute 192.168.100._
```

where `192.168.100._` is the IP address of the FTP server. From the output produced by the above command, retrieve the IP address of the current forwarding machine.

3. Disconnect the forwarding machine (e.g., kill the AODV process (CTRL+C) on it) whose IP is retrieved in the previous step. Observe how AODV redirects all the traffic through the other forwarding node. Try to estimate roughly the time it takes for this to happen. Has the established FTP session timed out before a new route is acquired?

Frequently Asked Questions (FAQ) / Troubleshooting

Q: The network in ad hoc mode is not working.

A-1: Check out if you have a FIXED data rate.

A-2: Check out if you are using the right/same FREQUENCY at all the stations.

Q: The iptables command is not working.

A: Check out if any word is missing. A common mistake is to miss one of the `-m mac --mac-source`.

Q: How to delete an entry from the iptables ?

```
#iptables -D INPUT 1
```

(to delete entry 1)

WARNING: In order to illustrate security aspects, these demos describe some techniques aiming at subverting the normal behavior of a network. Please note that the usage of these techniques outside of the scope of the practical exercises is strictly prohibited.

8 DEMO: Monitoring IEEE 802.11 traffic and cheating with IEEE 802.11

This exercise is a demo. In the first part of this exercise we aim to monitor the interframe spacings and the frame exchanges on the MAC layer. We use Orinoco 11a/b/g ComboCards along with their linux driver *madwifi*. We made this choice for configurability and monitoring convenience and flexibility. Two machines will be configured in monitor mode and station mode respectively.

In the second part, we will see how easy is to cheat with the IEEE 802.11 backoff mechanism. We start by transmitting from 2 different stations that use the original driver. The monitor is used as in the previous exercise. We then compute the average observed backoff of each station as well as its throughput. We proceed to loading a modified driver module into one of the stations. The modified modules “cheat” with the backoff values. Again, we compute the average observed backoff of each station as well as its throughput.

TASK 1: Monitoring the MAC layer

Configure the monitor station:

1. Load the monitoring driver:

```
cd PROXIM
make monitor
```

`make monitor` loads monitor-driver modules in the card. The driver has been changed to support monitor mode.

2. Configure the wireless card:

```
iwconfig ath0 essid "ws" channel 5
```

3. Configure and bring the interface up

```
ifconfig ath0 10.1.1.x up
```

where `x` is, for instance, the bench number.

4. Run Ethereal

Configure the transmitting station:

1. Load the monitoring driver:

```
cd PROXIM
make nocheat
```

`make nocheat` loads the original driver modules in the card.

2. Configure the wireless card:

```
iwconfig ath0 mode ad-hoc essid "ws" channel 5 rate 11M
```

3. Configure and bring the interface up

```
ifconfig ath0 10.1.1.x up
```

where `x` is, for instance, the bench number.

4. Use `mgen` to generate traffic

```
cd ../MGEN
./mgen input myex.mgn
```

the input file `myex.mgn` contains the traffic configuration, e.g.:

```
0.0 ON 1 UDP SRC 5011 DST 10.1.1.y/5010 PERIODIC [1500 200]
```

at 0.0 (immediately), turn traffic `ON`, sending UDP packets, using port 5011, to destination 10.1.1.y, port 5010, `PERIODIC`ally. Packet size is 1500, sending rate is 200 packet/s.

5. Observe the monitor output.
6. Now enable RTS/CTS mode

```
ifconfig ath0 rts 1
```

7. Observe the monitor output.
8. Figure out how to compute the backoffs ?

TASK 1: Cheating on the MAC layer

Test 1:

1. Configure the monitor as in the previous exercise

2. Configure the 2 transmitting stations as in the previous exercise, taking into account that one sends to the other, and vice versa, without using RTS/CTS.

```
ifconfig ath0 rts off
```

3. Observe the monitor. Print the “summary” output to a “text” file.
4. Compute the average observed backoff and throughput for each transmitting station, using `awk`

```
awk -f fil-thr.awk mix_nc_nc.tr  
awk -f fil-bkf.awk mix_nc_nc.tr
```

where `mix_nc_nc.tr` is the output file from Ethereal

5. Observations ?

Test 2:

We will load one card with a “cheated” driver, the other station remains unchanged.

1. Configure the monitor as in the previous exercises
2. Configure one station as in the previous exercises
3. Unload the driver from the other station

```
cd PROXIM  
make clean
```

4. Load it with a “cheated” driver, with a fixed contention window equal to 1

```
make 1
```

5. Go to step 3 of the previous test
6. Observations ?
7. Figure out how to compute backoffs in this case ?
8. Figure out how to detect cheaters ?
9. Figure out what happens when several stations cheat ?

References

- [1] *dsniff*. <http://naughty.monkey.org/~dugsong/dsniff/>.
- [2] *Implementation of AODV*. <http://user.it.uu.se/~henrikl/aodv/index.shtml>.
- [3] *Iptables*. <http://www.netfilter.org/>.
- [4] LAN/MAN Standards Committee. *ANSI/IEEE Std 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, 1999.
- [5] C. Perkins and E. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *Proceedings of IEEE WMCSA*, 1999.
- [6] C. Perkins, E. Royer, and S. Das. *Ad hoc On-Demand Distance Vector Routing*. IETF MANET Draft, 2002.