ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

LCA

# Hands-On Exercises: IEEE 802.11 Standard

Mohammad Hossein Manshaei and Jean-Pierre Hubaux
{hossein.manshaei,jean-pierre.hubaux}@epfl.ch

*Laboratory for Computer Communications and Applications (LCA)*

March 2009, Lausanne, Switzerland

# Contents

# 1 Introduction to IEEE 802.11 Standard

In recent years, high-speed *wireless local area networks* (WLANs) have become widely popular in various sectors, including health care, manufacturing, and academic centers. These sectors benefited from the productivity gains of using hand-held terminals and notebook computers to transmit real-time information within physically distributed environments. Currently, IEEE 802.11 is the de facto standard for WLANs [9]. It specifies both the *medium access control* (MAC) and the *physical* (PHY) layers for WLANs. The scope of IEEE 802.11 *working groups* (WGs) is to propose and develop MAC and PHY layer specifications for WLAN to handle mobile as well as portable stations. A portable station is one that is moved from location to location, but that is only used while at a fixed location (e.g., our stations in INF019). Mobile stations actually access the LAN while in motion.

In this standard, the MAC layer operates on top of one of several physical layers. Medium access is performed using *carrier sense multiple access with collision avoidance* (CSMA/CA). The increasing number of wireless users and the demand for high-bandwidth multimedia applications over WLANs led the IEEE working groups to provide powerful physical layers and to extend the MAC layer to provide QoS support (i.e., IEEE 802.11e [1]).

Concerning the physical layer, four IEEE 802.11 standards are currently available: 802.11a, 802.11b, 802.11g, 802.11n. The 802.11b standard is the first widely deployed WLANs [5]. Since the end of 2001, higher data rate products based on the 802.11a standard have appeared in the market [4]. More recently, the IEEE 802.11 working group has approved the 802.11g standard, which extends the data rate of the IEEE 802.11b to 54 Mbps [6]. The 802.11g specification offers transmission over relatively short distances at up to 54 Mbps. The 802.11g PHY layer employs all available modulations specified for 802.11a/b. The IEEE 802.11n significantly improves network throughput over previous standards, with a significant increase in the maximum raw (PHY) data rate from 54 Mbps to a maximum of 600 Mbps.

In general, the wireless networking can be implemented in two significantly different operating modes: *infrastructure* and *ad hoc* modes. The *infrastructure* mode consists of an *access point* (AP) acting as a hub for the network with each client communicating through it. This mode is generally for larger networks which may include sub-networks consisting of more than one access point. This means that an infrastructure network is more expensive to setup and usually requires more advanced configuration.

*Ad-hoc* mode essentially eliminates the need for an access point. In this mode, the mobile nodes can be connected dynamically in an arbitrary manner. All nodes of these networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. In other words, routing from one node to another requires an on-demand routing protocol, like DSR [3], AODV [7], or OLSR [2].

To evaluate the performance of IEEE 802.11 devices, a complete knowledge about the functionalities of these MAC and PHY layer protocols is required. Hereafter, we overview the salient features of the IEEE 802.11b MAC and PHY layers.

## 2 IEEE 802.11 MAC Layer

The *distributed coordination function* (DCF) is the basic medium access mechanism of IEEE 802.11, and uses a *carrier sense multiple access with collision avoidance* (CSMA/CA) algorithm to mediate the access to the shared medium. On the other hand, the *point coordination function* (PCF) is a centralized, polling-based access mechanism which requires the presence of a base station that acts as an access point. Here we focus on DCF protocol.

The DCF protocol in IEEE 802.11 standard defines how the medium is shared among stations. DCF is based on CSMA/CA [9]. It includes a basic access method and an optional channel access method with *request-to-send* (RTS) and *clear-to-send* (CTS) exchanged as shown in Figure 1 and 2 respectively. First, we explain the basic access method.

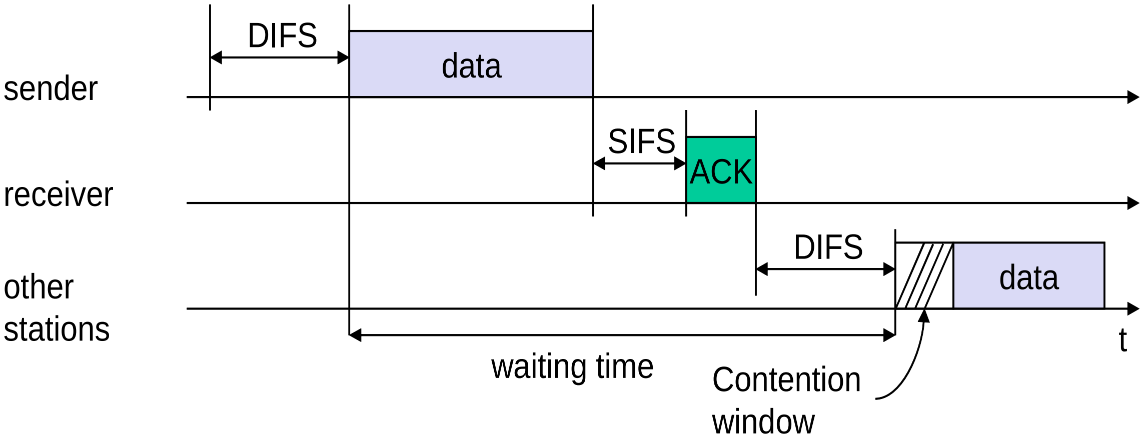


Figure 1: Basic access CSMA/CA protocol in DCF.

If the channel is busy for the source STA, a backoff time (measured in slot times) is chosen randomly in the interval $[0, CW)$, where $CW$ is called the *contention window*. The slot time is the sum of the RX-to-TX turnaround time, MAC processing delay, and CCA detect time [9]. The value of slot time for different PHY layer protocols is shown in Table 1.

This timer is decremented by one as long as the channel is sensed idle for a DIFS, i.e., *distributed inter frame space* time. DIFS is equal to $SIFS + 2 \times SlotTime$. It stops when the channel is busy and resumes when the channel is idle again for at least a DIFS period. $CW$ is an integer whose range is determined by the PHY layer characteristics: $CW_{min}$ and $CW_{max}$. $CW$ is doubled after each unsuccessful transmission, up to the maximum value which is determined by $CW_{max} + 1$.

When the backoff timer reaches zero, the source transmits the data packet. The ACK is transmitted by the receiver immediately after a period of time called SIFS, i.e., *short inter frame space* time which is less than DIFS. When a data packet is transmitted, all

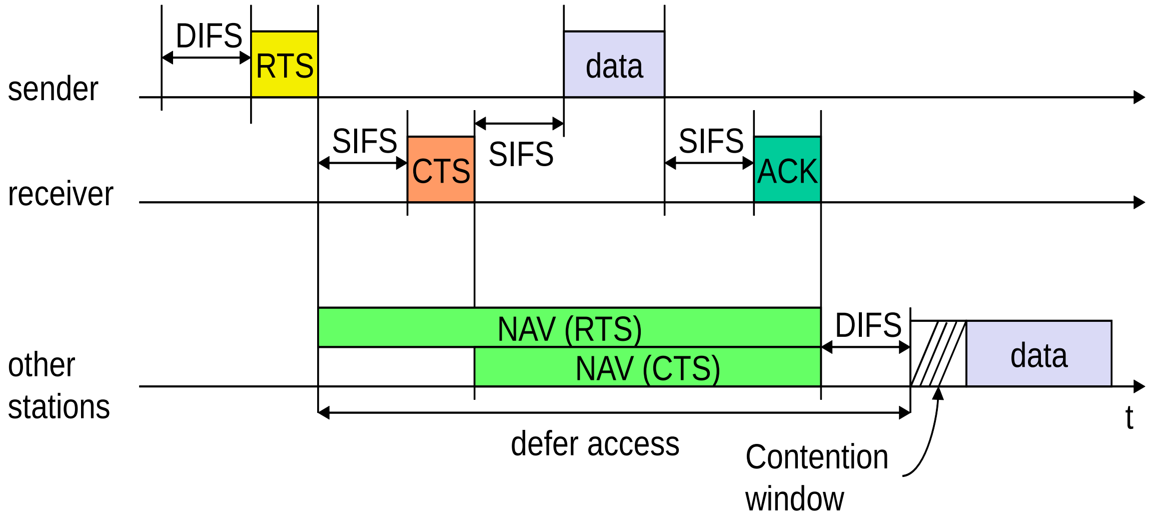


Figure 2: RTS/CTS exchange in CSMA/CA protocol.

other stations hearing this transmission adjust their *network allocation vector* (NAV), which is used for virtual *carrier sense* (CS) at the MAC layer. The NAV maintains a prediction of future traffic on the medium based on the duration information that is announced in Data frames (or RTS/CTS frames as will be explained in the following) prior to the actual exchange of data. In addition, whenever a node detects an erroneous frame, the node defers its transmission by a fixed duration indicated by EIFS, i.e., *extended inter frame space* time. This time is equal to the $SIFS + ACK_{time} + DIFS$ time.

If the optional access method is used, an RTS frame should be transmitted by the source and the destination should accept the data transmission by sending a CTS frame prior to the transmission of the actual data packet, as shown in Figure 2.

Table 1: Inter frame space and $CW$ time for different PHY layers.

| *Parameters* | *802.11a* | *802.11b (FH)* | *802.11b (DS)* | *802.11b (IR)* | *802.11b (High Rate)* |
|---|---|---|---|---|---|
| *Slot Time ($\mu s$)* | 9 | 50 | 20 | 8 | 20 |
| *SIFS ($\mu s$)* | 16 | 28 | 10 | 10 | 10 |
| *DIFS ($\mu s$)* | 34 | 128 | 50 | 26 | 50 |
| *EIFS ($\mu s$)* | 92.6 | 396 | 364 | 205 or 193 | 268 or 364 |
| $CW_{min}(SlotTime)$ | 15 | 15 | 31 | 63 | 31 |
| $CW_{max}(SlotTime)$ | 1023 | 1023 | 1023 | 1023 | 1023 |

Note that STAs in the sender's range that hear the RTS packet should update their

NAVs and defer their transmissions for the duration specified by the RTS. Nodes that overhear the CTS packet update their NAVs and refrain from transmitting. This way, the transmission of the data packet and its corresponding ACK can proceed without interference from other nodes (hidden nodes problem). Table 1 shows the important time interval between frames in different standard specification called *inter frame space* (IFS) [4, 5, 6]. It should be considered that the IEEE 802.11g uses one of the IFS set based on its operating mode.

# 3 IEEE 802.11 Physical Layer Characteristics

Table 2 shows three different PHY layers that are available for the IEEE 802.11 WLAN [4][5][6]. IEEE 802.11b radios transmit at 2.4 GHz and send data up to 11 Mbps using *direct sequence spread spectrum* (DSSS), *infrared* (IR), and *frequency hopping* (FH) [5]; whereas IEEE 802.11a radios transmit at 5 GHz and send data up to 54 Mbps using *orthogonal frequency division multiplexing* (OFDM) [4]. The IEEE 802.11g standard [6], extends the data rate of the IEEE 802.11b to 54 Mbps in an upgraded PHY layer named *extended rate* PHY layer (ERP).
Different PHY transmission modes are defined with different modulation schemes, and coding rates. The performance of the modulation schemes can be measured by their robustness against path loss, interferences and fading that causes variations in the received SNR. Such variations also cause variations in the BER, since the higher the SNR, the easier it is to demodulate and decode the received bits.

Table 2: Characteristics of the various physical layers in the IEEE 802.11 standard.

| *Characteristic* | *802.11a* | *802.11b* | *802.11g* |
|---|---|---|---|
| *Frequency* | 5 GHz | 2.4 GHz | 2.4 GHz |
| *Rate (Mbps)* | 6, 9, 12, 18, 24, 36, 48, 54 | 1, 2, 5.5, 11 | 1, 2, 5.5, 6, 9, 11, 12, 18 |
| | | | 22, 24, 33, 36, 48, 54 |
| *Modulation* | BPSK, QPSK, 16 QAM | DBPSK, DQPSK, CCK | BPSK, DBPSK, QPSK, DQPSK, CCK |
| | 64 QAM (OFDM) | (DSSS, IR, and FH) | 16 QAM, 64 QAM (OFDM and DSSS) |
| *FEC Rate* | 1/2, 2/3, 3/4 | NA | 1/2, 2/3, 3/4 |
| *Basic Rate* | 6 Mbps | 1 or 2 Mbps | 1,2, or 6 Mbps |

In each physical layer, there is a basic transmission mode (usually used to send ACK, RTS, CTS and PLCP header[1]) which has the maximum coverage range among all transmission modes.
This maximum range is obtained using BPSK or DBPSK modulations which have the minimum probability of bit error for a given SNR compared to other modulation schemes. The basic rates have the minimum data rate as well. The basic transmission rates for different standards are shown in Table 2. For instance, the basic rate is 1 Mbps (with DBPSK modulation and CRC 16 bits) for 802.11b and 6 Mbps (with BPSK and FEC rate equal to 1/2) for 802.11a.
As shown in Figure 3, each packet is sent with two different rates [9]: its PLCP header is sent at the basic rate while the rest of the packet might be sent at a higher rate.

[1] Note that the AP can define a set of data transfer rates, called basic rate set, which all the stations in a BSS need to be capable of using to receive and transmit frames to/from the wireless medium. These rates can be used to send control frames but the PLCP header should always be sent with the basic rates specified in Table 2.

The higher rate, used to transmit the physical-layer payload, is stored in the PLCP header. The receiver can verify that the PLCP header is correct (using CRC or Viterbi decoding with parity), and uses the transmission mode specified in the PLCP header to decode the MAC header and payload.

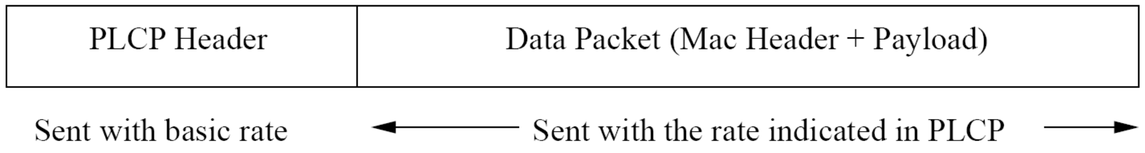

PLCP Header
Data Packet (Mac Header + Payload)
Sent with basic rate
Sent with the rate indicated in PLCP

Figure 3: Data rates for packet transmission.

Table 2 also shows that the IEEE 802.11 standard defines four PHY layer transmission techniques to send data over wireless channel: *direct sequence spread spectrum* (DSSS), *frequency hopping spread spectrum* (FH), *infrared* (IR), and *orthogonal frequency division multiplexing* (OFDM). Since in this exercise we use 802.11b devices we will focus hereafter on 802.11b data transmission.

# 4 IEEE 802.11b Data Transmission

The first standard specification of 802.11 WLAN defined a DSSS system that provides a wireless LAN with both 1 and 2 Mbps data payload communication capability. The DSSS system uses baseband modulations of *differential binary phase shift keying* (DBPSK) and *differential quadrature phase shift keying* (DQPSK) to provide the 1 Mbps and 2 Mbps data rates, respectively [9]. In 1999, the higher-speed physical layer extension of WLAN proposed to use CCK modulation to provide higher speeds in the 2.4 GHz Band. These high rates are based on the CCK modulation scheme for 5.5 Mbps and 11 Mbps. An optional PBCC mode is also provided to potentially enhanced performance.
Table 3 shows all the available transmission modes in IEEE 802.11b WLANs. Following we will address the procedure of data transmission for different modulations (i.e., DBPSK, DQPSK, and CCK).

Table 3: Transmission modes in IEEE 802.11b.

| *Mode* | *Modulation* | *Data Rate (Mbps)* | *FEC Rate* |
|---|---|---|---|
| 1 | DBPSK | 1 | NA |
| 2 | DQPSK | 2 | NA |
| 3 | CCK/PBCC | 5.5 | NA / 1/2 |
| 4 | CCK/PBCC | 11 | NA / 1/2 |

## 4.1 DBPSK and DQPSK Modulations

According to the standard specification, the transmitted signal for 1 and 2 Mbps data rates is differentially encoded and modulated by BPSK and QPSK for 1 and 2 Mbps respectively. Tables 4 and 5 show the Differentially BPSK and QPSK encoder respectively [9]. The receiver can detect the signal coherently or differentially. In the latter case, it is not necessary to lock and track the carrier phase precisely. If the signal is coherently detected, we denote the modulations as *differentially encoded*, i.e., DE-BPSK and DE-QPSK. In the second case, we denote the modulations as DBPSK and DQPSK. Both cases could be implemented at the receiver.

Table 4: 1 Mbps DE-BPSK encoding table.

| *Bit input* | *Phase change* $(+j\omega)$ |
|---|---|
| 0 | 0 |
| 1 | $\pi$ |

Table 5: 2 Mbps DE-QPSK encoding table.

| *Dibit input* | *Phase change (+$j\omega$)* |
|---|---|
| 00 | $0$ |
| 01 | $\pi/2$ |
| 11 | $\pi$ |
| 10 | $3\pi/2(-\pi/2)$ |

## 4.2 CCK Modulation

The high speed extension of the IEEE 802.11 standard specifies *Complementary Code Keying* (CCK) as the modulation scheme for 5.5 and 11Mbps data rates in the 2.4 GHz band [5]. The length 8 complementary codes which are used in 802.11b, can be written as a function of four phase elements $\phi_1$, $\phi_2$, $\phi_3$, and $\phi_4$ by:

$$\begin{aligned} C(\phi_1,\phi_2,\phi_3,\phi_4) &= [e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)}, -e^{j(\phi_1+\phi_4)}, \\ &\quad e^{j(\phi_1+\phi_2+\phi_3)}, e^{j(\phi_1+\phi_3)}, -e^{j(\phi_1+\phi_2)}, e^{j(\phi_1)}] \qquad (1) \end{aligned}$$

For example, to generate the $2^8 = 256$ codewords needed to transmit data at 11 Mbps from this expression, the four phase parameters are each allowed to take one of the four values 0, $\pi/2$, $\pi$, $3\pi/2$. This is similar to allowing each phase to be drawn from a QPSK constellation. In order to understand the mathematical representation of CCK Modulation, it is useful to show how a code is generated in CCK for 11 Mbps data rate. A signal in CCK Modulation starts out as an eight-bit binary word $\mathbf{D} = d_7d_6d_5d_4d_3d_2d_1d_0$. The 8 bits are used to encode the phase parameters (i.e., the $\phi_1$ to $\phi_4$). The encoding is based on the differential QPSK modulation. The first dibit ($d_0$, $d_1$) encodes $\phi_1$ based on the DQPSK specified in Table 5. Then, the data dibits ($d_2$, $d_3$), ($d_4$, $d_5$), and ($d_6$, $d_7$) encode $\phi_2$, $\phi_3$, and $\phi_4$ respectively, based on QPSK as specified in Table 6. Note that this table is binary (not Grey) coded. For example, for a data stream given as 01100011, we get from Table 5, 6, and 7: $d_1d_0 = 11$, $\phi_1 = \pi$, $d_3d_2 = 00$, $\phi_2 = 0$, $d_5d_4 = 10$, $\phi_3 = 3\pi/2$, $d_7d_6 = 01$, $\phi_4 = \pi/2$. Finally, using the Equation (1), we can find out the codes which should be sent from all possible codes.

## 4.3 Direct Sequence Spread Spectrum in IEEE 802.11b

In *code division multiple access* (CDMA) systems, all users transmit in the same frequency band simultaneously. Communication systems following this concept are called *spread spectrum* (SS) systems. In this transmission technique, the frequency spectrum of a data-signal is spread using a code uncorrelated with that signal. As a result the bandwidth occupancy is much higher than required. The codes used for spreading have

Table 6: QPSK encoding table for CCK 11 Mbps.

| *Dibit input* | *Phase change* $(+j\omega)$ |
|---|---|
| 00 | $0$ |
| 01 | $\pi/2$ |
| 10 | $\pi$ |
| 11 | $3\pi/2(-\pi/2)$ |

Table 7: Phase parameter encoding scheme.

| *Dibits* | *Phase Parameter* |
|---|---|
| $(d_1, d_0)$ | $\phi_1$ |
| $(d_3, d_2)$ | $\phi_2$ |
| $(d_5, d_4)$ | $\phi_3$ |
| $(d_7, d_6)$ | $\phi_4$ |

low cross-correlation values and are unique to every user. A receiver which has knowledge about the code of the intended transmitter, is capable of selecting the desired signal.

The SS techniques were first used in the military field, because of the difficulty to jam or detect spread spectrum signals. However, nowadays, spread spectrum systems are gaining popularity also in commercial applications. There exist different techniques to spread a signal like *direct sequence* (DS), *frequency hopping* (FH), *time hopping* (TH), and *multi carrier* CDMA (MC-CDMA). It is also possible to combine them in a single system.

DS is the best known SS technique. In this technique, the data signal is multiplied by a *pseudo random noise* (PN) code. A PN code is a sequence of $-1$ and 1 (polar) or 0 and 1 (non-polar) with an specific period named chip period. The following $11-$*chip* Barker sequence code shall be used as the PN code sequence in the IEEE 802.11 standard: $+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1$ [9].

A PN code has noise-like properties. This results in low cross-correlation values among the codes and the difficulty to jam or detect a data message.

In 802.11b DSSS, each information bit is combined via an XOR function with a PN sequence as shown in Figure 4. The result is a high speed digital stream which is then modulated. As shown in Figure 4, the effect of the PN code sequence is to spread the transmitted bandwidth of the resulting signal by a ratio of 11 : 1 (i.e., *spread spectrum*).

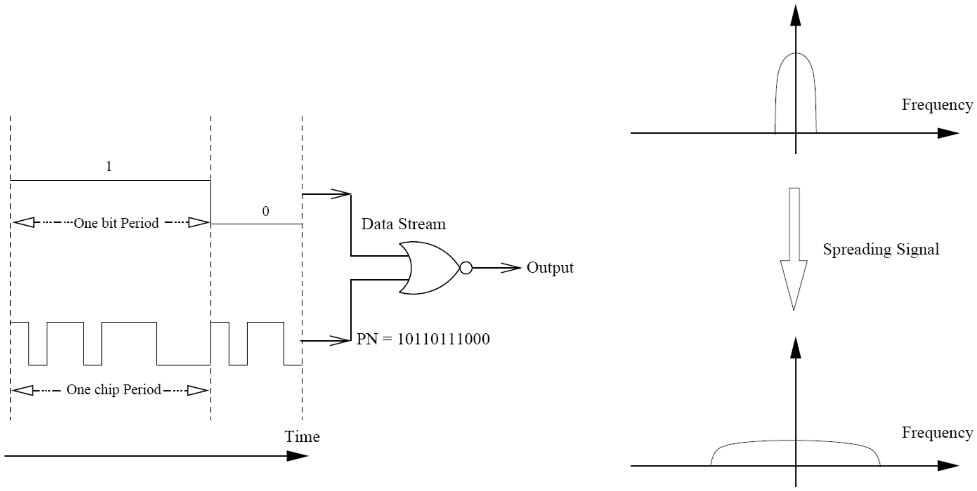


Figure 4: Using PN code in 802.11b to spread the signal.

## 4.4 IEEE 802.11b Channel Allocation

IEEE 802.11b defines 14 partially overlapping channels. As shown in Figure 5, these channels are defined in the frequency range of 2.4 GHz to 2.495 GHz. Any two channels are non-overlapping if and only if they are separated by four or more channels. In particular, the set of channels 1, 6, and 11 is the only set of three non-overlapping channels. Our access point in IEW works on channels 6. This is to prevent interference with the EPFL operational network (channels 1 and 11 for epfl and public-epfl access points).

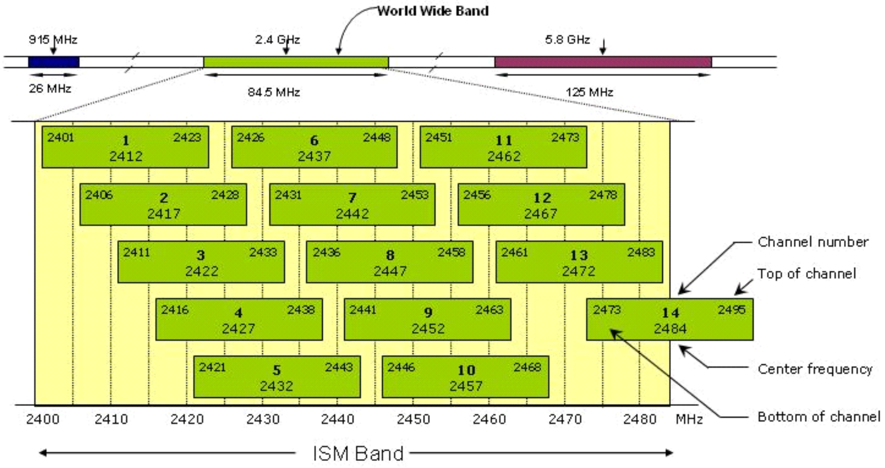


Figure 5: IEEE 802.11b channel specification.

# 5 IEEE 802.11 Frame Format

In the IEEE 802.11 MAC layer, each MPDU packet consists of the following basic components: a MAC header, IP/UDP or TCP headers, a variable length information frame body, and a *frame check sequence* (FCS). All the fields except the frame body, which is 28 octets in total, contribute to the MAC overhead for a data/fragment frame. The format of the MAC header with the exception of FCS is shown in Figure 6[2]. The frame format of RTS, CTS, and ACK packets are shown in Figure 7.

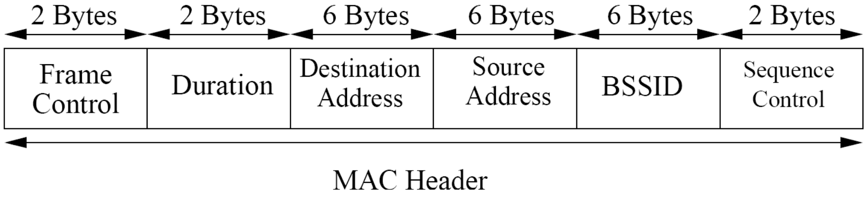
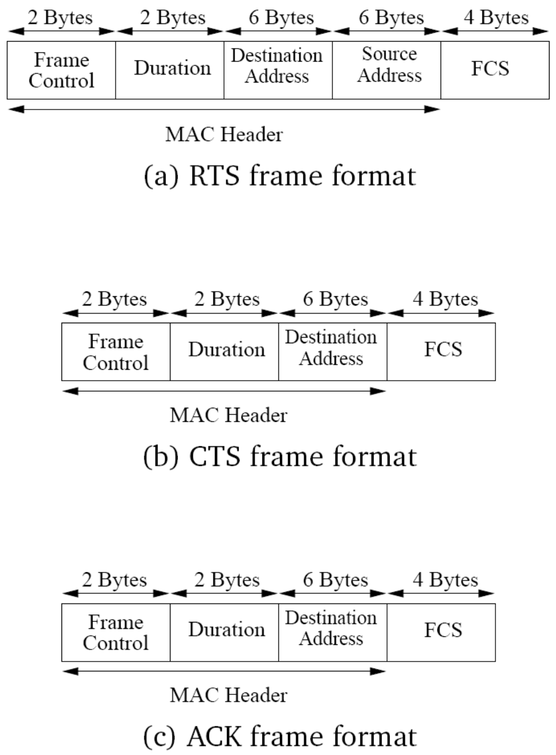


Figure 6: MAC header format in IEEE 802.11

There are two different PLCP frame formats in IEEE 802.11b: Long and Short PLCP as shown in Figure 8. The long PLCP including the High Rate PLCP preamble and the High Rate PLCP header. The PLCP preamble contains the two following fields: *synchronization* (SYNC) and *start frame delimiter* (SFD). The PLCP header contains the four following fields: SIGNAL, SERVICE, LENGTH, and CCITT CRC-16 (CRC). Each of these fields is described in detail in the standard. The PLCP header and preamble must be sent using the basic mode corresponding to 1 Mbps and DBPSK modulation in 802.11b. Note that the short frame format is not compatible with the PPDUs used in the classic DSSS PHY layer. The short PLCP header uses the 2 Mbps with DQPSK modulation and a transmitter using the short PLCP can only interoperate with the receivers which are capable of receiving this short PLCP format. The short PLCP preamble and header may be used to minimize overhead and thus maximize the network data throughput.

[2] There is another address field named *Address4* in MAC header which is assigned for *wireless distribution system* (WDS) frames being distributed from one AP to another AP. This field is omitted when it is not applicable (N/A).

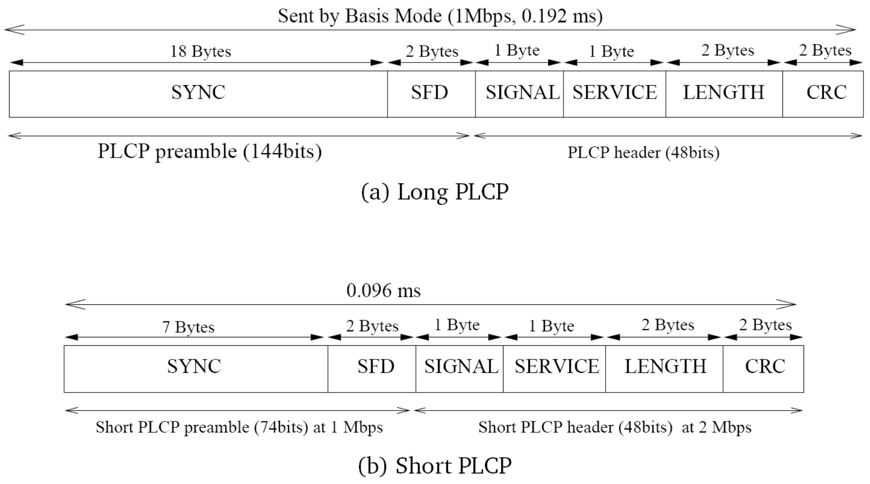


(a) RTS frame format

(b) CTS frame format

(c) ACK frame format

Figure 7: Control frames in IEEE 802.11 standard.



(a) Long PLCP

(b) Short PLCP

Figure 8: (a) Long and (b) short PLCP header format in 802.11b.

# 6 Infrastructure Mode Experiments

Our goal in this exercise is to set up an operational Wireless LAN and evaluate its performance. In our exercises, we will use D-Link AirPlus$^{TM}$G DWL-G122 Wireless USB Adapter, shown in Figure 9. The adaptor is fully cmpatible with 802.11b and 802.11g. 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.

Figure 9: D-Link AirPlus$^{TM}$G DWL-G122 Wireless USB Adapter that support IEEE 802.11b/g protocol.

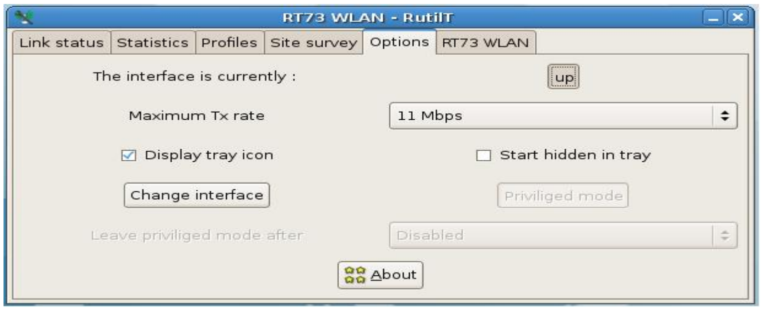
Please note that, once plugged in, the wireless adapters are ready to use, since the driver and the client utility program named *RutilT* are already installed.
Before we proceed, let us just remind that each bench (BANC) comprises two machines, one labeled *Router* and the other one *Station*. Please note that in our exercises there is no difference between the two. Do the following on both, Stations and Routers.

Task 1: **Minimal configuration**

1. Log on to a machine and start a terminal (*xterm*).
2. Remove any Ethernet cabling.
3. Plug in the wireless adapter (USB interface). Upon insertion you should be able to run *RutilT* program.

## TASK 2: Basic settings

1. Run the *RutilT* program and modify the *Options* and *RT73 WLAN* tabs, as shown in Figure 10 and Figure 11:



The interface is currently : up

Maximum Tx rate 11 Mbps

☑ Display tray icon

□ Start hidden in tray

Leave priviliged mode after Disabled

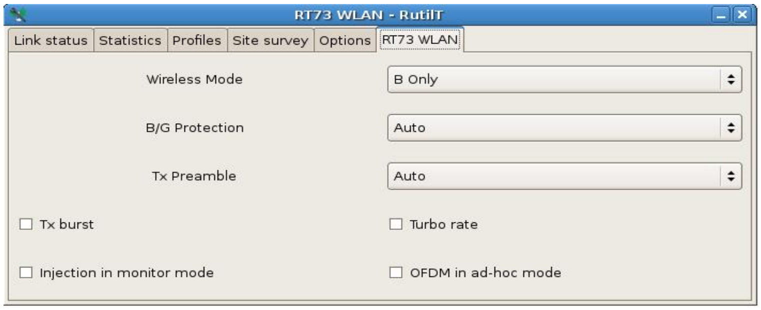
Figure 10: Activate the adapter (*up*) and select the transmission rate with *RutilT*.

   (a) Activate the interface (i.e., *up*).

   (b) Set the Maximum transmission rate to 11 Mbps.

   (c) Set Wireless Mode to *B Only*.

   (d) Set Tx Preamble to *Auto*.



Wireless Mode B Only

B/G Protection Auto

Tx Preamble Auto

□ Tx burst

□ Turbo rate

□ Injection in monitor mode

□ OFDM in ad-hoc mode

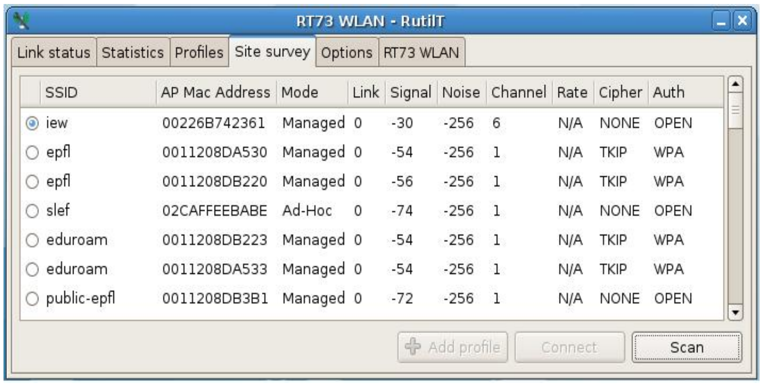
Figure 11: Select the IEEE 802.11b operational mode with *RutilT*.

### 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 all available frequency channels to find an Access Point. Thus, we do not have to set the channel by ourselves. Recall that our access point works on channel 6.

1. Start *RutilT*.

2. In Commands menu of *RutilT* select *Site Survey* tab and perform a search by clicking on scan, as shown in Figure 12. Our access point SSID is *iew*.



| | SSID | AP Mac Address | Mode | Link | Signal | Noise | Channel | Rate | Cipher | Auth |
|---|---|---|---|---|---|---|---|---|---|---|
| ● | iew | 00226B742361 | Managed | 0 | -30 | -256 | 6 | N/A | NONE | OPEN |
| ○ | epfl | 0011208DA530 | Managed | 0 | -54 | -256 | 1 | N/A | TKIP | WPA |
| ○ | epfl | 0011208DB220 | Managed | 0 | -56 | -256 | 1 | N/A | TKIP | WPA |
| ○ | slef | 02CAFFEEBABE | Ad-Hoc | 0 | -74 | -256 | 1 | N/A | NONE | OPEN |
| ○ | eduroam | 0011208DB223 | Managed | 0 | -54 | -256 | 1 | N/A | TKIP | WPA |
| ○ | eduroam | 0011208DA533 | Managed | 0 | -54 | -256 | 1 | N/A | TKIP | WPA |
| ○ | public-epfl | 0011208DB3B1 | Managed | 0 | -72 | -256 | 1 | N/A | NONE | OPEN |

Figure 12: Scan all available channels in 802.11b and find the available access points in the range.

3. Create a profile to connect to *iew* access point, as shown in Figure 13. Fill out the text fields in the window as follows:

   **Name -** It is the name of profile. You can put whatever in this field, but in this exercise we can use the following naming policy; *AP-station[bench No.]* for the Station and *AP-router[bench No.]* for the Router (e.g., in the case of bench 12, *AP-station12* and *AP-router12*).

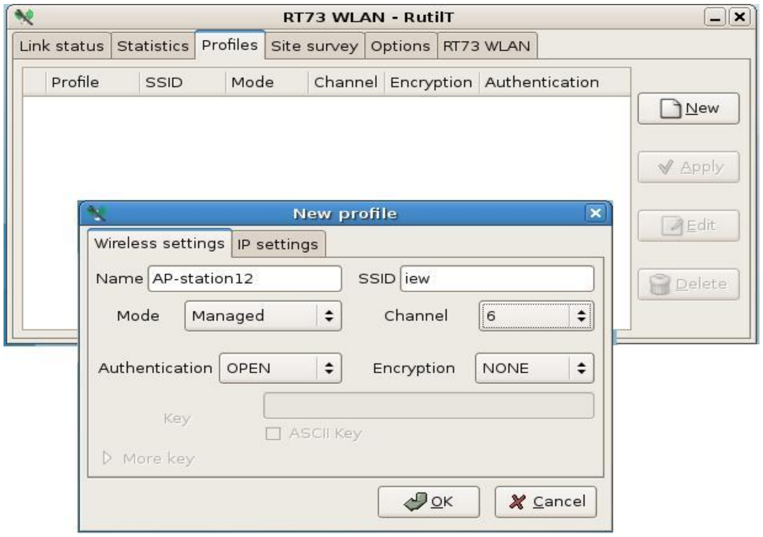
   **SSID -** Service Set ID (SSID) is a unique identifier that client devices use to associate with either AP or other client. This value MUST match the SSID of an access point that we want to communicate with. In our case, the SSID is `iew`. Do not forget that the SSID is case sensitive.

   **Mode -** Here we select *Managed* that corresponds to infrastructure mode.

   **Authentication -** It is *OPEN* as our access point is open.

   **Encryption -** It is *NONE*.

4. Set DHCP to assign the IP address automatically, as shown in Figure 14. To see the exchanged messages in DHCP protocol between the access point and your station, you can type `dhclient` in the command line.



Name AP-station12
SSID iew
Mode Managed
Channel 6
Authentication OPEN
Encryption NONE

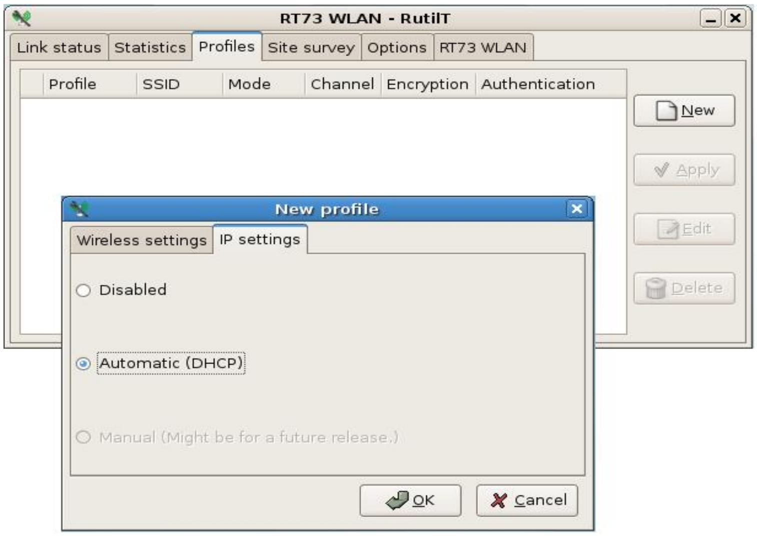
Figure 13: Profile definition for access point connection.

5. If everything works well, you can find your IP address as well as wireless link quality in Link Status tab of *RutilT*, as shown in Figure 15.

6. 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 192.168.1.1.

   The station having IP address 192.168.1.200 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 next exercise.

   IMPORTANT: Start the `wireshark` tool to monitor packets exchanged between your station and the station having IP address 192.168.1.200. From the output produced by `wireshark` retrieve the MAC address corresponding to the station with IP address 192.168.1.200. Note down this MAC address for exercises later on.

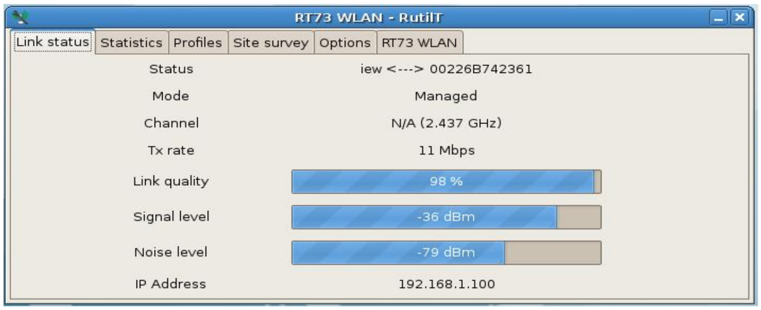


○ Disabled

● Automatic (DHCP)

○ Manual (Might be for a future release.)

Figure 14: Automatic address assignment by DHCP in Infrastructure mode.



| | |
|---|---|
| Status | iew <---> 00226B742361 |
| Mode | Managed |
| Channel | N/A (2.437 GHz) |
| Tx rate | 11 Mbps |
| Link quality | 98 % |
| Signal level | -36 dBm |
| Noise level | -79 dBm |
| IP Address | 192.168.1.100 |

Figure 15: Wireless link state (link quality, signal and noise level).

**WARNING: In order to illustrate security aspects, the next exercise describes 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.**

# 7 IEEE 802.11 Security: Password Sniffing

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. For example, the attacker, in order to sniff an IEEE 802.11 traffic, only has to be located somewhere within the communication range of possible victims (e.g., at a nearby parking lot).

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 a wireless card in monitor mode. 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 `wireshark` 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 192.168.1.200. 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 with `wireshark`?

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. With current setting you are not able to monitor the traffic from other machines in the room, but 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.

## 7.1 Channel Sniffing by Stations

You can perform the password sniffing on your stations as well, by taking the following steps:

1. Find the name of your wireless interface by `ifconfig -a`. Assume that your wireless interface is `wlan0`.

2. You have to change the mode of your wireless card to *monitor* by the following command:

   ```
   #iwconfig wlan0 mode monitor
   ```

3. Run `wireshark` and capture the data from `wlan0` interface.

4. Make sure that your interface works on channel 6 by running the following command.

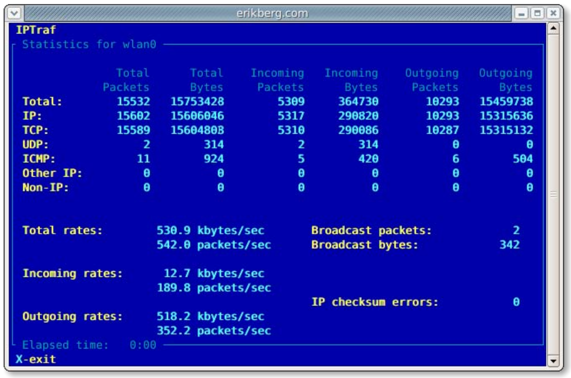
   ```
   #iwconfig wlan0 channel 6
   ```

5. Ask your neighbor station to make an FTP connection and use anonymous user id to connect to the ftp server.

6. You can filter out the pass phrase in the captured data and find out the password that your friend has used for the FTP connection.

# 8 IEEE 802.11b Throughput Evaluation

As explained in Section 4, IEEE 802.11b adapters operates at the maximum data rate of 11Mbps. Supported Data Rates for our adapter include 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. The data rate of 11 Mbps sounds fairly good (∼1MBps). Note that this is the 802.11b raw data rate at the physical layer of the network. In this exercise, we 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 named `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. Then open the "Detailed interface statistics" window for your wireless interface.



```
IPTraf
┌ Statistics for wlan0 ─────────────────────────────────────────────────────

                 Total      Total   Incoming   Incoming   Outgoing   Outgoing
               Packets      Bytes    Packets      Bytes    Packets      Bytes
 Total:          15532   15753428       5309     364730      10293   15459738
 IP:             15602   15606046       5317     290820      10293   15315636
 TCP:            15589   15604808       5310     290086      10287   15315132
 UDP:                2        314          2        314          0          0
 ICMP:              11        924          5        420          6        504
 Other IP:           0          0          0          0          0          0
 Non-IP:             0          0          0          0          0          0


 Total rates:         530.9 kbytes/sec      Broadcast packets:          2
                      542.0 packets/sec     Broadcast bytes:          342

 Incoming rates:       12.7 kbytes/sec
                      189.8 packets/sec
                                            IP checksum errors:         0
 Outgoing rates:      518.2 kbytes/sec
                      352.2 packets/sec
└ Elapsed time:   0:00 ───────────────────────────────────────────────────────
X-exit
```

Figure 16: Iptraf screen. In this example the throughput is about 0.5 Mbps.

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 192.168.1.200 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 and set maximum data rate to 11 Mbps as before.

2. Set up an FTP session between your machine and the FTP server (192.168.1.200), which is connected via a wired link to the LAN, by typing:

   ```
   #ftp  192.168.1.200
   ```

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

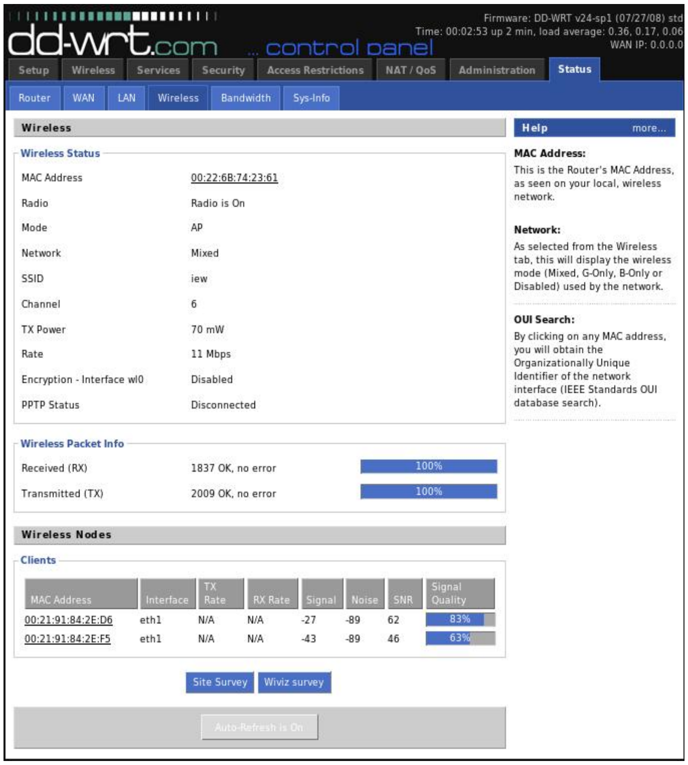
3. Before downloading `ws-test` file to the client, change the local directory the file `ws-test` will be stored to by typing:

```
ftp> cd pub
```

4. 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
```

5. Note that you can check the number of stations associated with AP and their MAC addresses by typing the IP address of AP (i.e., 192.168.1.1) in your browser, as shown in Figure 17. You can use this to evaluate the obtained throughput.



Firmware: DD-WRT v24-sp1 (07/27/08) std
Time: 00:02:53 up 2 min, load average: 0.36, 0.17, 0.06
WAN IP: 0.0.0.0

Wireless

Wireless Status

| | |
|---|---|
| MAC Address | 00:22:6B:74:23:61 |
| Radio | Radio is On |
| Mode | AP |
| Network | Mixed |
| SSID | iew |
| Channel | 6 |
| TX Power | 70 mW |
| Rate | 11 Mbps |
| Encryption - Interface wl0 | Disabled |
| PPTP Status | Disconnected |

MAC Address:
This is the Router's MAC Address, as seen on your local, wireless network.

Network:
As selected from the Wireless tab, this will display the wireless mode (Mixed, G-Only, B-Only or Disabled) used by the network.

OUI Search:
By clicking on any MAC address, you will obtain the Organizationally Unique Identifier of the network interface (IEEE Standards OUI database search).

Wireless Packet Info

| | | |
|---|---|---|
| Received (RX) | 1837 OK, no error | 100% |
| Transmitted (TX) | 2009 OK, no error | 100% |

Wireless Nodes

Clients

| MAC Address | Interface | TX Rate | RX Rate | Signal | Noise | SNR | Signal Quality |
|---|---|---|---|---|---|---|---|
| 00:21:91:84:2E:D6 | eth1 | N/A | N/A | -27 | -89 | 62 | 83% |
| 00:21:91:84:2E:F5 | eth1 | N/A | N/A | -43 | -89 | 46 | 63% |

Figure 17: Access Point summary table of the number of connected stations and the quality of the wireless links.

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?

### 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 192.168.1.201 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 (192.168.1.201), which is connected via the radio link to the LAN.
2. Start the download of the file `ws-test`, 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 and their MAC addresses.) by typing AP's IP address in your web browser.

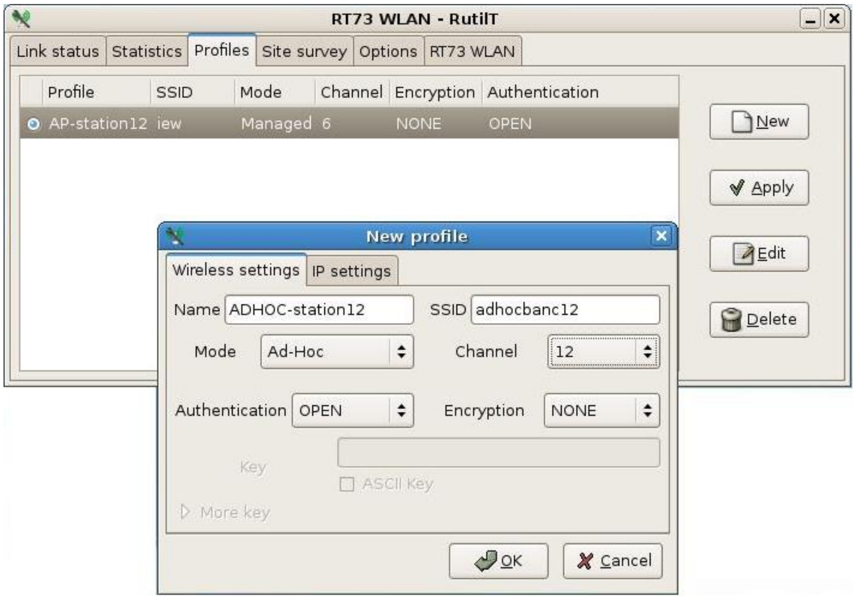
How does the obtained data rate compare with the data rate obtained in the Task 1 above? How does it compares with the 11Mbps?

# 9 IEEE 802.11 Ad Hoc Mode

In this exercise 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 RutilT select Profiles.



| Profile | SSID | Mode | Channel | Encryption | Authentication |
|---|---|---|---|---|---|
| ○ AP-station12 | iew | Managed | 6 | NONE | OPEN |

Name ADHOC-station12 SSID adhocbanc12

Mode Ad-Hoc Channel 12

Authentication OPEN Encryption NONE

Key

□ ASCII Key

▷ More key

Figure 18: Profile definition of ad hoc network by *RutilT* program.

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

**SSID -** As explained before, SSID is a unique identifier that client devices use to associate with either AP or other clients. This value MUST match the SSID of any other wireless client that you want to communicate with. In our exercises, we use *adhocbanc[bench No.]* as the SSID on both, the Station and Router (e.g. for bench 6 we use *adhocbanc6*). Note that the SSID is case sensitive.

**Network Type -** Select *Ad-Hoc.*

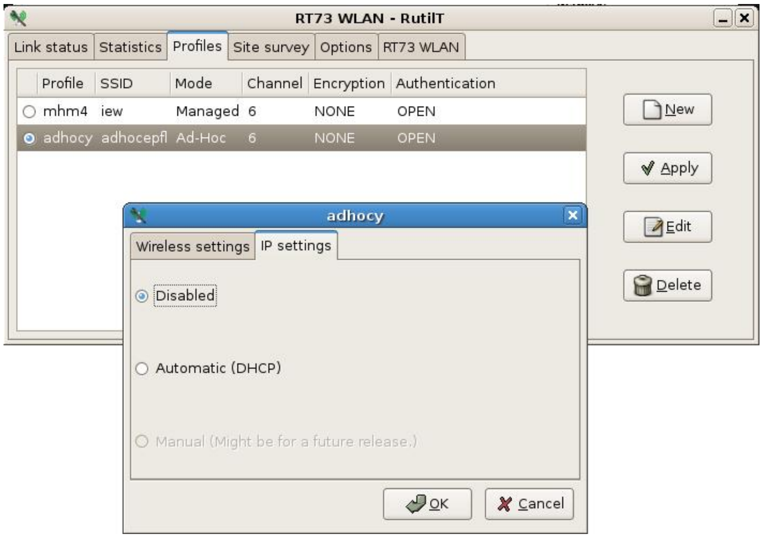
**Authentication -** Select *Open.*

**Encryption -** Select *NONE.*

**Channel -** In the Ad Hoc mode, the frequency channel must match the channel used by the other Adapters you wish to communicate with. Set the channel of your adapter according to a *channel assignment plan.* As mentioned before, you 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 use the bench number to determine the channel to be used (benches 1 and 11 should use some other values).

2. Set *Disable* for IP setting in ad hoc mode as shown in Figure 19.



| | Profile | SSID | Mode | Channel | Encryption | Authentication |
|---|---|---|---|---|---|---|
| ○ | mhm4 | iew | Managed | 6 | NONE | OPEN |
| ● | adhocy | adhocepfl | Ad-Hoc | 6 | NONE | OPEN |

adhocy

Wireless settings | IP settings

● Disabled

○ Automatic (DHCP)

○ Manual (Might be for a future release.)

Figure 19: Use the manual IP address configuration for ad hoc mode.

3. Assign an IP address to the wireless adapter by:

```
#ifconfig wlan0 10.0.0.abc netmask 255.255.255.0 up
```

where `wlan0` is the name assigned to the wireless interface and `10.0.0.abc` is the IP address assigned to the machine.[3] We use the following addressing plan to assign IP addresses to machines. All the machines use `10.0.0` 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)

4. Check the link status in *RutilT* for the status of connection.

5. 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 `/var/ftp/pub` directory:

[3] You can learn the identifier assigned to your wireless adapter by typing **ifconfig -a** at the command line.

```
#./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. Note that you can check the signal level at your adapter. For this, open the *Link status* tab in the commands menu of RutilT.

2. In this exercise we use `vsftpd` for the ftp server. Check if ftp server is up and running on your station by `netstat -a`.

3. 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  10.0.0._
   ```

   where `10.0.0._` 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.

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

   ```
   ftp> cd pub
   ```

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

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

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. 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.

3. Start to download the file `ws-test` simultaneously on both benches and observe the data rates with `iptraf` (possibly on both machines).

4. Check the Signal level in the *Link status* window. How does it compare with the *no*-interference scenario? Note that it may happen that you get no difference with respect to the Task 1, since we have 28 transmitters collocated at the same site. In any case, assuming existence of only your bench and the selected neighboring bench, in which scenario (Task 1 or Task 2) do you expect higher interference level? Why?

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

# 10 Routing in Mobile Ad Hoc Networks: AODV

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 [7] 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 [8] 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. Start the AODV process as follows:

```
#aodvd -R
```

4. 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.

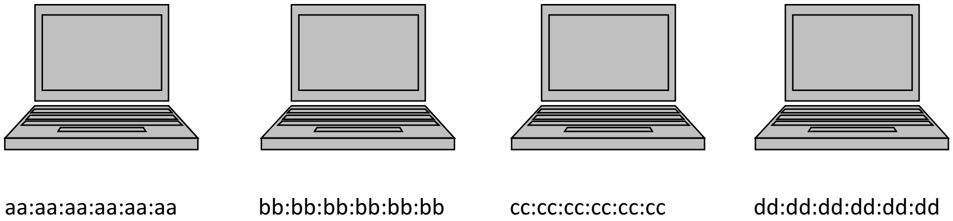


Figure 20: Multihop wireless network

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 Figure 20). However, both machines can hear machine bb:bb:bb:bb:bb:bb. To achieve this we will use the iptables utility. 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 `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 10.0.0._
   ```

   where `10.0.0._` 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?

# References

[1] IEEE 802.11e WG. Amendment : Medium Access Control (MAC) Quality of Service (QoS) Enhancements, January 2005.

[2] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). *Request for Comments-3626*, October 2003.

[3] D. B. Johnson, D. A. Maltz, and J. Broch. Ad Hoc Networking, Chapter 5: DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, 2001. Addison-Wesley.

[4] IEEE 802.11 WG part 11a. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, High-speed Physical Layer in the 5 GHz Band, 1999.

[5] IEEE 802.11 WG part 11b. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Higher Speed PHY Layer Extension in the 2.4 GHz Band, 1999.

[6] IEEE 802.11 WG part 11g. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Further Higher Speed Physical Layer Extension in the 2.4 GHz Band, 2003.

[7] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. *draft-ietf-manet-aodv-08.txt*, March 2001.

[8] Uppsala University and Ericsson AB. Aodv implementation aodv-uu v0.9 rfc 3561, 2007.

[9] IEEE 802.11 WG. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999.