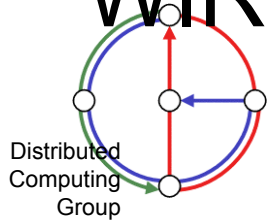


# Chapter 4

# WIRELESS LAN



Mobile Computing  
Summer 2002

## Overview



- Design goals
- Characteristics
- IEEE 802.11
  - Architecture
  - Protocol
  - PHY
  - MAC
  - Roaming
  - Security
  - a, b, d, etc.
- Short intermezzo on Cyclic Redundancy codes



## Design goals



- Global, seamless operation
- Low power consumption for battery use
- No special permissions or licenses required
- Robust transmission technology
- Simplified spontaneous cooperation at meetings
- Easy to use for everyone, simple management
- Interoperable with wired networks
- Security (no one should be able to read my data), privacy (no one should be able to collect user profiles), safety (low radiation)
- Transparency concerning applications and higher layer protocols, but also location awareness if necessary



## Characteristics



- + Very flexible (economical to scale)
- + Ad-hoc networks without planning possible
- + (Almost) no wiring difficulties (e.g. historic buildings, firewalls)
- + More robust against disasters or users pulling a plug
- Low bandwidth compared to wired networks (10 vs. 100[0] Mbit/s)
- Many proprietary solutions, especially for higher bit-rates, standards take their time
- Products have to follow many national restrictions if working wireless, it takes a long time to establish global solutions (IMT-2000)
- Security
- Economy

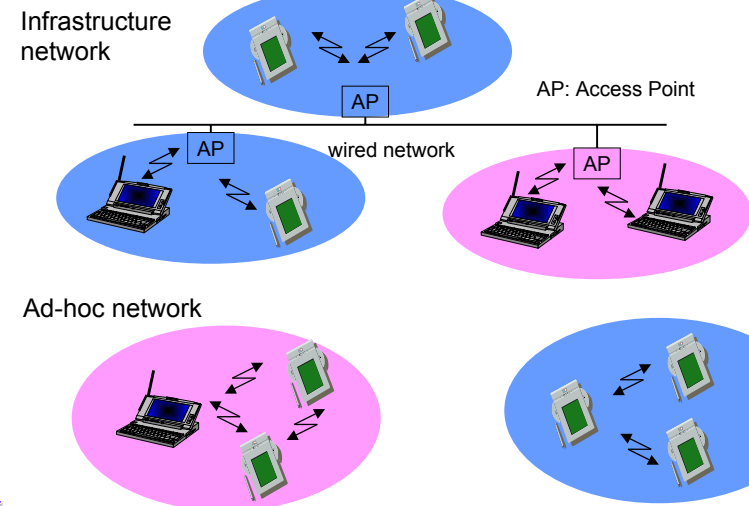


## Infrared vs. Radio transmission

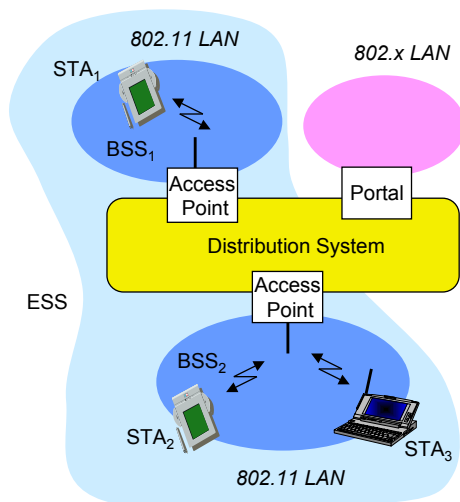
- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>• Infrared</li> <li>• uses IR diodes, diffuse light, multiple reflections (walls, furniture etc.)</li> <li>+ simple, cheap, available in many mobile devices</li> <li>+ no licenses needed</li> <li>+ simple shielding possible</li> <li>- interference by sunlight, heat sources etc.</li> <li>- many things shield or absorb IR light</li> <li>- low bandwidth</li> <li>• Example: IrDA (Infrared Data Association) interface available everywhere</li> </ul> | <ul style="list-style-type: none"> <li>• Radio</li> <li>• typically using the license free ISM band at 2.4 GHz</li> <li>+ experience from wireless WAN and mobile phones can be used</li> <li>+ coverage of larger areas possible (radio can penetrate walls, furniture etc.)</li> <li>- very limited license free frequency bands</li> <li>- shielding more difficult, interference with other electrical devices</li> <li>• Examples: HIPERLAN, Bluetooth</li> </ul> |
|--|--|



## Infrastructure vs. ad-hoc networks



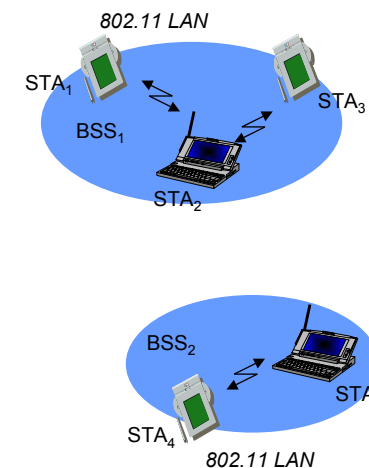
## 802.11 – Architecture of an infrastructure network



- Station (STA)
  - terminal with access mechanisms to the wireless medium and radio contact to the access point
- Basic Service Set (BSS)
  - group of stations using the same radio frequency
- Access Point
  - station integrated into the wireless LAN and the distribution system
- Portal
  - bridge to other (wired) networks
- Distribution System
  - interconnection network to form one logical network (EES: Extended Service Set) based on several BSS



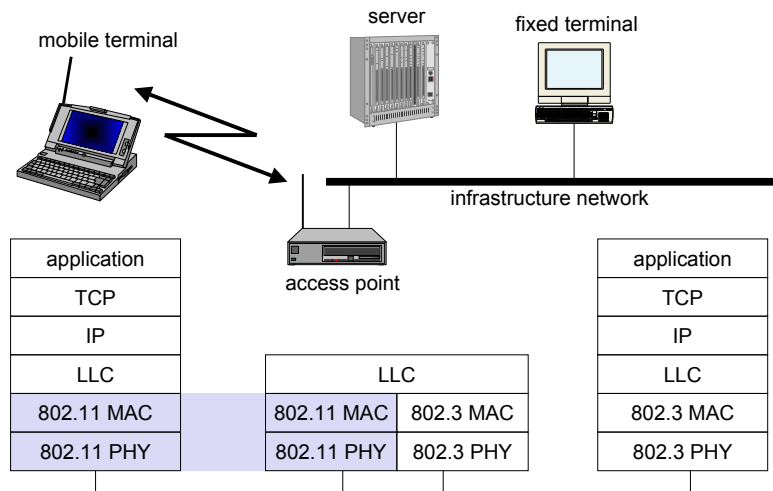
## 802.11 – Architecture of an ad-hoc network



- Direct communication within a limited range
  - Station (STA): terminal with access mechanisms to the wireless medium
  - Basic Service Set (BSS): group of stations using the same radio frequency
- You may use SDM or FDM to establish several BSS.

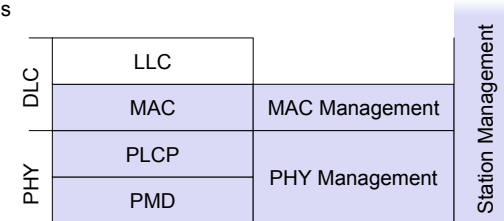


## 802.11 – Protocol architecture



## 802.11 – The lower layers in detail

- PMD (Physical Medium Dependent)
  - modulation, coding
- PLCP (Physical Layer Convergence Protocol)
  - clear channel assessment signal (carrier sense)
- PHY Management
  - channel selection, PHY-MIB
- Station Management
  - coordination of all management functions
- MAC
  - access mechanisms
  - fragmentation
  - encryption
- MAC Management
  - Synchronization
  - roaming
  - power management
  - MIB (management information base)



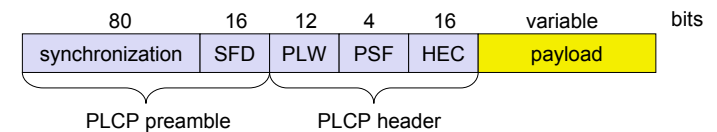
## 802.11 - Physical layer

- 3 versions: 2 radio (2.4 GHz), 1 IR:
- FHSS (Frequency Hopping Spread Spectrum)
  - spreading, despreading, signal strength, 1 Mbit/s
  - at least 2.5 frequency hops/s, two-level GFSK modulation
- DSSS (Direct Sequence Spread Spectrum)
  - DBPSK modulation for 1 Mbit/s (Differential Binary Phase Shift Keying), DQPSK for 2 Mbit/s (Differential Quadrature PSK)
  - preamble and header of a frame is always transmitted with 1 Mbit/s, rest of transmission 2 (or optionally 1) Mbit/s
  - chipping sequence: Barker code (+ - + + - + + - -)
  - max. radiated power 1 W (USA), 100 mW (EU), min. 1mW
- Infrared
  - 850-950 nm, diffuse light, 10 m range
  - carrier detection, energy detection, synchronization



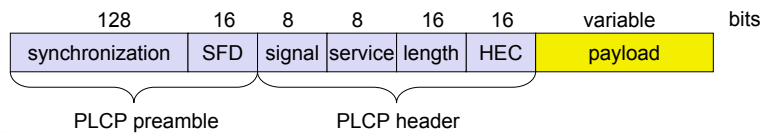
## FHSS PHY packet format

- Synchronization
  - synch with 010101... pattern
- SFD (Start Frame Delimiter)
  - 0000110010111101 start pattern
- PLW (PLCP\_PDU Length Word)
  - length of payload incl. 32 bit CRC of payload, PLW < 4096
- PSF (PLCP Signaling Field)
  - data rate of payload (1 or 2 Mbit/s)
- HEC (Header Error Check)
  - CRC with  $x^{16}+x^{12}+x^5+1$



## DSSS PHY packet format

- Synchronization
  - synch., gain setting, energy detection, frequency offset compensation
- SFD (Start Frame Delimiter)
  - 1111001110100000
- Signal
  - data rate of the payload (0x0A: 1 Mbit/s DBPSK; 0x14: 2 Mbit/s DQPSK)
- Service (future use, 00: 802.11 compliant)
- Length (length of the payload)
- HEC (Header Error Check)
  - protection of signal, service and length,  $x^{16}+x^{12}+x^5+1$



## Cyclic Redundancy Code (CRC): Ring

- Polynomes with binary coefficients  $b_k x^k + b_{k-1} x^{k-1} + \dots + b_0 x^0$
- Order of polynome: max  $i$  with  $b_i \neq 0$
- Binary coefficients  $b_i$  (0 or 1) form a field with operations “+” (XOR) and “.” (AND).
- The polynomes form a ring  $R$  with operations “+” and “.”:  $(R, +)$  is an abelian group,  $(R, \cdot)$  is an associative set, and the distributive law does hold, that is,  $a \cdot (b+c) = a \cdot b + a \cdot c$  respectively  $(b+c) \cdot a = b \cdot a + c \cdot a$  with  $a, b, c \in R$ .

• Example:

$$\begin{aligned} & (x^3+1) \cdot (x^4+x+1) && 1001 \cdot 10011 \\ & = x^3 \cdot (x^4+x+1) + 1 \cdot (x^4+x+1) && = 10011 \\ & = (x^7+x^4+x^3) + (x^4+x+1) && + 10011000 \\ & = x^7+x^3+x+1 && = 10001011 \end{aligned}$$



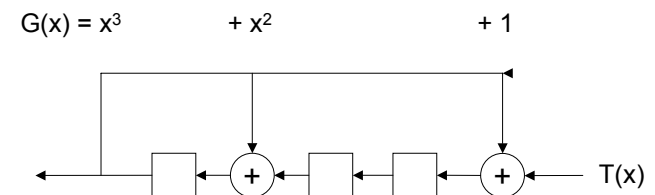
## Cyclic Redundancy Code (CRC): Division

- Generator polynome  $G(x) = x^{16}+x^{12}+x^5+1$
  - Let the whole header be polynome  $T(x)$  (order  $< 48$ )
  - Idea: fill HEC (CRC) field such that  $T(x) \bmod G(x) = 0$ .
  - How to divide with polynomes? Example with  $G(x) = x^2+1 (=101)$   
 $11101100 / 101 = 110110$ , Remainder 10
- $$\begin{array}{r} 100 \\ 011 \\ 111 \\ 100 \\ 010 \end{array}$$
- Idea: Fill CRC with remainder when dividing  $T(x)$  with  $HEC=00\dots0$  by  $G(x)$ . Then calculating and testing CRC is the same operation.



## Cyclic Redundancy Code (CRC): Division in Hardware

- Use cyclic shift register  $r$  registers, where  $r$  is the order of  $G(x)$
- Example



Finally the remainder of the division is in the registers



## Cyclic Redundancy Code (CRC): How to chose G(x)?

- Generator polynome  $G(x) = x^{16} + x^{12} + x^5 + 1$
- Why does  $G(x)$  have this complicated form?
- Let  $E(x)$  be the transmission errors, that is  $T(x) = M(x) + E(x)$
- $T(x) \bmod G(x) = (M(x) + E(x)) \bmod G(x)$   
 $= M(x) \bmod G(x) + E(x) \bmod G(x)$
- Since  $M(x) \bmod G(x) = 0$  we can detect all transmission errors as long as  $E(x)$  is not divisible by  $G(x)$  without remainder
- One can show that  $G(x)$  of order  $r$  can detect
  - all single bit errors as long as  $G(x)$  has 2 or more coefficients
  - all bursty errors (burst of length  $k$  is  $k$ -bit long  $1xxx1$  string) with  $k \leq r$  (note: needs  $G(x)$  to include the term 1)
  - Any error with probability  $2^{-r}$



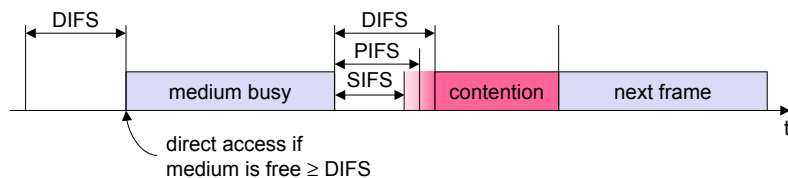
## MAC layer: DFWMAC

- Traffic services
  - Asynchronous Data Service (mandatory)
    - exchange of data packets based on “best-effort”
    - support of broadcast and multicast
  - Time-Bounded Service (optional)
    - implemented using PCF (Point Coordination Function)
- Access methods
  - DFWMAC-DCF CSMA/CA (mandatory)
    - collision avoidance via binary exponential back-off mechanism
    - minimum distance between consecutive packets
    - ACK packet for acknowledgements (not used for broadcasts)
  - DFWMAC-DCF w/ RTS/CTS (optional)
    - avoids hidden terminal problem
  - DFWMAC-PCF (optional)
    - access point polls terminals according to a list

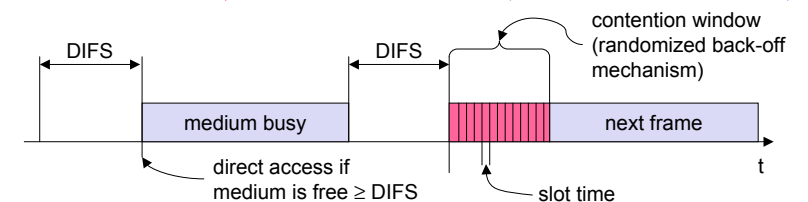


## MAC layer

- defined through different inter frame spaces
- no guaranteed, hard priorities
- SIFS (Short Inter Frame Spacing)
  - highest priority, for ACK, CTS, polling response
- PIFS (PCF IFS)
  - medium priority, for time-bounded service using PCF
- DIFS (DCF, Distributed Coordination Function IFS)
  - lowest priority, for asynchronous data service



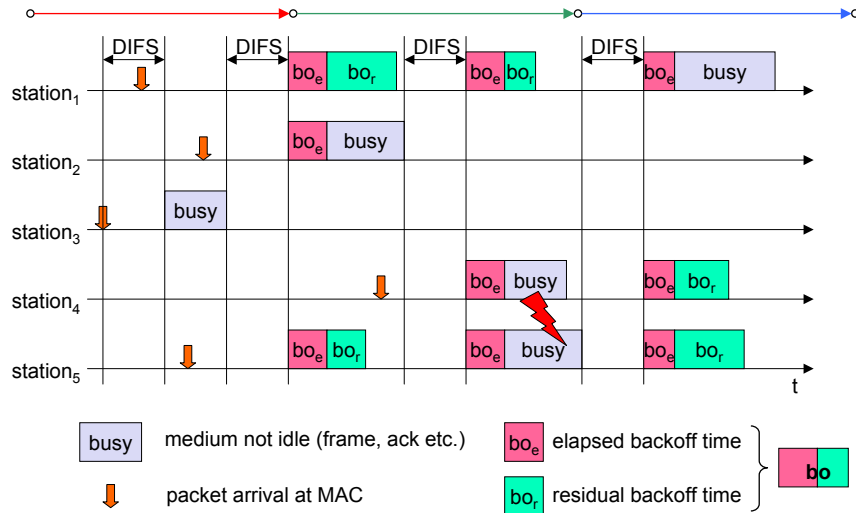
## CSMA/CA



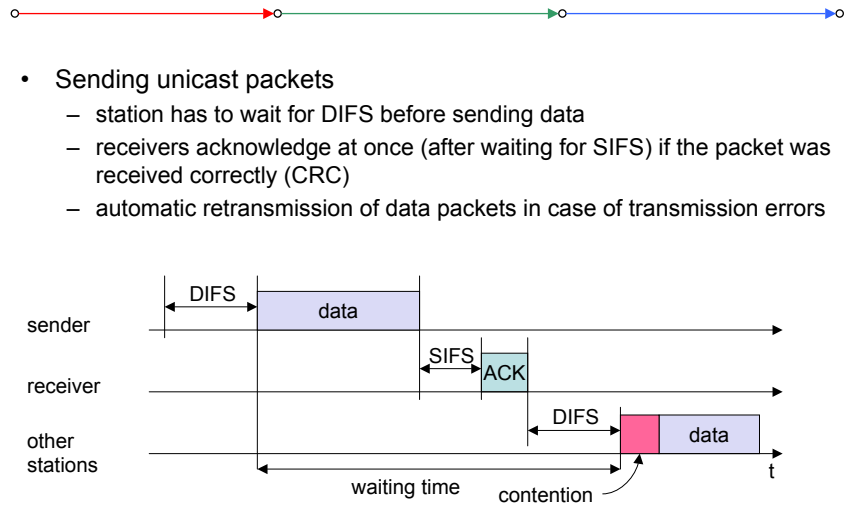
- station ready to send starts sensing the medium (Carrier Sense based on CCA, Clear Channel Assessment)
- if the medium is free for the duration of an Inter-Frame Space (IFS), the station can start sending (IFS depends on service type)
- if the medium is busy, the station has to wait for a free IFS, then the station must additionally wait a random back-off time (collision avoidance, multiple of slot-time)
- if another station occupies the medium during the back-off time of the station, the back-off timer stops (fairness)



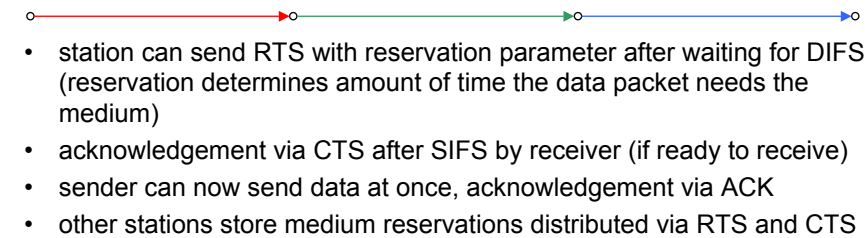
## Competing stations - simple example



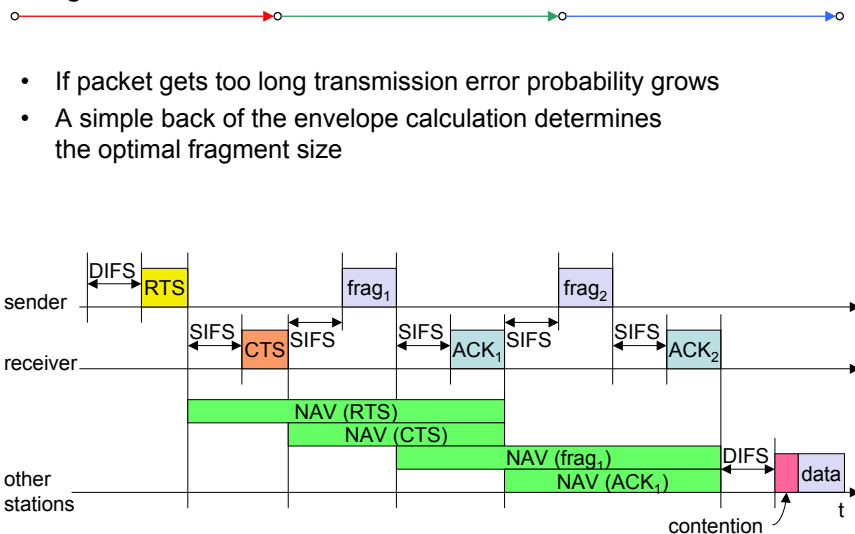
## CSMA/CA 2



## DFWMAC



## Fragmentation



## Fragmentation: What fragment size is optimal?

- Total data size:  $D$  bits
- Overhead per packet (header):  $h$  bits
- Overhead between two packets (acknowledgement):  $a$  “bits”
- We want  $f$  fragments, then each fragment has  $k = D/f + h$  data + header bits
- Channel has bit error probability  $q = 1-p$
- Probability to transmit a packet of  $k$  bits correctly:  $P := p^k$
- Expected number of transmissions until packet is success:  $1/P$
- Expected total cost for all  $D$  bits:  $f \cdot (k/P + a)$
- Goal: Find a  $k > h$  that minimizes the expected cost



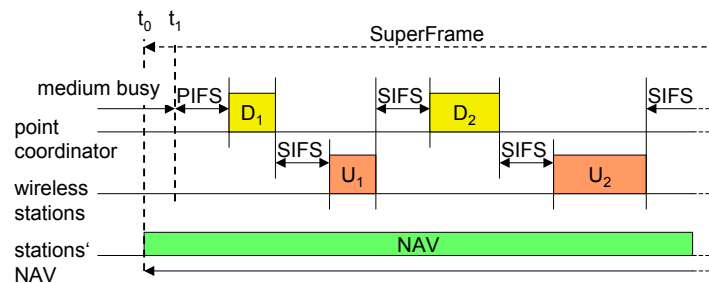
## Fragmentation: What fragment size is optimal?

- For the sake of a simplified analysis we assume  $a = O(h)$
- If we further assume that a header can be transmitted with constant probability  $c$ , that is,  $p^h = c$ .
- We choose  $k = 2h$ ; Then clearly  $D = f \cdot h$ , and therefore expected cost
 
$$f \cdot \left( \frac{k}{P} + a \right) = \frac{D}{h} \left( \frac{2h}{p^{2h}} + O(h) \right) = O\left( \frac{D}{p^{h^2}} \right) = O\left( \frac{D}{c^2} \right) = O(D).$$
- If already a header cannot be transmitted with high enough probability, then you might keep the message very small, for example  $k = h + 1/q$

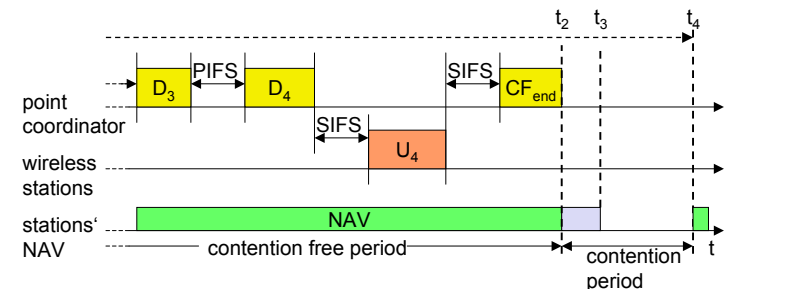


## DFWMAC-PCF

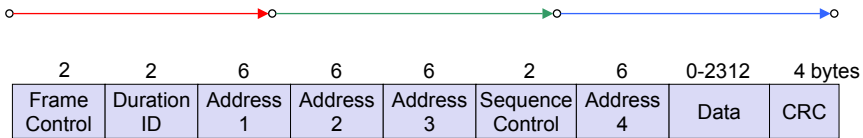
- An access point can poll stations



## DFWMAC-PCF 2



## Frame format



version, type, fragmentation, security, two DS-bits, ...

- Type
  - control frame, management frame, data frame
- Sequence control
  - important against duplicated frames due to lost ACKs
- Addresses
  - receiver, transmitter (physical), BSS identifier, sender (logical)
- Miscellaneous
  - sending time, checksum, frame control, data



## MAC address format



scenario	to DS	from DS	address 1	address 2	address 3	address 4
ad-hoc network	0	0	DA	SA	BSSID	-
infrastructure network, from AP	0	1	DA	BSSID	SA	-
infrastructure network, to AP	1	0	BSSID	SA	DA	-
infrastructure network, within DS	1	1	RA	TA	DA	SA

DS: Distribution System  
 AP: Access Point  
 DA: Destination Address  
 SA: Source Address  
 BSSID: Basic Service Set Identifier  
 RA: Receiver Address  
 TA: Transmitter Address



## MAC management



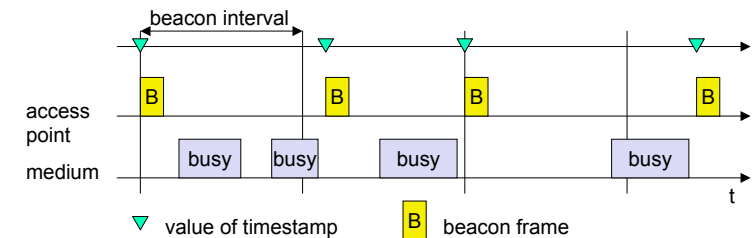
- Synchronization
  - try to find a LAN, try to stay within a LAN
  - timer etc.
- Power management
  - sleep-mode without missing a message
  - periodic sleep, frame buffering, traffic measurements
- Association/Reassociation
  - integration into a LAN
  - roaming, i.e. change networks by changing access points
  - scanning, i.e. active search for a network
- MIB - Management Information Base
  - managing, read, write



## Synchronization



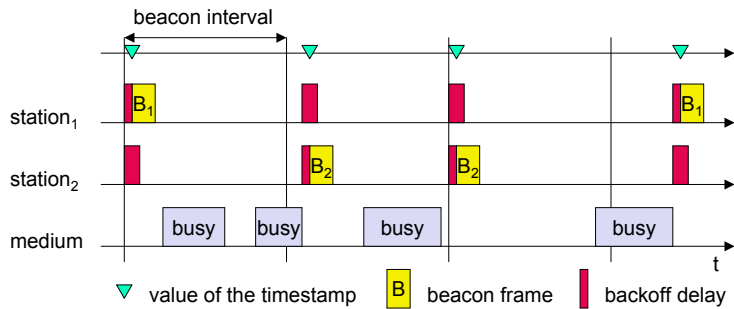
- In an infrastructure network, the access point can send a beacon





## Synchronization

- In an ad-hoc network, the beacon has to be sent by any station

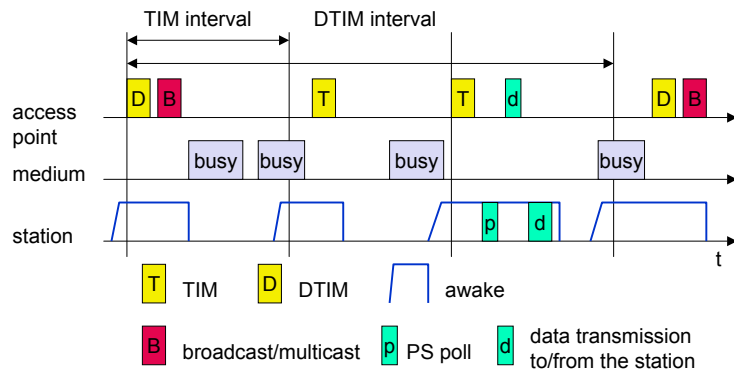


## Power management

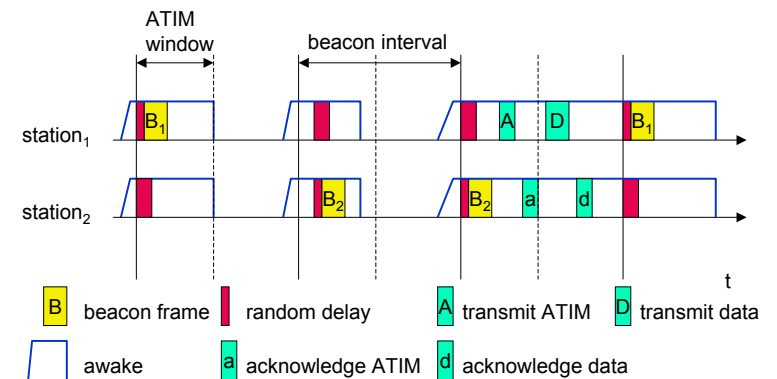
- Idea: if not needed turn off the transceiver
- States of a station: sleep and awake
- Timing Synchronization Function (TSF)
  - stations wake up at the same time
- Infrastructure
  - Traffic Indication Map (TIM)
    - list of unicast receivers transmitted by AP
  - Delivery Traffic Indication Map (DTIM)
    - list of broadcast/multicast receivers transmitted by AP
- Ad-hoc
  - Ad-hoc Traffic Indication Map (ATIM)
    - announcement of receivers by stations buffering frames
    - more complicated - no central AP
    - collision of ATIMs possible (scalability?)



## Power saving with wake-up patterns (infrastructure)



## Power saving with wake-up patterns (ad-hoc)



## Roaming

- No or bad connection? Then perform:
- Scanning
  - scan the environment, i.e., listen into the medium for beacon signals or send probes into the medium and wait for an answer
- Association Request
  - station sends a request to one or several AP(s)
- Association Response
  - success: AP has answered, station can now participate
  - failure: continue scanning
- AP accepts association request
  - signal the new station to the distribution system
  - the distribution system updates its data base (i.e., location information)
  - typically, the distribution system now informs the old AP so it can release resources

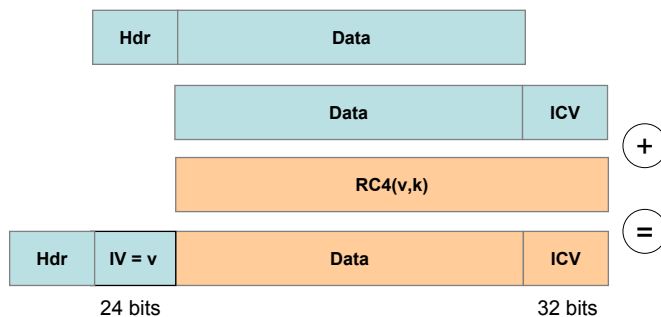


## 802.11 Security Today

- Existing security consists of two subsystems:
  - Wired Equivalent Privacy (WEP): A data encapsulation technique.
  - Shared Key Authentication: An authentication algorithm
- Goals:
  - Create the privacy achieved by a wired network
  - Simulate physical access control by denying access to unauthenticated stations



## WEP Encapsulation



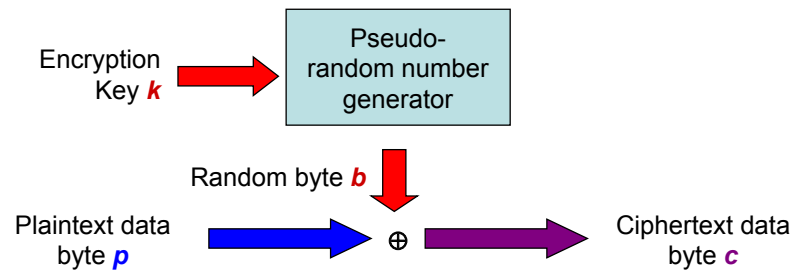
## WEP protocol

- The sender and receiver share a secret key  $k$
- Sender, in order to transmit a message:
  - Compute a CRC-32 checksum ICV, and attach it to the message
  - Pick a per-packet key IV  $v$ , and generate a keystream  $RC4(v,k)$
  - Attention: WEP Allows  $v$  to be re-used with any packet
  - Encrypt data and attached ICV by XORing it with  $RC4(v,k)$
  - Transmit header, IV  $v$ , and encrypted data/ICV
- Receiver:
  - Use received IV  $v$  and shared  $k$  to calculate keystream  $RC4(v,k)$
  - Decrypt data and ICV by XORing it with  $RC4(v,k)$
  - Check whether ICV is a valid CRC-32 checksum



## Vernam Ciphers

The WEP encryption algorithm RC4 is a Vernam Cipher:



Decryption works the same way:  $p = c \oplus b$



## Properties of Vernam Ciphers

Thought experiment: what happens when  $p_1$  and  $p_2$  are encrypted under the same "random" byte  $b$ ?

$$c_1 = p_1 \oplus b$$

$$c_2 = p_2 \oplus b$$

Then: 
$$c_1 \oplus c_2 = (p_1 \oplus b) \oplus (p_2 \oplus b) = p_1 \oplus p_2$$

Conclusion: it is a bad idea to encrypt any two bytes of data using the same byte output by a Vernam Cipher PRNG.



## How to read WEP encrypted traffic



- By the Birthday Paradox, probability  $P_n$  two packets will share same IV after  $n$  packets is  $P_2 = 1/2^{24}$  after two frames and  $P_n = P_{n-1} + (n-1)(1-P_{n-1})/2^{24}$  for  $n > 2$ .
- 50% chance of a collision exists already after 4823 packets.
- Pattern recognition can disentangle the XOR'd recovered plaintext.
- Recovered ICV can tell you when you've disentangled plaintext correctly (or help to recover the plaintext in the first place).
- Once you know a single RC4, you can inject your own packets



## How to read WEP encrypted traffic

- Ways to accelerate the process:
  - Send spam into the network, then you already know the plaintext.
  - Get the victim to send e-mail to you, the AP creates the plaintext, just for you.
  - For a given AP, everybody uses the same secret key  $k$
- Very bad: Many 802.11 cards reset their IV (=v) counter to 0 every time they are activated, and simply increment it for each packet they transmit. In this case a spy knows the RC(v,k) for low v values in short time.
- Naturally a spy would use a decryption dictionary to store the already found RC4(v,k)... needs at most  $2^{24} \cdot 1500$  bytes = 24GBytes



## Traffic Modification

Thought experiment: how hard is it to change a genuine packet's data, so ICV won't detect the change?

Represent an  $n$ -bit plaintext as an  $n$ -th degree binomial polynomial:

$$p = b_n x^n + b_{n-1} x^{n-1} + \dots + b_0 x^0$$

Then the plaintext with ICV can be represented as :

$$p x^{32} + \text{ICV}(p) = b_n x^{n+32} + b_{n-1} x^{n+31} + \dots + b_0 x^{32} + \text{ICV}(p)$$

If the  $n+32$  bit RC4 key stream used to encrypt the body is represented by the  $n+32^{\text{nd}}$  degree polynomial  $r$ , then the encrypted message body is

$$p x^{32} + \text{ICV}(p) + r$$



## Traffic Modification 2

But the ICV is linear, meaning for any polynomials  $p$  and  $q$

$$\text{ICV}(p+q) = \text{ICV}(p) + \text{ICV}(q)$$

This means that if  $q$  is an arbitrary  $n$ th degree polynomial, i.e., an arbitrary change in the underlying message data:

$$\begin{aligned} (p+q)x^{32} + \text{ICV}(p+q) + r &= p x^{32} + q x^{32} + \text{ICV}(p) + \text{ICV}(q) + r \\ &= ((p x^{32} + \text{ICV}(p)) + r) + (q x^{32} + \text{ICV}(q)) \end{aligned}$$

Conclusion: Anyone can alter an WEP encapsulated packet in arbitrary ways without detection, and without knowing RC4( $v,k$ )



## WEP Authentication

- Goal is that client joining the network really knows the shared key  $k$
- Protocol:
  - Access point sends a challenge string to client
  - Client WEP-encrypts challenge, and sends result back to AP
  - If the challenge is encrypted correctly, AP accepts the client
- Client can spoof protocol the same way as injecting a message.
- All a client needs is a valid RC4( $v,k$ ), for some  $v$ .



## WEB Message decryption revisited

- How can a client decrypt a specific packet with IV  $v$  for which the client does not have the RC4( $v,k$ ). (The first packet that uses  $v$ .)
- Idea: Use the access point (who knows  $k$ )
- Spoofing protocol (one of many possibilities):
  - Join the network (authentication spoofing)
  - Send a handcrafted message “encrypted” with key  $v$  to a destination you control, for example a node outside the wireless LAN.
  - The AP will “decrypt” the message for you, and forward it to your destination. When you XOR the “encrypted” with the “decrypted” message, you get the RC( $v,k$ ) for the  $v$  you wanted.
- There are some tedious details – but there are also other protocols



## WEP lessons

- What could one do to improve WEP:
  - Use long IV's that are used only once in the lifetime of a shared key  $k$
  - Use a strong message authentication code (instead of a CRC code), that does depend on the key and the IV.
- What you should do:
- Don't trust WEP. Don't trust it more than sending plain messages over an Ethernet. However, WEP is usually seen as a good first deterrent against so-called "war drivers."
- Put the wireless network outside your firewall
- There are new proprietary security solutions such as LEAP.
- Use other security mechanisms such as VPN, IPSec, ssh



## Future developments

- IEEE 802.11a
  - compatible MAC, but now 5 GHz band
  - transmission rates up to 20 Mbit/s
  - close cooperation with BRAN (ETSI Broadband Radio Access Network)
- IEEE 802.11b
  - higher data rates at 2.4 GHz
  - proprietary solutions offer 11 Mbit/s
- IEEE WPAN (Wireless Personal Area Networks)
  - market potential
  - compatibility
  - low cost/power, small form factor
  - technical/economic feasibility
  - Example: Bluetooth

