A bit more about the technologies involved...

Jean Tourrilhes
Hewlett Packard Laboratories, Palo Alto

3 August 00

La culture c'est comme la confiture, moins on en a, plus on l'étale...

1 Introduction

I'm not pretenting to teach a course on **Wireless LAN**. I guess that many books explain the subject in more details and accuracy than me (anyway, I hope). I just feel that many users of Wireless LANs don't really know what is inside their magic piece of kit and are curious about it. I hope that this document will help you to understand a bit more of the different technological aspects and compare the different Wireless LANs functionalities.

While working on the Wavelan driver and the Wireless Extensions, I've gathered much information trying to understand how it works. The vendors documentation and web sites have been also very helpful, many of them really try to explain the technologies behind their products and provide *white papers*. The Net contains also a lot of papers and reports on the subject of wireless LANs and radio communications.

I have still a limited knowledge and understanding of the wide number of technologies used by Wireless LANs, so I hope that it is mostly accurate, complete and that it will help you. If some knowledgeable person could help me to improve this document, or if anybody could give me some suggestions or corrections, I would be glad...

This document is the third part of the **Linux Wireless LAN Howto**, located at <http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/>, and available in HTML, PostScritp or PDF form. Please refer to its first part for details (copyright, disclaimers ...) and a list of some other web pages on the subject.

2 MAC, LAN, Layer and other strange words...

- **Bandwidth** : commonly it is the size of the channel used by the radio (the amount of frequency available to the system). By extension, it can also sometimes refer to the speed of the system (the bit rate).
- **Bit-rate** : speed at which bits are transmitted over the physical layer, also called signalling rate. Quite different from throughput (see *chapter 5.4.1*).
- **Carrier** : the base frequency used by the system. The modulation process will generate a signal centered on the carrier, of width equal to the bandwidth.
- **Carrier Sense** : checking the transmission medium to assess if it is free or if there is a transmission going on. Usually measure of the received power. See CSMA.
- **CDMA (Code Division Multiple Access)** : technique used to share the same bandwidth between different channels using codes. The code is a signature multiplexed with the signal and used to recover it. See *chapter 4.3.1*.
- **CSMA (Carrier Sense Multiple Access)** : using carrier sense to access the medium. One of the main MAC methods, see a verbose explanation in *chapter 5.1.2*.

- **Cell** : radio neighbourhood, area where all nodes can communicate with each other. As the range over radio is limited, the network is split into independent cells and a cell to cell communication is provided (via access point or internal routing).
- **Channel** : On the radio, this is usually synonym of a specific frequency, and by extension the communication medium. It can also mean a stream of data between two nodes (a point to point link in connection oriented systems).
- **dB (decibel)** : logarithmic way to express a value. Usually the signal strength (transmitted and received power) is expressed in dBm (the reference is 1 mW - 0 dBm). A difference between two values in dBm is without unit, in dB (in fact, this is a factor between the two values). See *chapter 4.6* .
- **Ethernet** : standard wired LAN protocol. Includes physical and link layers.
- **Fading** : variation in channel performance due to the dynamicity of the environment, make the receive signal strength change. See *chapter 4.8.1* .
- **FEC (Forward Error Correction)** : technique used to overcome some type of errors created by transmission on noisy channels, by adding redundancy bits to the main data transmission. See *chapter 4.8.3* .
- **Frequency band** : portion of the radio spectrum delimited for a particular use. For example, most wireless LANs use the 2.4 to 2.48 GHz band. A frequency band is usually divided in channels.
- **Header** : informations added by the protocol in front of the payload in the packet for its own use (addresses, packet type, sequence number, CRC...). Each protocol adds a different header, so in a typical TCP/IP packet as transmitted, we have a MAC header, an IP header and a TCP header, followed by the payload.
- **IP** : see TCP/IP.
- **IPX** : network protocol used in Netware, usually with SPX.
- **LAN (Local Area Network)** : network on a short distance, as opposed to WAN (typically inside a building).
- **Latency** : measure of the performance of a network for short requests and multimedia traffic. There is no real standard measurement, it might be the time to send and transmit a packet, or the time spent in the transmit queue, or the time for an answer to come back, or a number of requests per second...
- **Layer** : this terminology comes from the OSI specification. It divides any communicating system into 7 layers, each having a different functionality. Layer 1 is the physical layer, and layer 2 is the link layer. IP could be assimilated as layer 3 (network layer), and TCP as layer 4 (transport layer).
- **Link layer** : This is the part of the protocol managing the direct delivery between two devices on a specific physical layer (coaxial bus, point to point link, radio...). This includes packetisation and addressing. Most of this is implemented in the MAC.
- **MAC (Medium Access Control)** : this is the part of the radio device managing the protocol and the usage of the link. The MAC decides when to transmit and when to receive, creates the packets headers and filters the received packets. See *chapter 5.1* for the main examples of MAC protocols.

- **Medium** : name to describe the mean used to transfer information. This could be a wire (twisted pairs, coax...), an optic fibber, the radio waves (the air), infrared light...
- **Modem (modulator/demodulator)** : in a radio device, this is the part converting the bits to transmit into a modulation of the radio waves and the reverse at the reception. It does the analog to digital conversion, the generation of the frequency, the modulation and the amplification.
- **Modulation** : specific way of coding information on a radio frequency. Basically, there is amplitude modulation (AM - change waveform strength) and frequency/ phase modulation (FM - change waveform timing), but there exist many variations and combinations each designated by a specific acronym. See *chapter 4.7* .
- **NetBeui** : network protocol used in Lan Manager.
- **Node** : a device part of the network, source or destination of the data. For us, a computer with a radio card in it.
- **Noise** : any unwanted signal. Background noise, interferences, transmissions from nodes not belonging to the network... See *chapter 4.8* .
- **Packet** : Unit of transmission over the network. The data to be transmitted is split into packets, which are sent individually over the network.
- **Protocol** : specification of the interactions between systems and the data manipulated. This describes what to do and when (the rules), and the format of the data exchanged on the lower communication layer.
- **Physical layer** : this is the part of the device interacting with the medium. For a radio LAN, the physical layer is also called modem.
- **Radio** : electromagnetic waves. By extension, a device transmitting or receiving radio waves.
- **Roaming** : ability to move between cells of the same network. See *chapter 5.3.2* .
- **SNR (Signal to Noise Ratio)** : difference in strength between the signal we want to receive and the background noise (or any unwanted signal). See *chapter 4.6.4* .
- **TCP/IP** : network protocol used by Unix and Internet. Better in some respects than NetBeui and IPX (allows routing, for example).
- **TDMA (Time Division Multiple Access)** : technique used to share the same bandwidth between different channels using periodic time slots. See *chapter 5.1.1* .
- **Throughput** : measure of the performance of a network for large data transfer (such as FTP, NFS, HTTP 1.1). This speed is expressed in bits per seconds or a multiple.
- **WAN (Wide Area Network)** : network on a large scale : a town, a country or the world. Definitely not a LAN.
- **Wired** : using a wire.
- **Wireless** : not using a wire. For networks, it might be radio or infrared.

3 Anatomy of a radio LAN

A **radio network** is a collection of nodes communicating together through radio devices , using radio waves to carry the information exchanged (obvious , isn't it ?). It is sometime called a *radio Ethernet* , by analogy of the wired technology . Most **radio**

devices are a card (ISA, Pcmcia) to plug in a PC (or workstation), and interact directly with the standard networking stack on it (no need of PPP or any specific protocol stack).

3.1 The radio modem

A radio device is composed of two main parts. The first is the **radio modem**. This is the part transmitting (modulating) the data onto the frequency and receiving other transmissions. It is composed of *antenna(s)*, *amplificators*, *frequency synthesisers*, *filters* and other bits of magic. These are mainly analog parts, and a bit of digital (in an ASIC, the *Baseband*).

Usually, you can't see all those analog bits (and the cleverness of the board layout) because all the modem is encapsulated in a metal shield to protect your PC from those high frequency radiations.

The modem main characteristics are the *frequency band*, the *signalling rate*, the *modulation* and the *transmitted power*. People building modems are also talking a lot of SNR and dB ...

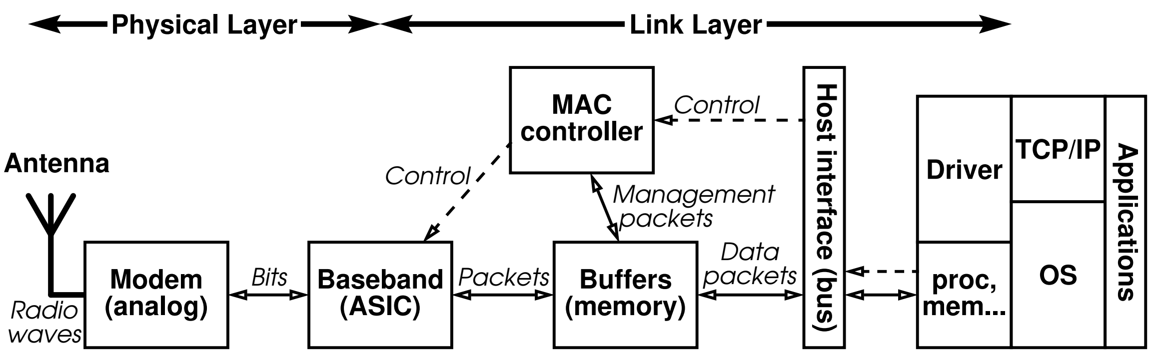
3.2 The MAC controller

The second part of the radio device is the **MAC controller**, responsible to run the MAC protocol. This is implemented mainly in an ASIC and/or a microcontroler on the card, but some functionalities of the MAC may be as well in the driver on the PC. The card also includes some memory for the MAC controller to store incoming and outgoing packets (buffers) and other data (configuration, statistics).

Most of the time the few most time critical parts are handled in the radio modem ASIC (the baseband), the bulk of the MAC in a microcontroller and only some management functionality in the driver. But, the different manufacturers place the boundary between the different functionalities differently (cost/performance tradeoff), and some have implemented driver only MACs for lower cost.

The main characteristics of the MAC are the *packet format* (size, headers), the *channel access mechanisms* and the *network management* features. The amount of on-board memory is also important, because the MAC may need a significant number of buffers to compensate the PC and interface latencies.

Functional diagram of a Wireless device:



3.3 The host interface

The card **interface** to the PC through one of its buses (*ISA, PCI, Pcmcia ...*) or communication ports (*serial, parallel, USB* or *Ethernet*). This interface allows the software (mostly the driver) to communicate with the MAC controller and most of the time directly to the on board memory (the software writes packets to a specific location of it, then the controller reads them and sends them).

The main characteristic of the interface is mainly the speed (i/o, shared memory or DMA) and the ability to process requests in parallel. The flexibility and functionality of it are usually more a concern for the person writing the driver :-)

3.4 The driver

With all modern operating systems, the end application doesn't access directly the hardware but use a standard API. The operating system needs a **driver** to interface the hardware to the network stack (*TCP/IP*, *NetBeui*, *IPX*...). The main function of the driver is to manage the hardware and to answer its request (to service interrupts). In most of the Wireless LANs, the driver also implements some parts of the MAC protocol.

The main characteristic of the driver is the bugs :-(

3.5 Wireless LAN or not

Wireless LANs are not the only devices to make use of wireless technology, and it's easy to get confused between the different products (especially that sometimes they call themselves incorrectly wireless networks). Some example are *wireless bridges*, *wireless distribution systems* and *cable replacement*, and they are quite different from local area networking. There is also *wide area wireless* network products, which are again quite different from LANs.

Wireless Bridges are used to connect two different LAN segments via radio, for example between two buildings across the street. **Wireless distribution systems** is what are used by ISP to connect multiple independant customers to a base station, like houses in a neighbourhood. **Cable replacement** is mostly like IrDA (Infrared data link) to transfer data between two computers without a serial or parallel cable.

Sometimes those products use standard Wireless LAN modules, and most of the time they are based on the same technologies as Wireless LANs but with restricted functionality (like no broadcasting) and only allow a set of point to point links (so, no native TCP/IP topology). They interface to the serial port (cable replacement) or ethernet port (wireless bridges, wireless distribution system).

In this document we mostly restrict ourselves to true wireless LANs, because what doesn't run natively TCP/IP is not "fun" :-)

3.6 Professional and Home Wireless LANs

Now that Wireless LANs are getting towards lower price, Wireless LAN manufacturers are no longer targeting mobile commercial users only but also the home market. Some vendors, such as Proxim, offer two distinct line of product based on the same technology (and same protocol), the RangeLan2 for professionals and Symphony for home users.

As the business version of those Wireless LANs are more expensive than the home products, one might wonder what justify the price difference apart from the packaging, the marketing and software bundle.

The radio modems may present **different performances**. The modem is usually the most expensive part of the device, and replacing analog parts by less performant ones may reduce the price. The result may be a lower sensitivity, or less filtering of the adjacent bands or channels, which may reduce range and performance, especially for high number of nodes or collocated networks (which matter most for business environment).

The host interface may be different. The business line may offer more options, such as Ethernet, Serial and PCI, whereas home version may offer USB. The home line may also lack security (through encryption) or power management.

But in most cases, the hardware between the two lines is exactly the same. In fact, most of the differences usually reside in the **Access Points**. This is why Lucent offer 4 different Access Points depending on usage and targeted at different kind of users, but only one type of card for all types of users.

Access Points for home users are mostly designed to interface with a phone line (or ISDN, DSL or cable modem) and provide a proxy or masquerading feature, allowing the user to share its ISP access between the nodes of the network.

On the other hand, Access Points for businesses connect directly to the LAN via Ethernet or act as wireless repeaters, with optimised bridge functionality, higher performance, offer a wide range of management features (diagnostic, statistics, access control...) roaming and out of range forwarding (see *chapter 5.3.2*).

So, before investing your money, you have to ask yourself what network configuration you are really after and which features you really do need...

3.7 Digital radios and changing the protocol

One question popping up in my mailbox is the ability of doing protocol ‘X’(TDMA, Wireless ATM) with device ‘Y’(a well known Wireless LAN). A variant of this question is people trying to implement a specific scheme or optimisation in the 802.11 protocol.

This is usually not possible. As we have seen above, most of the MAC protocol is actually embedded in the device and only a few non performance critical functions are handled by the driver on the host. Usually, manufacturers don’t tell you how to reprogram the firmware of their devices, but even if it was possible, it would not be enough.

The very low part of the MAC protocol, which is time critical, is implemented in the baseband ASIC, so quite a challenge to change. For example the carrier sense and MAC acknowledgement need reaction in the order of a dozen microseconds, so are prime candidate for the ASIC. Unfortunately, these are precisely the functions that those people want to change.

In fact, many people have been thinking of universal radios, which can be simply reprogrammed to receive (and transmit) any radio standard. The main idea is having a big block of reprogrammable logic on the card and to download a new configuration for each protocol that the system wants to use, making it a fully **digital radio**.

To achieve that goal, we need to go one step further down, and be able to adapt to any modulation and bit rate. Most implementations of common Wireless LANs use fixed analog components in the modem, so are not suitable. So, a digital radio needs to digitise (with a fast AtoD) the whole bandwidth and to feed that the a fast super DSP or EPLD (Electric Programable Logic Device, like a Xilinx or Altera) and to work entirely in the digital domain to demodulate (and modulate) the signal. Unfortunately this is not really cost effective and doesn’t work that well at the frequency we are talking about (GHz).

4 The radio modem (physical layer)

This section of the document deals with all the issues related to the physical layer (bottom of the pile, OSI wise :-), or in our case the radio modem.

4.1 ISM frequency bands (900 MHz & 2.4 GHz)

In every country, the use of the radio spectrum is **regulated** by some organisations. This is the *FCC* for North America and the *ETSI* for Europe. These regulators define the allocation of each radio frequency bandwidth: for TV and radio broadcasting, for the telecommunication operators, for the army... Usually, to use a frequency band, you must negotiate with these bodies, register your architecture and buy the right to use the frequency.

These organisations, aware of the prospects of local radio communications for individual users, have allocated some specific frequency bands to be used in a more flexible way. The oldest and most commonly used ones are located at 900 MHz and 2.4 GHz and called the **ISM bands** (*Industrial, Scientific* and *Medical*). The main characteristic of these bands is that they are **unlicensed**, this means that the user is free to use them without having to register or to pay anything (apart from the radio hardware).

Of course, to avoid abuses, these organisations have imposed a set of rules for these frequency bands and only the products certified to conform to those rules are allowed to emit in the bands. These rules specify at least the maximum power transmitted in the band and the out of band emissions (to not pollute adjacent bands). The ISM bands rules specify as well that **Spread Spectrum** has to be used (either *Direct Sequence* or *Frequency Hopping*, see *chapter 4.3*), and how the channels are defined, to allow the peaceful cohabitation of different systems (that's the theory).

The Spread Spectrum rules mandate *Direct Sequence* systems must spread their signal at least 11 times, and that *Frequency Hopping* systems stay on a channel a maximum of 0.4 s and use 75 channels at minimum in each 30 s period. But, don't trust me, check the exact wording of the rules...

These rules may vary depending on the country: the FCC allocates both the 900 MHz and 2.4 GHz band with 1 W maximum power, whereas the ETSI allocates only the 2.4 GHz band with 100 mW maximum power (900 MHz is used for GSM cell phones in Europe). The 2.4 GHz band is available worldwide and the regulations are mostly compatible between the different authorities (usually 80 MHz of bandwidth between 2.4 GHz and 2.48 GHz). The main exception is Japan which has some additional constraints.

The Spread Spectrum rules originally allowed around 2 Mb/s maximum bit rate (both FH and DS), but the Direct Sequence people managed to find a loophole and now offer 11 Mb/s systems (see *chapter 4.7.3*).

Because these bands are "free", they may be heavily **polluted** by other unlicensed systems. The 2.4 GHz band also suffers from the microwave oven radiations (this explains why it was given for free).

Please note that the regulation for unlicensed bands is quite different from the bands reserved for radio amateurs (HAM). HAM people are not happy because their regulations are much more strict (they have to pass an examination including morse code and follow stricter etiquette) and the bandwidth available to them much more scarce.

4.2 5 GHz frequency bands (HiperLan and UNII band)

The 5 GHz unlicensed bands are another very complicated story.

ETSI was the first to open the 5 GHz band, and so far, the 5.2 GHz band is dedicated to **HiperLan** (see *chapter 6.3*), and the 5.4 GHz band reserved for **HiperLan II** (alias BRAN, see *chapter 6.4*). As they have done for *GSM* and *DECT*, only systems that fully conform to those standards (Phy and MAC) may operate in the band.

In the States, the FCC has allocated the band between 5.2 and 5.8 GHz (**UNII band**) with some very liberal rules (no spread Spectrum mandated, no channels allocated). To limit systems, they have introduced complicated power rules, making the use of around 20 MHz bandwidth optimal (system using less bandwidth can transmit less power, system using more bandwidth don't get more power), and divided the band in 3 chunks, for low power systems (5.2 GHz), medium power (5.4 GHz) and high power (5.6 GHz). Some people have tried to come up with some "etiquette" for the UNII band (stricter set of rules) but they couldn't accommodate the conflicting requirement of all parties.

In the 5 GHz band, because of the availability of more bandwidth, higher speed are possible (10 to 40 Mb/s). But, operating in a higher frequency band increases the noise level, obstacles and walls are more opaque to transmissions (see *chapter 4.8.4*), and a higher bit rate require more SNR (Signal Noise Ratio - see *chapter 4.6.4*), which means a reduced range compared to 2.4 GHZ products, which is bad news.

In summary, in Europe it's HiperLan or nothing. In the USA, the low power chunk of the UNII band (5.2 GHz) is likely to be used by 802.11 at 5 GHz (see *chapter 6.2*) and HiperLan, and people are unlikely to propose yet another standard. The high power chunk will be used by wireless distribution systems, and both type of system will fight for the medium power chunk...

4.3 Spread Spectrum techniques

Spread spectrum is a technique (mainly pioneered by the army) trading bandwidth for reliability. The goal is to use more bandwidth than the system really needs for transmission to reduce the impact of localised interferences (bad frequencies) on the system. Spread spectrum, as it prevents one system to use the full bandwidth capacity, also force independant systems to share the bandwidth (in a mostly fair way). In the 2.4 GHz band, the regulation specifies that systems have to use one of the two main spread spectrum technique: *Direct Sequence* or *Frequency Hopping*.

Which one is better? This is the main technical war between the radio LAN vendors. Everybody, of course, argue that its own technology is better. For now, no one has come with some decisive arguments about the comparative performance and robustness of these two technologies (estimating performance of radio systems is a tricky job). Of course, comparing products doesn't make sense because the performance of a system depend on many other components (the MAC protocol, the signalling rate), the optimisation chosen (performance versus reliability versus cost) and the actual implementation (hum, hum...).

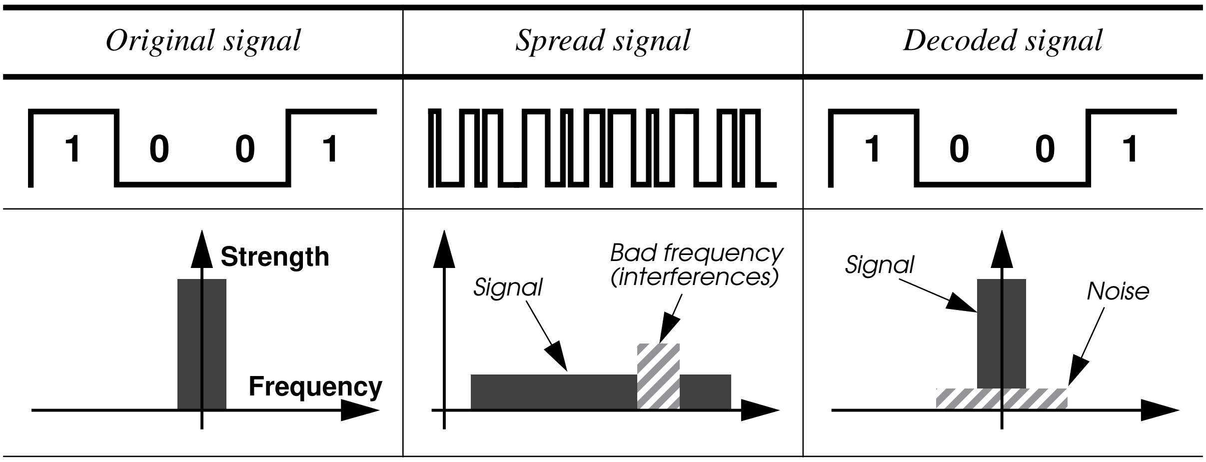
4.3.1 Direct Sequence

The principle of **Direct Sequence** is to *spread* the signal on a larger band by multiplexing it with a signature (the code), to minimise localised interference and background noise.

The system works over a fixed large channel. To spread the signal, each bit of the packet to transmit is sur-modulated by a *code* (a fast repetitive pattern). In the receiver, the original signal is recovered by receiving the whole spread channel (averaging effect) and demodulating by the same code (processing gain). For a 2 Mb/s signalling rate modulated by a 11 chips code (like the Wavelan), the result is a signal spread over 22 MHz of bandwidth.

Any narrowband interferer, because it uses only a small part of the total bandwidth used by the system, will appear much weaker to the Direct Sequence system (I think it will be much clearer if you look at the picture below). Moreover, the demodulator use the same code as the transmitter to match the received signal, which decrease further signals not modulated by the code (this is called the processing gain of the code, 11 chips as used in 802.11 gives in theory a 10 dB processing gain).

Direct Sequence :



Direct Sequence is also the principle used by *CDMA* (Code Division Multiple Access - one of the cellular phone technique), but in CDMA each individual phone channel is given a different code on the same frequency. By having each channel having a orthogonal code and the same received power (so, using power control), it is possible to recover every CDMA channel using its code. The only limit of the scheme is that the noise is proportional of the number of channels (so the degradation with increased capacity is graceful). The configuration also needs to be a star topology (to use power control), which doesn't suit well Wireless LAN.

The spreading with the code produces a faster modulation, therefore a DS modem is quite complicated (it usually require faster circuits and a DSP or equivalent logic for the spreading). One the other hand, the fact of having one single fixed channel (as opposed to Frequency Hopping) eases the task of the higher layers (MAC).

Because it uses a large channel, a Direct Sequence system has only a few channels available in the bandwidth (3 for the *Wavelan* - on different frequencies). Those channels are totally separate (they don't generate interferences on each other). Direct Sequence also offers the possibility to use partially overlapping channels for systems in adjacent areas, increasing slightly the number of channels. But this last solution tends to increase the noise and decrease the performance of the system, because all those systems usually operate with the same code (and not one code per frequency).

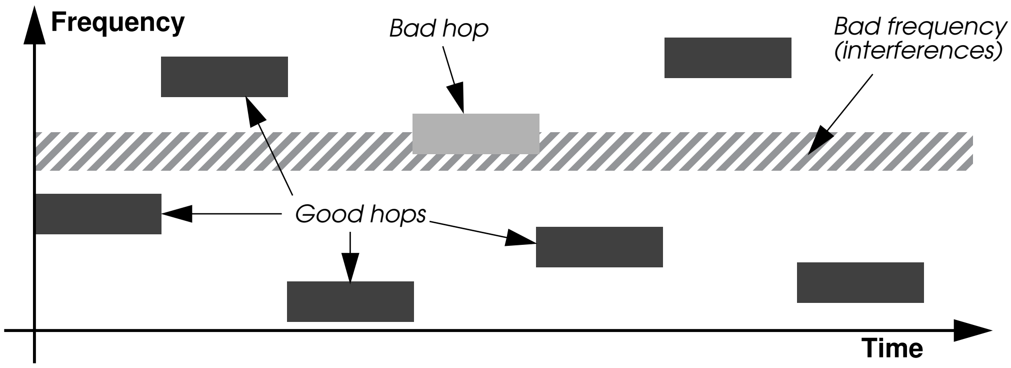
4.3.2 Frequency Hopping

Frequency Hopping uses a set of narrow channels and walk through all of them in sequence. For example, the 2.4 GHz ISM band is divided in 79 channels of 1 MHz. Periodically (every 20 to 400 ms usually), the system *hop* to a new channel, following a predetermined cyclic *hopping pattern*.

The system avoids interferences by never staying on the same channel : if a channel is bad, the system might not be able to use it and just waits for the next good channel. As the pattern makes the whole network hop through all the bandwidth available, the system average the effect of bad channels over the time.

This is where Frequency Hopping has a slight advantage over Direct Sequence : in the very specific case of strong narrow-band interferer present in the band, Frequency Hopping loose some hops but will manage to get some hops on good frequencies. On the other hand, if the noise is stronger than the received signal, there is not much that the Direct Sequence node can do. But, for most interferers at common power levels, it's not totally clear which will give the highest performance (it depends).

Frequency Hopping :



On the other hand, Frequency Hopping introduces more complications at the MAC level : scanning to find the network at the initialisation (a moving target), keeping the synchronisation of the nodes, managing the hops.

This complexity of the MAC has a price in term of performance, and the Frequency Hopping mechanism has some overhead. There is management overhead to manage the synchronisation, and there is some dead time in the transmission when the system hop. In theory, this can be kept to a minimum.

Also, the Frequency Hopping system have to include a process called whitening, to conform to radio transmission constraints, inserting some regular stuff bits in each packets (to avoid long strings of 0 or 1), adding more overhead (on the other a Direct Sequence signal is withened by the Direct Sequence process).

The Frequency Hopping technique can accommodate many more independent systems collocated in the same area than the Direct Sequence technique by using different hopping pattern (up to 15 for the *RangeLan2*). On the other hand, the different hopping patterns of Frequency Hopping will "collide" on the same (or adjacent) frequency from time to time. The collisions of the Frequency Hopping patterns may reduce the throughput significantly : the systems "colliding" on the same (or an adjacent) frequency will have to share the bandwidth between them (see discussions on aggregate throughput in *chapter 5.4.6*).

4.3.3 Comparison...

In term of complexity, the Direct Sequence modem is more complicated than the Frequency Hopping one, and the Direct Sequence has a simpler MAC protocol. With the increasing integration of digital hardware, it doesn't cost much more to implement the specific MAC functionalities required for the Frequency Hopping system, and as the price of the modem is a big portion of a radio LAN and doesn't follow the same cost reduction trends, Frequency Hopping systems will tend to be cheaper.

In term of bandwidth sharing, the two technologies perform really differently. The same is true in term of resistance to interferences (it depend on the strength and pattern of the interferer). Direct Sequence systems tend also to have a lower overhead on the air.

In summary, most vendors are going to Frequency Hopping because of the lower cost and try to convince people that it is better, and vendors having heavily invested in Direct Sequence try to push their raw performance advantage (especially now with 802.11 HR, see *chapter 6.2*), so it is still a kind of religion war.

4.4 Diversity

Diversity is a generic concept of introducing redundancy in the system to overcome noise and to increase the reliability of the system. For example, *spread spectrum* is a type of frequency diversity, using more bandwidth than necessary to avoid bad parts of the spectrum. *Retransmission* is a very usual temporal diversity. *FEC* (Forward Error Correction) is another kind of temporal diversity. Very often, “diversity” is associated with *antenna diversity* only. Antenna diversity is only one form of diversity (a spacial diversity).

Antenna diversity means that the radio device has two (or more) antennas. The transmission conditions on the channel vary a lot over the time. The channel tends to fade in and fade out (see *chapter 4.8.1*), so the device has moment of good reception and moment of bad reception. But, these conditions are also dependant on the spacial position. By having two antennas, even quite close (a few cm), the condition at each antenna is very often totally different. One antenna may give a poor signal and the other a good one, and a few ms later it might be the reverse. So, before receiving each packet, the receiver chooses the best antenna of the two by comparing the signal strengths, and so can avoid most of the fade out periods.

4.5 Directional antennas

Most wireless LANs use omnidirectional antennas, but may offer **directional antennas** in option. Instead of receiving in every directions, the directional antenna favour reception in a more or less narrow angle. The narrower the angle is, the higher the gain is (and the range), because you get rid of more unwanted emissions and background noise in the other directions.

With directional antennas, it is quite common to have a few kilometres of range in line of sight with products in the ISM band. The first problem is that you must of course point each antenna towards the node you intend to communicate with (depending on the angle this needs to be more or less precise). The second problem is that very directional antennas tend to be quite big.

This is why directional antennas are only suited for fixed point to point links (products like **Wireless Bridges**). For most networks where nodes need to talk to different other nodes in different directions and might need to move, omnidirectional antennas are much more practical.

Sectored antennas are very similar to directional antennas, and heavily used in cellular phone base stations. A set of wide angle directional antenna are assembled on a vertical pole, each one covering one portion of the horizon (a sector, for example 3 antennas 120 degrees wide). When talking to a specific node, the base station just select the sector of the sectored antenna that cover this node, giving the benefit of directionality without sacrificing the coverage.

People are also investigating **beam forming antennas**. This is an adaptive directional antenna, using a set of unidirectional antennas and interferometry to enhance the signal. Basically, by adding all the signal of the different antennas with specific offset (to compensate propagation delay), it is possible to aim the system towards a specific direction and have the same benefit as directional antenna. As this system is adaptive and dynamic, it could be used for Wireless LANs

4.6 Range issues

The **propagation** of radio transmissions is influenced by many factors. Walls and floors tend to decrease and reflect the signal, and background noise makes it more difficult to demodulate. In a typical environment, all the shadows due to obstacles and reflections on the walls create a very unpredictable quality of transmission for each specific location. The channel quality also vary quite a lot over the time (*fading*, see *chapter 4.8.1*) because the environment is not static.

Because of the way radio transmissions are affected by the environment in such a complex way, it is quite difficult to predict the comportment of the system and to define a **range**. You will have some good, fair and bad area/period, the closer the two devices are the more likely they are to be in a good one.

Most vendors attempt to define a range for their products, which is the average maximum distance in usual operating conditions between two nodes (diameter of a **cell** - radio neighbourhood). Some even give different ranges for different typical environments. For example: open environment (no obstacles), semi-open (cubicles) and closed (real walls).

But there is no standard and common operating procedure to measure a range (except in free space, but this is useless), so we can't really compare the different products from the ranges as indicated in their data-sheets, and you must take these values with a bit of caution.

If you want to compare products in term of range performance, you must look closely at the *transmitted power* and *sensitivity* values. These are some measurable characteristics of the hardware which indicate the performance of the product in that respect. In fact, I would also recommend to do some benchmark of different products in your own environment to get a better idea of what coverage you can expect.

4.6.1 Transmitted power

The **transmitted power** is the strength of the emissions measured in Watts (or milliWatts). We have already seen that the regulations limit this power (see *chapter 4.1*). Products having a high transmit power will also be likely to drain the batteries faster. But, having a high transmit power will help to emit signals stronger than the interferers in the band (and other systems).

Having a strong transmitted power has some drawback for *frequency reuse*. This means that if you want to put many different networks in areas close to each other, they will tend to pollute each other. With less transmitted power you can make smaller cells. This is why some product may allow to select different transmitted powers.

4.6.2 Sensitivity

The **sensitivity** is the measure of the weakest signal that may be reliably heard on the channel by the receiver (it is able to read the bits from the antenna with a low error probability). This indicates the performance of the receiver, and the lower the value the better the hardware (higher in absolute value). The figure is given in dBm, the magic formula to transform power in Watts to dBm is : *P dBm = 30 + 10.log(P W)*. Usual values are around -80 dBm (the lowest, the better, for example -90 dBm is better).

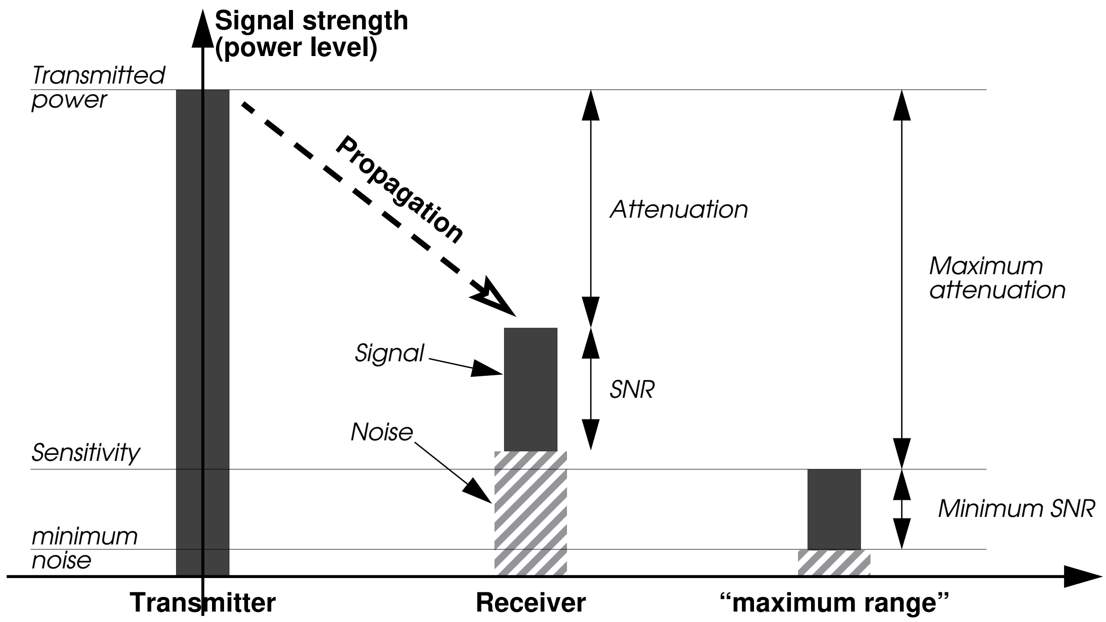
One problem is that all manufacturer and standards use the same reference to define sensitivity. 802.11 specify the sensitivity as the point when the system suffer from 3 % of packets losses (for packets of 400 Bytes in a Gaussian channel). Some products use 50 % packet losses as the definition of sensitivity, which of course gives a better number. The use of a Gaussian channel also gives better numbers (the use of a Rayleigh Fading channel with antenna diversity would give results approximately 7 dB worse).

4.6.3 Attenuation

Knowing those two values, you may calculate the maximum possible **attenuation** of the packets (this is the difference between the two values, in dB). The larger the maximum possible attenuation, the larger the range. For a 100 mW system with a -80 dBm sensitivity, we have 100 dB maximum attenuation.

The attenuation is the decrease of signal strength between the transmitter and the receiver. In the air, the attenuation is simply proportional to the square of the distance. If you know exactly the composition of the signal paths between the two nodes (distance in the air, type of obstacles, reflections ...), you may calculate the attenuation. But usually it is quite tricky to determine the attenuation as a function of the distance, especially that the signal may be the composite from different propagation paths (see *chapter 4.8.4*). Moreover, the variation in the environment make the attenuation change over the time (see *chapter 4.8.1*).

Propagation and Range :



Because of this non straightforward relationship, knowing the maximum possible attenuation won't give you the maximum range, but just a feeling. The only safe thing is that products with a greater maximum possible attenuation are very likely to have a larger range.

4.6.4 Signal to noise ratio (SNR)

In the case of multirate systems, I've been talking of **Signal to Noise ratio** (SNR). The sensitivity is in fact closely linked to the minimum SNR of the modem. The SNR defines the difference of power in the receiver between a valid signal and a noise. To be able to decode successfully the received signal, the receiver needs a minimum SNR (i.e. the signal not too much polluted by the noise). This minimum SNR depends on the quality of the receiver hardware and the modulation chosen (see *chapter 4.7.1* on multi rate systems).

So, the link between sensitivity and minimum SNR is quite obvious. If you add the minimum SNR to the background noise in the receiver (hardware noise and background noise on the channel), you will find the sensitivity. So, having a low sensitivity means also a low minimum SNR, so the ability to receive reliably packets with potentially higher interference strength, which explain why the sensitivity is such an important performance characteristic.

4.7 Modulations

The main job of the radio modem is to transform bits into modulations of the radio waves, but there is many way to do that. Most systems use a carrier (a base frequency) and modulate it. The simplest way is to modulate the strength of the signal (Amplitude Modulation), but as the attenuation of the channel is usually not constant (see *chapter 4.6.3*), this lead to poor performance. Most modern systems modulate either the frequency of the signal or the phase of the signal (frequency offset), which gives much greater performance.

4.7.1 Multi-rate systems

If you want a better throughput, the most simple way is to use more bandwidth. The problem is that the ISM spread spectrum regulations limits the amount of bandwidth usable (1 MHz channels for Frequency Hopping). Also, in most hardware the filters used to recover the signal are fixed, so the channel width is fixed. This limit the rate of symbols that you can use (1 Mbauds for Frequency Hopping).

So, how could some Frequency Hopping systems offer 3 Mb/s in 1 MHz channels ? The use of more **complex modulation** schemes allows to overcome this limitation. For example, the standard 2FSK allows to put 1 bit per symbol, whereas 4FSK allows 2 bits per symbols, doubling the signalling rate.

Of course, there is a drawback : a more complex modulation scheme is less robust and will require a higher received Signal to Noise Ratio to work (SNR - see *chapter 4.6.4*). When going from 2FSK to 4FSK, each time the receiver reads a symbol, instead of having to distinguish two fairly separated values, now it has to distinguish 4 closer to each other (see *chapter 4.7.2*). More complex modulations stuff even more values in the same space, but then the slightest perturbation of the signal (noises) will make the receiver reads the wrong value for the symbol.

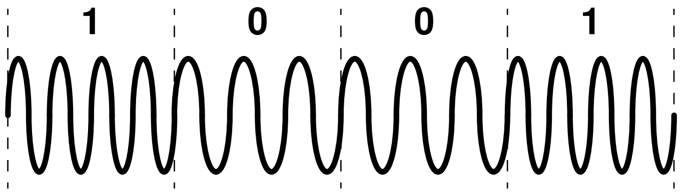
So, we have the choice between a high speed modulation which requires strong received signal and a slower modulation which works even on weak signals. In other words, the higher the signalling rate, the shorter the range.

Because users want both range and speed, some vendors have build some systems using multiple levels of modulations, changing automatically from the fast modulation to the robust one depending on the channel conditions (when a packet fail, the rate is automatically reduced). This introduces a bit of overhead and complexity, but the system offer a much better performance characteristic (range or speed).

4.7.2 2FSK and 4FSK

2FSK (Frequency Shift Keying) is the simplest form of frequency modulation. Basically, the system use two different frequencies for the values 0 and 1 of each bit. For example, if *B* is the base frequency (the carrier) and *d* the carrier deviation, each time the system want to transmit a 0 it creates a waveform of frequency *B-d* (a symbol), and each time it want to transmit a 1 it creates a waveform of frequency *B+d*. The receiver just need to measure the deviation of the signal to the reference frequency *B* to know which value of the bit was transmitted.

Frequency Modulation (2FSK) :



Measuring this deviation is not easy, because each symbol is very short in time : the transmitter change it for every bit to transmit at the speed given by the baudrate. The receiver needs of course to know when the bits are transmitted, which require timing synchronisation on the received signal. The carrier deviation has to be chosen carefully to enable enough differentiation between the two symbols but to have the signal generated fitting in the band allocated to it (usually around one hundred kHz for a 1 MHz channel at 2.4 GHz).

As mentioned above, it is possible to put more than one bit per symbol (see *chapter 4.7.1*), like using **4FSK**. 4FSK use 4 different symbols having 4 different carrier deviation, *B+1/2d, B-1/2d, B+3/2d* and *B-3/2d*, each symbol is mapped to a combination of two bits (00, 01, 10, 11).

Note that the difference in frequency between each symbol for 4FSK is smaller than for 2FSK, to allow the signal to fit in roughly the same channel width. Between each symbol, the difference is only *d* for 2FSK, instead of *2d* for 4FSK, which explains why 4FSK is more sensitive and requires a better SNR (see *chapter 4.6.4*).

4.7.3 802.11 HR (11 Mb/s)

When 802.11 was eventually released, 1 and 2 Mb/s was no longer considered as decent speed for Wireless LAN and people were already talking of using the 5 GHz band for higher throughput (HiperLan and 802.11 at 5 GHz). However, the migration from 2.4 GHz to 5 GHz requires to change all nodes and doesn't provide backward compatibility (it's mot the same frequency band, so a new modem is necessary).

Therefore, people producing 2.4 GHz products tried to find way to extend the life of their technology (mostly Harris and Lucent). They cheated with the Spread Spectrum rules, and got away with it, enabling them to offer 5 and 11 Mb/s systems.

Basically, a DS system generate signal which occupy around 22 MHz of bandwidth. They designed their 11 Mb/s system to generate signal similar to a standard DS system. Then, they went to the FC and claimed that as their new system was generating the same type of signal as a DS system, it's impact on other systems in the band was the same, so it should be authorised as well. After a bit of negociation, the FCC did accept this extension of the rule. Note that some FH vendors also tried to get 5 MHz FH channels in the 2.4 GHz band but failed to obtain it.

Lucent came up with the simplest solution, PPM (Pulse Position Modulation), which is included in their "Turbo" line of products, offering 5 and 10 Mb/s. PPM simply shift the code used in the DS modem, each position can encode some more bits. PPM is simple, cheap, but low performance.

Harris tried MBOK (M-ary Bi-Orthogonal Keying), offering 5.5 Mb/s and 11 Mb/s, which is a more complex modulation than PPM, so more expensive and more robust. The signal produced by the transmitter is also less similar to a DS signal.

They both went back to the 802.11 group, but neither wanted to adopt the system of the other. So, they settled down on yet another modulation, **CCK** (Complementary Code Keying), which eventually got adopted for the 802.11 HR standard and approved by the FCC. CCK is the most complex of the 3 modulations, offering better performance, but higher cost, and signals even less similar to the original DS signals.

802.11 HR offer 11 and 5.5 Mb/s rate (using the CCK modulation) and is backward compatible with original 802.11 DS systems. However, the higher bit rate require a higher SNR, which reduce the range significantly. Note as well that because of backward compatibility most of the underlying protocol is still designed for the 1 Mb/s standard (headers and management frames are 1 Mb/s, contention window size is still based on 1 Mb/s systems), which mean that at higher rate the overhead of the system is much higher.

4.7.4 OFDM

People building high speed system like *HiperLan* were complaining that adding to their products an Equaliser necessary to combat delay spread (see *chapter 4.8.4*) was a major cost. So, they invented a new technique to get similar or better performance at lower cost, called **OFDM** (Orthogonal Frequency Division Multiplex).

Using equalisation is a post-processing technique, which tries to overcome delay spread by brute force. OFDM is a pre-processing technique, where the signal transmitted on the band is prepared in such a way that the impact of delay spread is reduced.

Delay spread is damaging because the symbol time is very short, so OFDM will only use large symbol time. However, by increasing the symbol time we reduce the bit-rate. To overcome this constraint, OFDM transmit the symbols no longer serially but in parallel! This way, we have very high bit rate with large symbol time.

OFDM use a set of subcarrier frequencies, the frequencies being orthogonal. Each subcarrier is modulated individually, the bit rate and signal strength of each subcarrier can be adapted to get maximum performance of the system (we put more bits on the good subcarriers and less on the bad ones). Then, the system splits the bits to transmit between the subcarriers, each subcarrier is modulated and then combined to produce the transmitted signal (using a Fast Fourrier Transform).

The main drawback of OFDM is that it require a greater frequency accuracy (we traded timing accuracy to frequency accuracy). As the OFDM signal contains many subcarrier very close to each other in frequency, the system must be very accurate to match all of them.

The first use of OFDM was in the HiperLan II standard (see *chapter 6.4*), but since 802.11 at 5 GHz has adapted a very similar modulation (see *chapter 6.2*).

4.8 Interferences and noises

In the previous section we have examined what does affect the range performance of a system. Unfortunately, other phenomenon on the radio waves affect the performance of a system (even if they may not reduce the range), and all kind of interferences and background radio noises will impact the system.

4.8.1 Fading

Fading defines all the temporal variations of the signal attenuation due to its propagation in a real environment like an office or a house. The radio signal interact in various way with the environment (see *chapter 4.6* and *chapter 4.8.4*), so vary a lot with the environment configuration. Moving a few centimetres can make a big different in signal quality (see *chapter 4.4*).

Moreover, the environment is not static, humans are moving, things are moving, and the nodes may be moving themselves. All these small movements may produce important **variations in time** in the attenuation of the signal. For example the propagation between two nodes may alternate from poor to good on a packet basis.

People usually describe the pattern of attenuation with a *Rayleigh fading model* (case where there is no line of sight) or a *Ricean model* (line of sight + additional paths). The main consequence is that transmission errors on the channel tend to be clustered and are anything but following a Gaussian distribution.

Fading cause transmissions errors that need to be overcome by the system. Of course, recovering from these error will add overhead. The greater the range the greater will be the impact of the fading and the system will degrade with higher range until it loose communication.

The most efficient technique to overcome the effect of fading is *antenna diversity* (see *chapter 4.4*).

4.8.2 Microwave oven and other interferers

As we have mentioned earlier, Wireless LANs tend to be implemented in the unlicensed bands, which adds more constraints. The vast majority of the Wireless systems (cellular phone, telecoms, aviation, military ...) are designed for dedicated radio bands, so benefit from an absence of interferers in the band they are using. This is not the case for Wireless LANs, they have to cope with the emissions of other systems.

The deployment of unlicensed systems is totally uncoordinated. So, other radio systems operating in the area do create interferences. This includes other Wireless LANs, cordless phones (900 MHz and now 2.4 GHz) and other communication systems.

The 2.4 GHz band is also the frequency where water molecules resonate, so is used for microwave oven. Domestic microwave oven (the one used to heat food in the kitchen) generates a limited amount of interferences, the various regulations limit the power of the radiation they can leak to less than 1W, they emit periodic short bursts and pollute only a limited portion of the 2.4 GHz band. Commercial microwave ovens (for example a huge dryer in a paper factory) generate much more interferences.

The result of interferences is that packets collide with interference signal and can be received corrupted. If the SNR between the packet and the interferer is high enough (see *chapter 4.6.4*), the receiver can “capture” the packet, otherwise it is corrupted.

Most Wireless LANs cope very well with interferers, in fact usually much better than cordless phones, but interferences do reduce performance.

4.8.3 FEC (Forward Error Correction)

The most obvious way to overcome transmission errors is to use **FEC**. FEC goes further than CRC which just detects errors, FEC adds in every transmission some additional redundancy bits. Depending on the number of bits added and the FEC code used (the strength of the code), this allows to repair a certain number of errors in the transmission.

FEC has been used with success in many systems, and the **Turbo Codes** are probably the most efficient one : they are very close to the Shannon limit in a Gaussian channel. In other world, if the error follow Gaussian distribution (and the parameters are known), there is a turbo code nearly optimal giving the highest throughput in this channel.

Unfortunately for us, errors on a radio channel (for Wireless LAN) follow a fading model and are clustered. This means that most of the time the signal is strong, so the packet is error free, but when the signal is weak the packet contains lots of error. Interferences has roughly the same effect as fading, either the packet is collision free so intact, or when a collision occur most of the packet is corrupted.

To correct all those errors in corrupted packets, it would require a very strong FEC code. Unfortunately, this code would add lots of redundancy bits, so lots of overhead. A normal FEC code would add less overhead, but be useless with the correct packets and inefficient with the highly corrupted packets.

So, for Wireless LANs, using FEC tends to be ineffective against fading and interferers, and no Wireless LAN do implement FEC. A much better solution is to use **retransmissions** (just retransmit the original packet in case of errors - some form of packet scheduling and retransmission has been proven to be nearly optimal in Rayleigh fading channels). This is usually implemented at the MAC level (see *chapter 5.2.1*).

However, in a few case FEC might be needed in Wireless LANs. Some receivers, either due to poor implementation or specific design (like having an Equaliser), generate random (Gaussian) errors, and might benefit from FEC.

4.8.4 Multipath and delay spread

Radio waves reflect or diffract on obstacles, and are attenuated differently by different materials. This is exactly like light, which goes through glass, is reflected by mirrors and stop by most obstacles, except that much more materials are transparent or reflector to radio than to light.

In a real environment like an office or a house, there is a lot of surface reflecting radio (walls, ceilings, metal), being semi-transparent to radio (walls, ceilings, humans) or opaque to radio (metal). This gives trouble estimating the range of the system (see *chapter 4.6*). This also mean that the signal received at a node may come from different directions (depending on reflections on the environment) with different strength (depending on attenuations), and the receiver sees only the combinations of all these reflections. This phenomenon is called **multipath**.

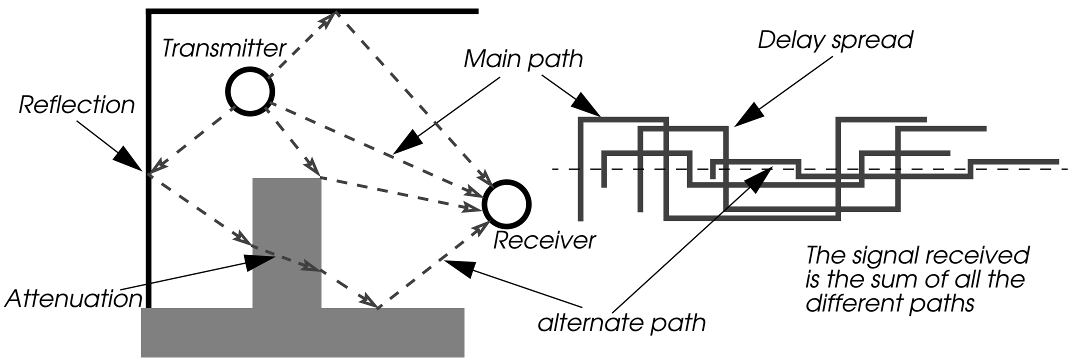
Most of the time, multipath is good, because the addition of all the reflections of the signal increase its strength. The main effect of multipath is that *range* is very difficult to evaluate (see *chapter 4.6.3*) and the receiver experiences *fading* (see *chapter 4.8.1*).

But, the main problem of multipath is that it creates **delay spread**. Depending on the number of reflections and the propagation speed in different signals, all these signals don't arrive exactly at the same time at the receiver. It's like the "echo" you may hear in the mountains, the signal going directly will be faster than one reflecting twice on the walls.

Of course, as radio propagate at the speed of light, those difference are very small (below the microsecond). But, when the bitrate of the system increases, those time differences becomes significant with regards to the symbol time (see *chapter 4.7.2*), to the point of creating destructive interferences (the current symbol will be corrupted by the echo of the previous symbols).

Bit rate lower than 1 Mb/s are relatively immune to delay spread problems (the symbol time is 1 µs and higher), but as the bit rate increase above 1 Mb/s the effect of delay spread increases. It is considered that systems faster than 5 M/s should have some technique to overcome delay spread.

Multipath and Delay Spread :



The main technique to overcome delay spread is using an **Equaliser**. An equaliser is a big digital circuit that try to estimate the different components of the signals. The equaliser need to be trained (packets includes a specific well known training sequence) to determine what are the different path, their relative timings and strength. Then, the equaliser separate the different components of the signal and recalculate the signal removing the delay spread.

The main disadvantage of Equaliser is that they are expensive. Recently, some standards are starting to use **OFDM** (see *chapter 4.7.4*), which is a clever modulation technique minimising the impact of delay spread.

5 The MAC level (link layer)

This section of the document focus on the next layer up, the link layer. This mostly comprise the **MAC** (Medium Access Control) protocol. Different MAC protocols and techniques are presented.

5.1 Main channel access mechanisms

The main job of the MAC protocol is to regulate the usage of the medium, and this is done through a channel access mechanism. A **channel access mechanism** is a way to divide the main resource between nodes, the radio channel, by regulating the use of it. It tells each node when it can transmit and when it is expected to receive data. The channel access mechanism is the core of the *MAC protocol*. In this section, we describe

TDMA, CSMA and *polling* which are the 3 main classes of channel access mechanisms for radio.

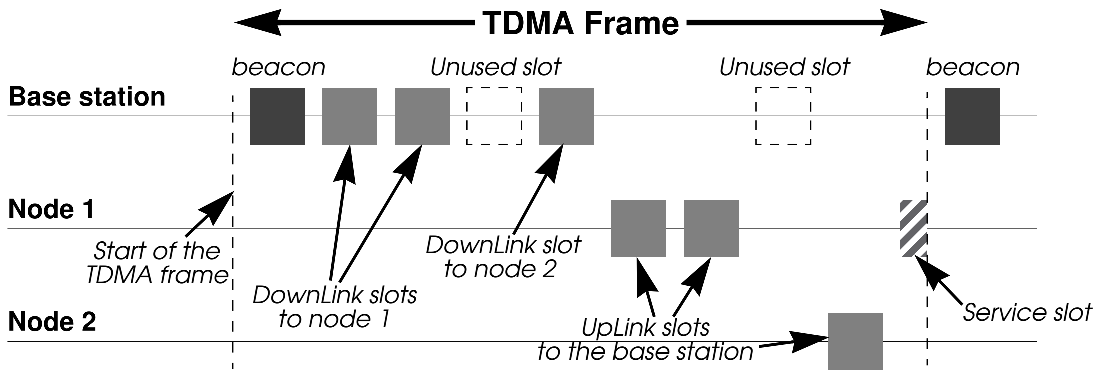
5.1.1 TDMA

In this chapter, we discuss TDMA as a channel access mechanism and not its applications and protocols based on it.

TDMA (Time Division Multiplex Access) is very simple. A specific node, the **base station**, has the responsibility to coordinate the nodes of the network. The time on the channel is divided into *time slots*, which are generally of fixed size. Each node of the network is allocated a certain number of slots where it can transmit. Slots are usually organised in a frame, which is repeated on a regular basis.

The base station specify in the beacon (a management frame) the organisation of the frame. Each node just needs to follow blindly the instruction of the base station. Very often, the frame is organised as downlink (base station to node) and uplink (node to base station) slots, and all the communications goes through the base station. A service slot allows a node to request the allocation of a connection, by sending a connection request message in it (see *chapter 5.2.4*). In some standards, uplink and downlink frames are one different frequencies, and the service slots might also be a separate channel.

TDMA channel access mechanism :



TDMA suits very well phone applications, because those application have very predictable needs (fixed and identical bit rate). Each handset is allocated a downlink and a uplink slot of a fixed size (the size of the voice data for the duration of the frame). This is no surprise why TDMA is used into all cellular phone standards (GSM in Europe, TDMA and PCS in the USA) and cordless phone standards (DECT in Europe). TDMA is also very good to achieve low latency and guarantee of bandwidth (where CSMA/CA is quite bad).

TDMA is not well suited for data networking applications, because it is very strict and inflexible. IP is connectionless and generates bursty traffic which is very unpredictable by nature, while TDMA is connection oriented (so it has to suffer the overhead of creating connections for single IP packets). TDMA use fixed size packets and usually symmetrical link, which doesn't suit IP that well (variable size packets).

TDMA is very much dependant of the quality of the frequency band. In a dedicated clean band, as it is the case for cellular phone standard, TDMA is fine. But, because of it's inflexibility, and because it doesn't really take care of what's happening on the channel, TDMA can't cope and adapt to the bursty interference sources found in the unlicensed bands (unless a retry mechanism is put on top of it).

5.1.2 CSMA/CA

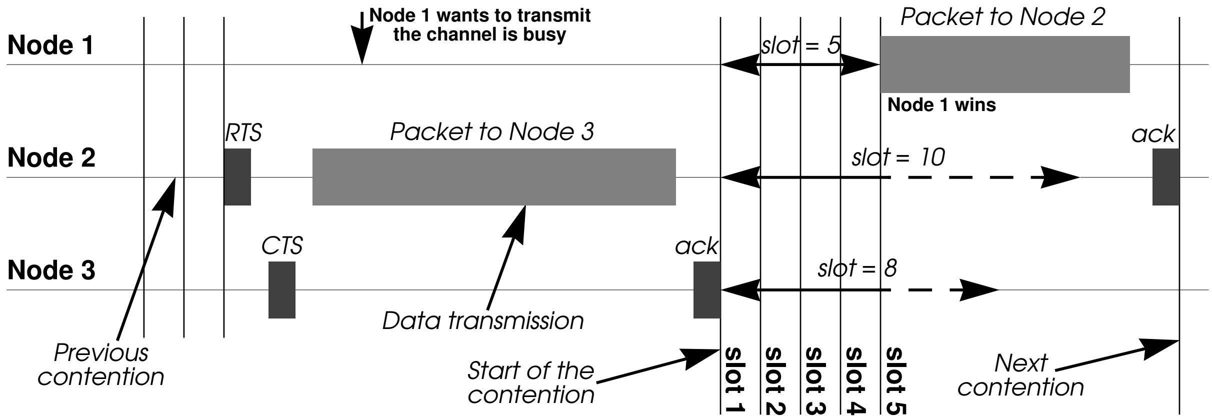
CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) is the *channel access* mechanism used by most wireless LANs in the ISM bands. A channel access mechanism is the part of the *protocol* which specifies how the node uses the medium : when to listen, when to transmit...

The basic principles of CSMA/CA are *listen before talk* and *contention*. This is an *asynchronous* message passing mechanism (connectionless), delivering a best effort service, but no bandwidth and latency guarantee (you are still following ?). It's main advantages are that it is suited for network protocols such as TCP/IP, adapts quite well with the variable condition of traffic and is quite robust against interferences.

CSMA/CA is fundamentally different from the channel access mechanism used by cellular phone systems (see *TDMA* in *chapter 5.1.1*).

CSMA/CA is derived from CSMA/CD (Collision Detection), which is the base of *Ethernet*. The main difference is the *collision avoidance* : on a wire, the transceiver has the ability to listen while transmitting and so to detect collisions (with a wire all transmissions have approximately the same strength). But, even if a radio node could listen on the channel while transmitting, the strength of its own transmissions would mask all other signals on the air. So, the protocol can't directly detect collisions like with *Ethernet* and only tries to avoid them.

CSMA/CA channel access mechanisms :



The protocol starts by listening on the channel (this is called *carrier sense*), and if it is found to be idle, it sends the first packet in the transmit queue. If it is busy (either another node transmission or interference), the node waits the end of the current transmission and then starts the **contention** (wait a random amount of time). When its contention timer expires, if the channel is still idle, the node sends the packet. The node having chosen the shortest contention delay wins and transmits its packet. The other nodes just wait for the next contention (at the end of this packet). Because the contention is a random number and done for every packets, each node is given an equal chance to access the channel (on average - it is statistic).

As we have mentioned, we can't detect collisions on the radio, and because the radio needs time to switch from receive to transmit, this contention is usually **slotted** (a transmission may start only at the beginning of a slot : 50 µs in 802.11 FH and 20 µs in 802.11 DS). This makes the average contention delay larger, but reduces significantly the collisions (we can't totally avoid them).

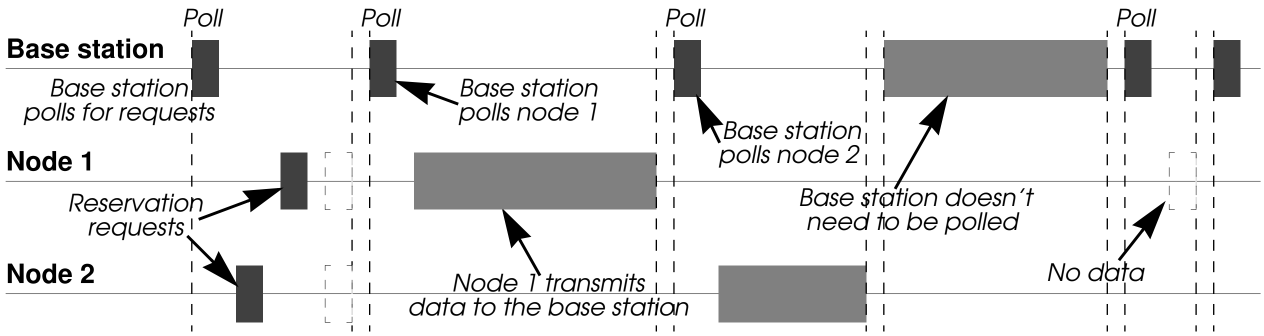
5.1.3 Polling MAC

Polling is the third major channel access mechanism, after *TDMA* and *CSMA/CA* (see *chapter 5.1.1* and *chapter 5.1.2* respectively - There exist also Token Ring, but I guess that nobody would be crazy enough to implement it on a radio link). The most successful networking standard using polling is 100vg (IEEE 802.12), but some wireless standard are also using it. For example, *802.11* offers a polling channel access mechanism (Point Coordination Function) in addition to the CSMA/CA one.

Polling is in fact in between TDMA and CSMA/CA. The base station retains total control over the channel, but the frame content is no more fixed, allowing variable size packets to be sent. The base station sends a specific packet (a poll packet) to trigger the transmission by the node. The node just wait to receive a poll packet, and upon reception sends what it has to transmit.

Polling can be implemented as a connection oriented service (very much like TDMA, but with higher flexibility in packet size) or connection less-service (asynchronous packet based). The base station can either poll permanently all the nodes of the network just to check if they have something to send (that is workable only with a very limited number of nodes), or the protocol use reservation slots (see *chapter 5.2.4*) where each node can request a connection or to transmit a packet (depending is the MAC protocol is connection oriented or not).

Polling channel access mechanism :



In the case of 100vg, the polling mechanism doesn't use any bandwidth (it's done out of band through tones), leading to a very efficient use of the channel (over 96 % user throughput). For 802.11 and wireless LAN, all the polling packets have to be transmitted over the air, generating much more overhead. More recent system use reservation slots, which is more flexible but still require significant overhead.

As CSMA/CA offers ad-hoc networking (no need of a base station) and similar performance, it is usually preferred in most wireless LANs. For example, most 802.11 vendors prefer to use the distributed mode (CSMA/CA) over the coordinated mode (polling).

5.1.4 Reservation protocols and WATM

The most interesting feature of protocols based on TDMA or Polling mechanism is that the Base Station has absolute control of the traffic and can guarantee bandwidth and latency for applications that require it. Sceptics might wonder what can be guaranteed anyway in an environment open to interferers and without deployment control (see *chapter 4.1*), but that's another topic of discussions.

The guarantee of bandwidth is essential for people deploying Wireless Distributions Systems (also called Last Mile Delivery Systems), like replacing the cable between your house and your ISP with wireless. Those people want to be able to restrict and segregate users and guarantee fairness. Standards such as HiperLan II (Broadband Radio Access Network project - see *chapter 6.4*) is aiming at those usages.

The basic idea is to put ATM (Asynchronous Transfer Mode) over radio, as ATM implement all the Quality Of Service features that they are dreaming off. The network is centrally managed (so uses TDMA or Polling mechanism with reservation slots), the base station implement a call admission control (accept or reject new ATM circuits) and scheduler (prioritise and send ATM cells) to guarantee the quality of service requested. On top of the MAC, all the usual ATM layers are needed (virtual circuits, segmentation/ reassembly, IP adaptation...), as well as some specific mobile features (to manage roaming).

Unfortunately, radio transmission has a lot of overhead (like large synchronisation field and headers) which is somewhat incompatible with the small ATM cells. The main benefit of ATM small cells is to allow very efficient switching, but this is not needed over radio. At the end of the day, WATM doesn't resemble at all to ATM; ATM uses individual channel for each node and is asynchronous, whereas WATM uses a shared medium and is totally synchronous.

5.2 MAC techniques

We have described the main principle of CSMA/CA (see *chapter 5.1.2*), but most MAC protocols use additional techniques to improve the performance of CSMA/CA.

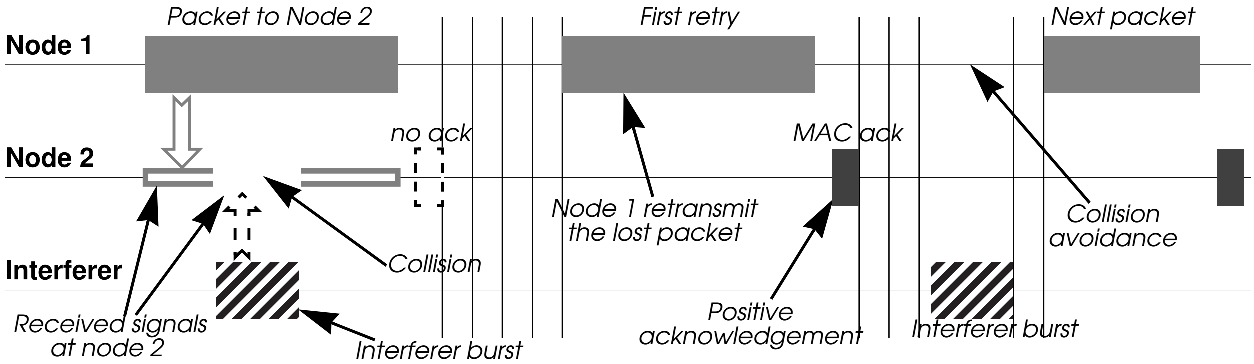
5.2.1 MAC retransmissions

As we have seen in the previous chapter, the main problem of the *CSMA/CA protocol* is that the transmitter can't detect collisions on the medium. There is also a higher error rate on the air than on a wire (see *chapter 4.8*), so a higher chance of packets being corrupted. TCP doesn't like very much packet losses at the *MAC layer* (see TCP and packet losses problem - *chapter 5.4.5*). Because of that, most MAC protocols also implement **positive acknowledgement** and **MAC level retransmissions** to avoid losing packets on the air.

The principle is quite simple: each time a node receives a packet, it sends back immediately a short message (an ack) to the transmitter to indicate that it has successfully received the packet without errors. If after sending a packet the transmitter doesn't receive an ack, it knows that the packet was lost, so it will retransmit the packet (after contending again for the medium, like in Ethernet).

Most MAC protocols use a stop and go mechanism, they transmit the next packet of the queue only if the current packet has been properly acknowledged (no sliding window mechanism like in TCP). The rationale is that it makes the protocol simpler, minimise latency and avoid desenquencing packets (something that TCP doesn't like as well).

MAC retransmissions in CSMA/CA :



The acks are “embedded” in the MAC protocol, so they are guaranteed not to collide (the contention starts after the ack - see figure). These acks are very different from the TCP acks, which work at a different level (and on a different time frame). Of course, broadcast and multicast packets are not acknowledged, so they are more likely to fail...

If all modern Wireless LAN protocols implement this essential feature, some old products may lack it. Wireless WAN protocols (like satellite links) don’t implement that either, because the round trip delay in their case is so long that by the time they would receive the ack they could have sent another packet. If your Wireless LAN doesn’t implement MAC level retransmissions, all is not lost : students of Berkeley have created a protocol called *snoop* (see at ftp://daedalus.cs.berkeley.edu/pub/snoop/) which filters the TCP acks and retransmits the lost packets before TCP even notices that they are lost (this is still a link level retransmission, but done just over the MAC).

5.2.2 Fragmentation

The radio medium has a higher *error rate* than a wire. We have explained in the previous chapter that it was why most products were including MAC level retransmissions to avoid losing packets.

MAC level retransmissions solve this problem, but is not really performant. If the packet to transmit is long and contains only one error, the node needs to retransmit it entirely. If the error rate is significantly high, we could come to some situation were the probability of error in large packet is dangerously close to 1 (we can’t fit a packet between the bursts of errors due to fading or interferers), so we can’t get packet through.

This is why some products use **fragmentation**. Fragmentation is sending the big packets in small pieces over the medium. Of course, this adds some overhead, because it duplicates packet headers in every fragments. Each fragment is individually checked and retransmitted if necessary. The first advantage is that in case of error, the node needs only to retransmit one small fragment, so it is faster. The second advantage is that if the medium is very noisy, a small packet has a higher probability to get through without errors, so the node increases its chance of success in bad conditions.

5.2.3 RTS/CTS

In the chapter about range (*chapter 4.6*), we have seen that the main effect of transmission on radio waves is the attenuation of the signal. Because of this attenuation, we have very commonly a problem of *hidden nodes*.

The hidden node problem comes from the fact that all nodes may not hear each other because the attenuation is too strong between them. Because transmissions are based on the carrier sense mechanism, those nodes ignore each other and may transmit at the same time. Usually, this is a good thing because it allows *frequency reuse* (they are effectively in different cells).

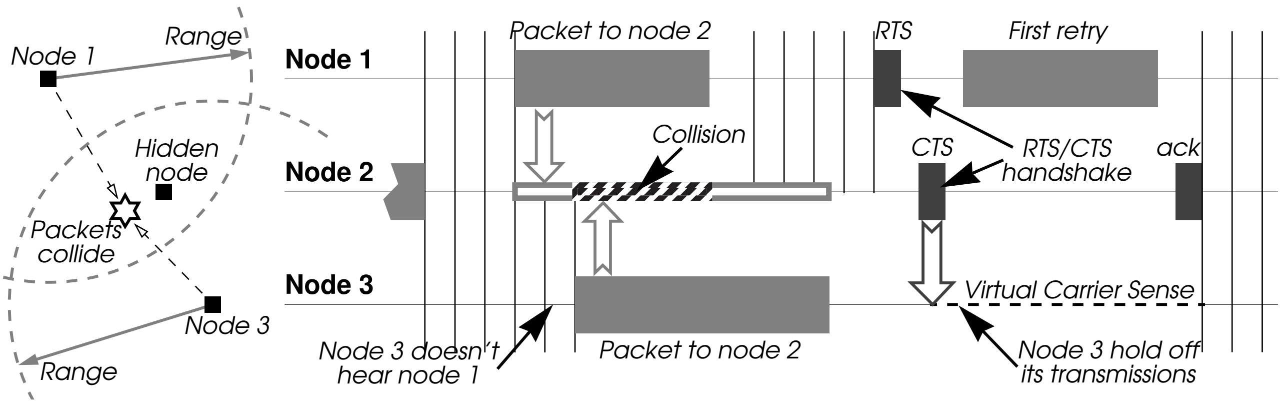
But, for a node placed in between, these simultaneous transmissions have a comparable strength and so collide (in its receiver). This node could be impossible to reach because of these collisions.

The fundamental problem with carrier sense only is that the transmitter tries to estimate if the channel is free at the receiver with only local information. The situation might be quite different between those two locations.

An simple and elegant solution to this problem (proposed by Phil Karn in his MACA protocol for AX.25) is to use **RTS/CTS** (Request To Send/Clear To Send). RTS/CTS is a *handshaking* : before sending a packet, the transmitter sends a RTS and wait for a CTS from the receiver (see figure below). The reception of a CTS indicates that the receiver is able to receive the RTS, so the packet (the channel is clear in its area).

At the same time, every node in the range of the receiver hears the CTS (even if it doesn't hear the RTS), so understands that a transmission is going on. The nodes hearing the CTS are the nodes that could potentially create collisions in the receiver (assuming a symmetric channel). Because these nodes may not hear the data transmission, the RTS and CTS messages contain the size of the expected transmission (to know how long the transmission will last). This is the *collision avoidance* feature of the RTS/CTS mechanism (also called *virtual carrier sense*) : all nodes avoid accessing the channel after hearing the CTS even if their carrier sense indicate that the medium is free.

RTS/CTS and hidden nodes in CSMA/CA :



RTS/CTS has another advantage : it lowers the overhead of a collision on the medium (collisions are much shorter in time). If two nodes attempt to transmit in the same slot of the contention window, their RTS collide and they don't receive any CTS, so they loose only a RTS, whereas in the normal scenario they would have lost a whole packet.

Because the RTS/CTS handshaking adds a significant overhead, usually it is not used for small packets or lightly loaded networks.

5.2.4 Reservation and service slots

One of the main problem of TDMA and Polling protocol is for the base station to know when the nodes want to transmit. In CSMA/CA, each node simply waits to win a contention, so this problem doesn't exist. However, TDMA and Polling usually require a **service slot** or **reservation slot** mechanism.

The idea is to offer a period of time where nodes can contend (compete) and send to the base station some information about their traffic requirements (a reservation request packet), this period of time coming at regular interval (the remaining of the time, nodes just obey the base station normally). The base station feeds the reservation requests to its *scheduling algorithm* and decides the main frame structure (when each node will transmit). This period of time for sending reservation requests is either called service slot (if it is use for more purpose like cell location and roaming) or reservation slot (if it is use only to request a transmission or connection).

If the MAC is connection oriented, the rate of new connection is low, so usually a single service slot is enough (see figure in *chapter 5.1.1*). If the MAC is packet oriented, the rate of requests is higher, so usually the protocol offer many reservation slots together (see *chapter 5.1.3*). Nodes use a simple *Aloha protocol* in the slots : they transmit, and if it fail (collision with other requests or medium errors) they backoff a random number of slots before retrying.

Protocols which use many different channels, such as cellular phone, can even have a dedicated service channel separate from other transmissions, instead of multiplexing service requests with the data traffic.

5.3 Network topology

The topology of Wireless LAN is very different from traditional LANs. The connectivity is limited by the range, so we usually don't have complete coverage (some node may not see each other). This breaks some assumptions of higher layers. To overcome this, either the network is divided in cells managed by an *Access Point*, or the network use *MAC level forwarding*.

5.3.1 Ad-hoc network

Ad-hoc network is the simplest form of Wireless LAN is a network composed of a few nodes without any bridging or forwarding capability. All nodes are equal and may join or leave at any time, and have equal right to the medium. In fact, it's very much like an Ethernet, where you may add or remove node at discretion. This is the kind of radio networks deployed in homes of small offices.

Of course, for this to work all nodes must be able to see all the other nodes of the network, to be able to establish communication with them. When a nodes goes out of range, he just loose connection with the rest of the ad-hoc network. Effectively, this is a single cell network.

One of the node of the ad-hoc network may provide routing or proxying to communicate to the rest of the work, but nodes are still confined to the area within that cell.

5.3.2 Access Points and Roaming

Wireless networks are sometime isolated networks (called ad-hoc), but most of the time they need to be connected to the rest of the world (and the Internet :-). This is usually done through **Access Points**.

In fact, an Access Point is simply a **bridge**, connected on one side to the radio network and on the other side to *Ethernet* (usually), forwarding packets between the two networks. A bridge works at the MAC level, just looking through the MAC headers to make its decisions (filtering) and changing MAC headers according to the MAC protocol used. This means that *NetBeui* and *IPX* work across the access point, and that the nodes connected to the radio must use the same *TCP/IP subnet* as the Ethernet segment the access point is connected to.

Because of the interactions with MAC level acknowledgement, most of the time bridging on Wireless LAN is not as simple and transparent as on Ethernet, and a specific scheme is designed in the MAC protocol. When a node sends a packet, the source address must be his to properly receive the MAC level ack coming back (and vice versa). In theory, if the MAC and the driver are carefully implemented it could be possible to support transparently Ethernet bridges (like in a Linux box), but most manufacturers don't bother (especially that they want you to buy an Access Point).

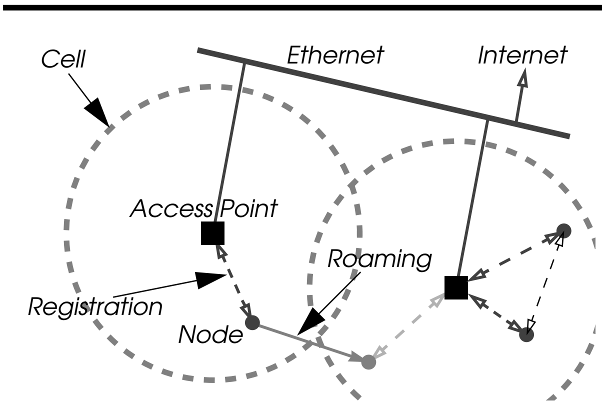
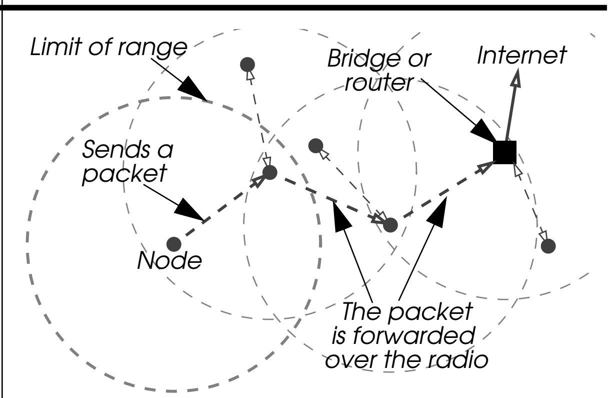
Using Access Points allows to divide the network in **cells**. Each Access Point is at the centre of a cell and is given a different channel (frequency, hopping pattern... - the goal is for each cell to interferer the least with the others). By careful deployment of those Access Point, it is possible to give network access in all parts of large areas.

In fact, most radio access points provide more than this simple bridging functionality. Most of them provide *access control* (to prevent any unwanted radio node to access the network), *roaming* and *out of range forwarding*.

The use of the last two features requires that all the access points that are used to cover the desired area are connected on the same wired segment (IP subnet). Each node needs to register to one of the access point (to avoid confusion between the APs), the nearest one, usually (in fact, more likely the one having the strongest signal, which might not be the nearest). If the node moves, it will automatically switch from one access point to another to retain its access to the wired network (that is **roaming**). If a node wants to communicate with a node which is not in its reach, its access point forwards the packets through the wired network and via the access point where the destination is registered (that is **out of range forwarding**).

A few systems use as well the access point as a network central coordinator of the channel access mechanism (TDMA and polling mode). This is a bad idea, because it decreases the overall reliability and flexibility of the system: every node must be able to communicate at any time the access point in order to work, even if it wants to communicate with a close neighbour.

Access Points, roaming and radio MAC forwarding:

Roaming & Access Points	*Radio MAC forwarding*
	

5.3.3 Radio MAC forwarding

The forwarding mechanism designed around *Access Points* (see *chapter 5.3.2*) requires a fixed wired infrastructure to link the Access Point. This might be satisfactory for most usages, but is not adequate for ad-hoc networks.

Some MAC protocol (such as HiperLan - see *chapter 6.3*) provide a **MAC level forwarding**, where every node of the network can be used to relay the message on the air to the destination. The protocol doesn't rely any more on a fixed infrastructure, but on all the wireless nodes on the path.

So, how do we found the optimal path through the nodes to the correct destination? This forwarding mechanism use management message to propagate network changes and topology information, and from those messages nodes can compute the optimal forwarding tables. Nodes must implement the forwarding capability and propagate message based on those routing tables. In fact, each node of the network acts as a ad-hoc wireless bridge.

Broadcast and multicast messages are a bit of a problem (they have always been on bridging technologies): all nodes just repeat them and the strategy is to flood the network with them (that's the only way to make sure they reach all possible destinations).

Some *access points* also offer the possibility to be configured as **Wireless Repeaters**, which provide the same kind of radio forwarding but in a managed way.

Radio MAC forwarding is elegant and interesting, but all the forwarding consume some more radio bandwidth, which is already limited to start with.

5.4 Some throughput considerations

If the physical layer people are mostly talking range and dB, MAC layer people are (or should be) concerned about the throughput of the system.

5.4.1 Bit-rate versus maximum user throughput

Like for wired products, most radio LAN vendors indicate only the **bit-rate** of their products (also called signalling rate). For example, *Ethernet* is 10 Mb/s, 100 Mb/s or 1 Gb/s, and most radio LAN products between 0.5 and 3 Mb/s (higher rate like 10 Mb/s are slowly coming to the market). The signalling rate is the speed at which bits are transmitted over the medium, but, because of the many overheads of the protocols used to communicate, the user throughput is usually less (note also that they use decimal multiplicators, so for them 1 Mb/s is 10^6 b/s !). The Wireless LANs protocols have usually a **higher overhead** than their wired counterpart (such as Ethernet). This is due to different factors:

The first is the *radio technology*: radio receivers require large synchronisation fields (receiver training, antenna selection...); the radio itself is slow to react (switch from receive to transmit), so needs large slots in the contention window and between packets.

The second is the addition of the *features* necessary for the radio protocol which makes the packet MAC headers larger (fields for network id, encryption parameters ...) or introduces new management packets (synchronisation, authentication, access point registration).

The third is that some *trade-offs* are made to improve the reliability. For example, we might split big packets into small independent fragments to decrease the error probability (see *chapter 5.2.2* on fragmentation). Acks and RTS/CTS add also some overhead. Having a slotted contention decreases the collisions but makes the average contention delay larger as well.

When you add all this, it starts to make a significant difference. If in the case of Ethernet you may hope to reach 80-95 % of the signalling rate, for most radio products, despite being slower, the user throughput is usually between 50 and 70 % of the signalling rate (or even less ...).

5.4.2 Multirate system considerations

Most vendors offer multirate systems (see *chapter 4.7.1*), the lower rate allowing a greater coverage and the higher rate allowing greater throughput at lower range, and offer a mechanism for each node to adapt the bit-rate depending on channel conditions. Basically, when packets start to fail, the node reduce the rate.

Of course, people are likely to benchmark nodes in relatively close proximity (two nodes on the table), when the system will use the highest rate, but the real advantage of Wireless LANs is usually given at higher range (in the garden, moving around), and in this case the system is likely to select the lower rate (and maybe suffer from packet losses and retransmission due to range), so the performance will be less.

However, those rate adaptation schemes are not always the most clever. When there is an interferer in the band, reducing the rate may increase marginally chances of packets to get through, but most of the time having longer transmission time just increase the probability of collision. In cases where there is lot's of contention (lot's of nodes with lot's of traffic), some products do reduce the rate which doesn't help to reduce to congestion (I've seen that personally). In those particular cases, you may want to fix the rate yourself to the highest and disable the rate reduction feature.

Having a multi-rate system also impact the overhead of the system, especially at high rates. All the basic part of the protocol (headers, management messages, contention) is designed for the slowest rate, so when going to higher rate their relative size increase (their duration remain the same while the payload duration decreases).

For example, when sending the same 1500 B packet at 4FSK instead of 2FSK with 802.11, the overhead of the contention window double, the overhead of the MAC level acknowledgement and RTS/CTS double and the overhead of the header increases by 28 %. I've heard that the overhead for 802.11 HR at 11 Mb/s was significantly noticeable compared to 1 and 2 Mb/s speeds, and Lucent claims that increasing the bit rate from 2 to 10 Mb/s (Lucent turbo PPM DS modulation), the effective throughput (user level) is increased only by a factor 3.

5.4.3 Shared throughput versus individual throughput

In the previous chapter, we have examined the overhead added by the protocol and talked about the maximum user throughput usable by the Wireless LAN. But, sometimes, even in a clear channel, the maximum **node to node throughput** may be even less than that. This is usually caused by implementation problems.

The most obvious is for example a slow interface between the PC and the Wireless device. A serial or parallel interface is slower than an ISA or Pcmcia bus and may be a bottleneck.

The second example is devices implementing only one transmit buffer. This saves some cost (memory, complexity), but, as the buffer may be either written by the driver or transmitted over the air but not both at the same time, this creates dead time over the air while the driver refills the buffer and reduces the available throughput. This was one of the performance gain between the first and the second generation of Ethernet cards in the old days.

The protocol might also performs better when many node are active than when only one of them transmits. For example, the contention window in CSMA/CA (number of contention slots) impact the performance; a larger contention window will decreases the collisions but when there is a few nodes, those will wait on average longer to access the channel (the common 802.11 parameters gives better performance for 2 active nodes than for 1). A polling protocol which uses a round robin scheduling mechanism (asks each node in turn if it has a packet to transmit) performs better is every node has something to send than only one node (in this case, between each packet of this node the protocol has to pool all the other nodes of the network for nothing).

Lastly, in the case of MACs being connection oriented (TDMA and some implementation of pooling), an individual node may not be able to use the full link capacity, limiting its performance. For example, if a TDMA system has 10 slots per frame, some physical layer or MAC layer constraints may prevent a node to use more than one slot in each frame, even if the 9 other slots of the frame are free. If the node implementation can only manage one slot, the node individual throughput is only 1/10th of the shared throughput. For the individual throughput to be the maximum throughput, the node must be able to manage multiple slots and multiplex data between these slots.

5.4.4 Contention and congestion

In the previous chapter we examine why the shared throughput could be higher than the individual throughput. But, the reverse can also be true (and is actually more likely for CSMA/CA systems).

When there is many nodes sending packets on the network, the probability of having two nodes choosing the same slot in the contention window increases. When two nodes choose the same slot (and they are first), their packets collide and are lost. This mean that when the level of **contention** increases, the number of retry increases as well, so the performance of the network drop up to the point of congestion.

In fact, 802.11 has a relatively short contention window (16 slots but with a memory effect), and is very sensitive to contention. Unfortunately, it's very easy for any kind of device to generate enough traffic to saturate the wireless link, especially those which assume being on an Ethernet. I have personally seen a nodes composed of 3 nodes and 1 access point (802.11) where the number of retransmissions was higher than the number of packets sent (each packet transmitted on average more than twice).

A solution to this problem is to use RTS/CTS (see *chapter 5.2.3*), because RTS/CTS makes each collision much shorter. In fact, with RTS/CTS enabled, 802.11 can support more than a dozen active nodes without significant reduction in performance due to contention (apart that those nodes have to share the bandwidth). As the RTS/CTS handshake is usually done at the basic rates, its benefit tends to decrease for the highest transmission rates.

5.4.5 TCP and packet losses problem

TCP has been developed for wired LANs, where packet losses are minimal. If a packet is lost, TCP assumes that it is dropped in a router or a bridge because of **congestion**. To try to reduce the congestion, TCP slows down drastically.

On the radio medium, collisions can't be detected and the error rate is higher, so there is more packet losses (if we don't do anything about it). TCP sees that as congestion and reduces its throughput, and so doesn't use all the available bandwidth.

In modern Wireless LAN, **MAC level retransmissions** (see *chapter 5.1.3*) solve totally this problem by detecting and eliminating packet losses due to errors and collisions (and also avoid desequencing packets), so TCP sees a reliable channel and has no reason to slow down (except if MAC level retransmissions are poorly implemented).

5.4.6 Aggregate throughput

It's quite common practice for vendors to advertise for their products something called **aggregate throughput**. This figure indicates the maximum throughput that it is possible to transmit in the full bandwidth by having different adjacent and independent networks on different frequencies or hopping patterns.

Of course, the user of the Wireless LAN will never see such a throughput, and it is a bit like advocating that by having 10 *Ethernet 10baseT* cables you are able to have a 100 Mb/s throughput... But, it gives an indication of how well overlapping cells will share the bandwidth.

For example, with a Frequency Hopping system having 1.6 Mb/s user throughput, by putting 15 networks, each on a different hopping pattern, we should have in theory a 24 Mb/s aggregate throughput. In fact, because the different Frequency Hopping patterns "collide" on the same frequency (and also suffer from co-channel interference) from time to time, the actual aggregate throughput is less, and is in this example only 15 Mb/s.

These collisions of the hopping patterns is why Frequency Hopping can't offer up to 79 networks on the 79 channels (but only up to 15 in this case)...

6 Some Wireless LAN standards

A short gallery of the most famous Wireless LAN standard (but unfortunately not necessarily the most widespread...).

6.1 IEEE 802.11

The main problem of radio networks acceptance in the market place is that there is not one **unique standard** like Ethernet with a guaranteed compatibility between all devices, but many proprietary standards pushed by each independent vendor and incompatible between themselves. Because corporate customers require an established unique standard, most of the vendors have joined the IEEE in a effort to create a standard for radio LANs. This is **IEEE 802.11** (like *Ethernet* is *IEEE 802.3*, *Token Ring* is *IEEE 802.5* and *100vg* is *IEEE 802.12*).

Of course, once in the 802.11 committee, each vendor has pushed its own technologies and specificities in the standard to try to make the standard closer to its product. The result is a standard which took far too much time to complete, which is overcomplicated and bloated with features, and might be obsoleted before products come to market by newer technologies. But it is a standard based on experience, versatile and well designed and including all of the optimisations and clever techniques developed by the different vendors.

The 802.11 standard specifies one MAC protocol and 3 physical layers: Frequency Hopping 1 Mb/s (only), Direct Sequence 1 and 2 Mb/s and diffuse infrared (can we really call it a "standard" when in includes 3 incompatible physical layers ?). Since then, it has been extended to support 2 Mb/s for Frequency Hopping and 5.5 and 11 Mb/s for Direct Sequence (802.11b). The MAC has two main standards of operation, a distributed mode (CSMA/CA), and a coordinated mode (polling mode - not much used in practice). 802.11 of course uses MAC level retransmissions, and also RTS/CTS and fragmentation.

The optional power management features are quite complex. The 802.11 MAC protocol also includes optional authentication and encryption (using the WEP, Wired Equivalent Privacy, which is RC4 40 bits - some vendors do offer 128 bits RC4 as well). On the other hand, 802.11 fails to defines some area (multirate, roaming, inter AP communication...), that might be covered by future developments of the standard or complementary standards. Some 802.11 products also implement proprietary extensions (bit-rate adaptation, additional modulation schemes, stronger encryption...), those extensions may or may not be added to the standard over time.

When 802.11 was finalised (september 97), most vendors were slow to implement 802.11 products because of the complexity of the standard and the number of mandatory features (and in some cases they also need to provide backward compatibility with their own previous line of products). Some of the optional features (encryption and power saving) did only appear months after the initial release of the product. But things seem to be sorted out and we now have fully featured products on the market. The complexity of the specification, the tightness of the requirements and the level of investment required made 802.11 products expensive compared to the previous generation of wireless LANs, but because of the higher standardisation and higher volumes, prices are now dropping.

Even if vendors eventually have launched 802.11 products, the standard doesn't fully guarantee inter-operability: the products have to use at least the same physical layer, the same bit rate and the same mode of operation (and there is so many other little important details...). The most cooperative vendors have been busy lately sorting out interoperability issues with independent testing labs, but it is still a touchy subject...

6.2 802.11-b and 802.11-a (802.11 at 5 GHz)

After 7 years of arguing in sub-committees making 802.11, you would think that most people would had enough of it. In fact no, the 802.11 committee is now busy pushing a new standard at 5 GHz, and also higher speed at 2.4 GHz (by tweaking the Direct Sequence physical layer). Both standard makes changes only to the physical layer, so that the 802.11 MAC can be reused totally unmodified, saving costs.

802.11-a (802.11 at 5 GHz) was standardised first (spring 99), based on *OFDM* (see *chapter 4.7.4*), and using the UNII band (see *chapter 4.2* - so it won't be available in Europe and Japan). The OFDM physical layer is a very close copy of the one used in *HiperLan II* (so they might be some sort of compatibility - see *chapter 6.4*), using 52 subcarriers in a 20 MHz channel, offering 6, 12 and 24 Mb/s and optional 9, 18, 36, 48 and 54 Mb/s bit-rates. No products are yet on the market.

Very soon after, 802.11 did standardise **802.11-b** (802.11 HR), based on a modified DS physical layer (see *chapter 4.7.3*). The goal was to extend the life of the 2.4 GHz band by overcoming the major drawback: low speed. On top of the original 802.11-DS standard, 802.11-b offer additional 5.5 Mb/s and 11 Mb/s bit rates. It was approved by the FCC and they are now products on the market (which are quite popular).

6.3 HiperLan

HiperLan is the total opposite of *802.11*. This standard has been designed by a committee of researcher within the **ETSI**, without strong vendors influence, and is quite different from existing products. The standard is quite simple, uses some advanced features, and has already been ratified a while ago (summer 96 - we are now only waiting for the products).

The first main advantage of Hiperlan is that it works in a dedicated bandwidth (5.1 to 5.3 GHz, allocated only in Europe), and so doesn't have to include spread spectrum. The signalling rate is 23.5 Mb/s, and 5 fixed channels are defined. The protocol uses a variant of CSMA/CA based on packet time to live and priority, and MAC level retransmissions. The protocol includes optional encryption (no algorythm mandated) and power saving.

The nicest feature of Hiperlan (apart from the high speed) is the ad-hoc routing: if your destination is out of reach, intermediate nodes will automatically forward it through the optimal route within the Hiperlan network (the routes are regularly automatically recalculated). Hiperlan is also totally ad-hoc, requiring no configuration and no central controller.

The main deficiency of Hiperlan standard is that it doesn't provide real isochronous services (but comes quite close with time to live and priority), doesn't fully specify the access point mechanisms and hasn't really been proved to work on a large scale in the real world. Overhead tends also to be quite large (really big packet headers).

HiperLan suffers from the same disease as 802.11: the requirements are tight and the protocol complex, making it very expensive.

6.4 HiperLan II

HiperLan II is the total opposite of *HiperLan* (see above ;-). The first HiperLan was designed to build ad-hoc networks, the second HiperLan was designed for managed infrastructure and wireless distribution systems. The only similarities is the HiperLan II is being specified by the ETSI (Broadband Radio Access Network group), operate at 5 GHz (5.4 to 5.7 GHz) and the band is dedicated in europe.

HiperLan II was the first standard to be based on **OFDM** modulation (see *chapter 4.7.4*). Each sub-carrier may be modulated by different modulations (and use different convolutional code, a sort of FEC), which allow to offer multiple bit-rates (6, 9, 12, 18, 27 and 36 Mb/s, with optional 54 Mb/s), with likely performance around 25 Mb/s bit-rate. The channel width is 20 MHz and includes 48 OFDM carriers used to carry data and 4 additional are used as references (pilot carriers - total is 52 carriers, 312.5 kHz spacing).

HiperLan II is a **Wireless ATM** system (see *chapter 5.1.4*), and the MAC protocol is a TDMA scheme centrally coordinated with reservation slots. Each slot has a 54 B payload, and the MAC provide SAR (segmentation and reassembly - fragment large packets into 54 B cells, see *chapter 5.2.2*) and ARQ (Automatic Request - MAC retransmissions, see *chapter 5.2.1*). The scheduler (in the central coordinator) is flexible and adaptive, with a call admission control, and the content of the TDMA frame change on a frame basis to accommodate traffic needs. HiperLan II also defines power saving and security features.

HiperLan II is designed to carry ATM cells, but also IP packets, Firewire packets (IEEE 1394) and digital voice (from cellular phones). The main advantage of HiperLan II is that it can offer better quality of service (low latency) and differentiated quality of service (guarantee of bandwidth), which is what people deploying wireless distribution system want. On the other hand, I'm worried about the protocol overhead, especially for IP traffic.

6.5 OpenAir

OpenAir is the proprietary protocol from **Proxim**. As Proxim is one of the largest Wireless LAN manufacturer (if not the largest, but it depends which numbers you are looking at), they are trying to push OpenAir as an alternative to 802.11 through the **WLIF** (Wireless LAN Interoperability Forum). Proxim is the only one having all the detailed informations on OpenAir, and strangely enough all the OpenAir products are based on Proxim's module.

OpenAir is a pre-802.11 protocol, using Frequency Hopping and 0.8 and 1.6 Mb/s bit rate (2FSK and 4FSK). The radio turnaround (size of contention slots and between packets) is much larger than in 802.11, which allow a cheaper implementation but reduces performance.

The OpenAir MAC protocol is CSMA/CA with MAC retransmissions, and heavily based on RTS/CTS, each contention slot contains a full RTS/CTS exchange, which offer good robustness but some overhead. A nice feature of the protocol is that the access point can send all its traffic contention free at the beginning of each dwell and then switch the channel back to contention access mode.

OpenAir doesn't implement any encryption at the MAC layer, but generates Network ID based on a password (Security ID). This provide some security only because Proxim controls the way all the implementation behave (they don't provide a way to synchronise to any network as 802.11 manufacturers do). OpenAir also provide coarse power saving.

6.6 HomeRF & SWAP

NOTE : this chapter was written when I was finishing writing the SWAP 1.0 specification in December 98. After I left the HomeRF, a lot of big political game did happen, which triggered some critical changes to the specification (SWAP 1.1). I don't really know how much of it is still accurate, but I believe that the standard is no longer as open and vendor neutral as it was and that performance has been dramatically reduced.

The **HomeRF** is a group of big companies from different background formed to push the usage of Wireless LAN in the home and the small office. This group is developing and promoting a new Radio Lan standard : **SWAP**.

The Home is a good market for Wireless LAN because very few houses are nowadays cabled with Ethernet wire between the different rooms, and because mobility in the home is desired (browse the web on the sofa). The use of the 2.4 GHz band allows a free worldwide deployment of the system.

The HomeRF has decided to tackle the main obstacle preventing the deployment of Wireless LAN : the **cost**. Most users just can't afford to spend the money required to buy a couple of Radio LAN cards to connect their PCs (without talking of the access point).

The main cost of a radio LAN is the modem. As this is analog and high power electronics, it doesn't follows *Moore's law* (the market trend that allow you to buy a *Cray* at the price of a calculator after a few years) and modems tend to be fairly stable in price. *Frequency Hopping* modems tend to be less expensive, but the *802.11* specification impose tight constraints on the modem (timing and filtering), making it high cost. The

SWAP specification, by releasing slightly those constraints, allows for a much cheaper implementation, but still keeps a good performance.

The MAC protocol is implemented in software and digital, so doesn't contribute that much to the final cost of the product (except in term of development cost). Releasing some hardware constraints prevented the use of the 802.11, which anyway was much too complex and including too many features not necessary for the task.

The main killer application that the HomeRF group envisages is the integration of digital cordless telephony and the computing word, allowing the PC to reroute the phone calls in the home or to offer voice services to the users.

A new MAC protocol has been designed, much simpler, combining the best feature of DECT (an ETSI digital cordless phone standard) and IEEE 802.11 : a digital cordless phone and ad-hoc data network, integrated together.

The voice service is carried over a classical *TDMA* protocol (with interference protection, as the band is unlicensed) and reuse the standard DECT architecture and voice codec. The data part use a *CSMA/CA* access mechanism similar to 802.11 (with MAC level retransmission, fragmentation...) to offer a service very similar to Ethernet.

The 1 Mb/s Frequency Hopping physical layer (with optional 2 Mb/s using 4FSK) allows 6 voice connections and enough data throughput for most users in the Home. The voice quality should be equivalent to DECT in Europe and much better than any current digital phone in the US. Data performance should be slightly lower than 802.11. The MAC protocol has also been designed in a very flexible way, allowing to develop very cheap handset or data terminals and high performance multimedia cards for PCs...

The **SWAP** specification is an open standard (in fact, more open than 802.11, because there should be no royalty or patent issues), quite simple and straightforward. In fact, the combination of voice and data gets already most marketing people drooling ! The only drawback is that you will have to wait a bit before seeing SWAP products in your favourite supermarket...

6.7 BlueTooth

BlueTooth should not even be mentioned in this document, but people keep thinking that BlueTooth is a Wireless LAN. BlueTooth is a **cable replacement** technology mostly developed and promoted by **Ericsson** with the help of **Intel**, offering point to point links and no native support for IP (need to use PPP). It may be good for some applications, but not for Wireless LANs.

I personally read the BlueTooth specification, and I was not impressed, expect by the size of the thing (more than 1500 pages !). My take is that BlueTooth offers the functionality of a **Wireless USB**, and in fact looking into the huge specification we can see some similarities in the design.

BlueTooth offers the possibility to create a set of point to point wireless serial pipes (*RfComm*) between a master and up to 6 slaves, with a protocol (*SDP*) to bind those pipes to a specific application or driver. The BlueTooth mindset is very vertical, with various profiles defining every details from bit level to application level. TCP/IP is only one profile, implemented through PPP in a specific pipe. There are other pipes for audio, Obex... With BlueTooth, nodes need to be explicitely connected, but they remember bindings from one time to another.

This is miles away from the current wireless LAN approach (connectionless broadcast interface, native IP support, cellular deployement, horizontal play), so

BlueTooth doesn't fit TCP/IP and wireless LAN applications too well. On the other hand, as a wireless USB, it fulfil a role that regular wireless LANs can't, because TCP/IP discovery and binding protocols are more heavyweight.

Currently, BlueTooth is moving very slow (my first reading of the spec was autumn 97 - then called MC-Link) due to its complexity and the inherent limits due to the protocol design (people are learning how to workaround "features"), but eventually some products should reach the market and later on software support should come ...

In summary, if all you want is to run TCP/IP, you may find it cheaper and more effective to NOT wait for BlueTooth and live without the hype.