University of Portsmouth
BSc (Hons) Computer Science
Third Year

Advanced Networks (ADNET)

M21276

September 2025 - January 2026

20 Credits

Thomas Boxall
`thomas.boxall1@myport.ac.uk`

Contents

Page 1

Lecture - Signal Encoding Techniques

2025-09-29 11:00 Asim

There is a deck of slides on Moodle introducing this module's structure & assessments, etc.

1.1 Introduction to Concepts

1.1.1 Computer Networks

Definitions

Computer Network A system that connects two or more computing devices for transmitting and sharing information (data)

There are a number of different activities which can be done on a network:

- Watching Videos
- Playing Games
- Sending and Receiving Messages (including not just text)
- Paying Bills

The core function of the network is to exchange data between interconnected devices.

1.1.2 Data & Signals

Data is the information which is transmitted between devices on the network. The type of the data depends on the context and may include:

- Video
- Audio
- Text
- Images

For data to be able to travel on the network - it has to be converted into digital or analog format. Once in either of these formats, the resultant data is known as *signals*.

Definitions

Signal Electromagnetic Waves that carry data

Analog Signal are signals which vary smoothly over time and have no fixed point to change at and don't have fixed levels

Discreet Signal are signals which maintain constant level for some time then then change to another constant level

Digital Signal are signals which have only two levels - one high to indicate on, or 1, and one low to indicate off, or 0

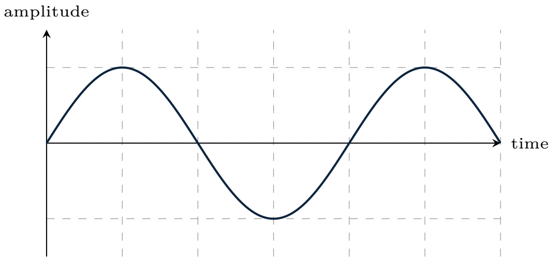

Figure 1.1: Analog Signal

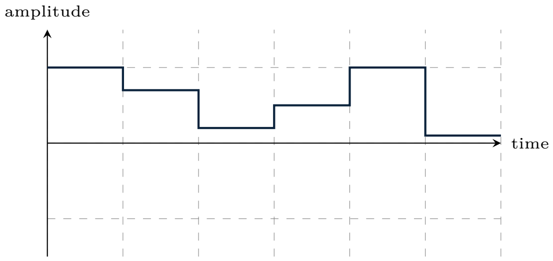

Figure 1.2: Discrete Signal

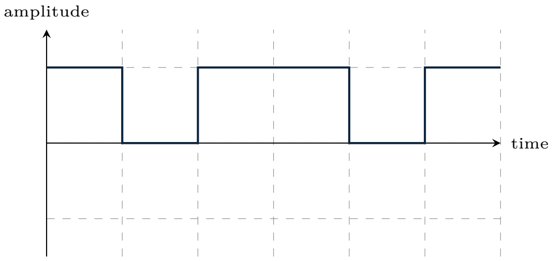

Figure 1.3: Digital Signal

1.2 Data Communications

Within a Data Communications Network - the data will convert between Digital and Analog data at various points. A *modem* may be used to complete this conversion.

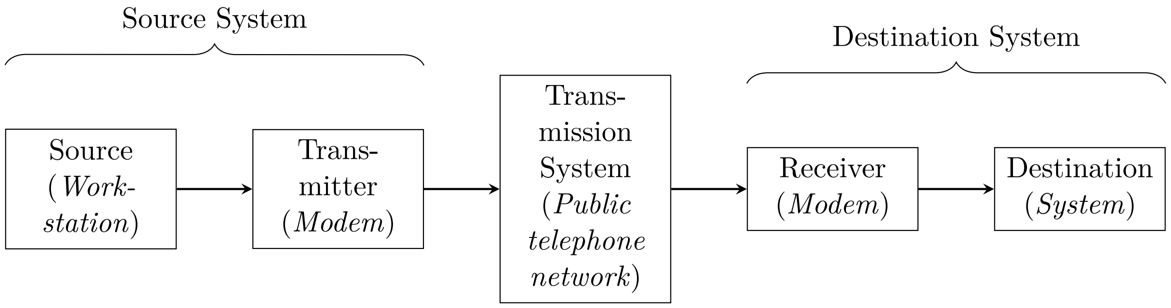

Figure 1.4: Block Diagram of Example Data Communications Signal Chain

In the above diagram, the *workstation* provides the *modem* with a digital signal. The modem then converts this digital signal into an analog signal which can be transmitted across the *Public Telephone Network* that uses analog signals. The receiving *modem* converts the signal to a digital format which the *server* receives.

In saying this, however, there are some devices which still work entirely on analog or digital signals.

For example, the analog telephone network including the switching and transmission is entirely analog. There is also the inverse whereby there are entirely digital signals are processed.

1.3 Digital-To-Digital Signal Encoding

Digital signals are transmitted such that an individual value is transmitted for a defined period of time called the *bit duration*. The *bit duration* is known to both the sender and receiver, allowing the receiver to correctly interpret the transmitted signal which the sender will always transmit at the beginning of the bit duration. We assume the sender and receiver are synchronised and therefore the clock is not transmitted.

The move between two different defined voltages within the transmission is called the *transition*.

As digital signals are discrete (meaning there are defined, absolute data levels) encoding a digital signal as another digital signal is considerably simpler as we don't have to convert one data level to another.

1.3.1 Nonreturn to Zero Level (NRZ-L)

This method only has two levels of data transmitted. It works through mapping the high data level (1) and low data level (0) to a signal level:

- 0 - represented by a high level signal
- 1 - represented by a low level signal

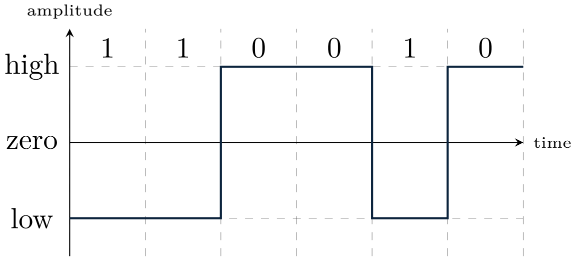

Figure 1.5: Example of NRZ-L Encoding using 110010

1.3.2 Nonreturn to Zero Inverted (NRZ-I)

This method is similar to NRZ-L in that it uses two levels of data transmitted: high (1) and low (0). However it uses the transitions to define the data being transmitted:

- 0 - no transition at the beginning of the bit duration
- 1 - transition at the beginning of the bit duration

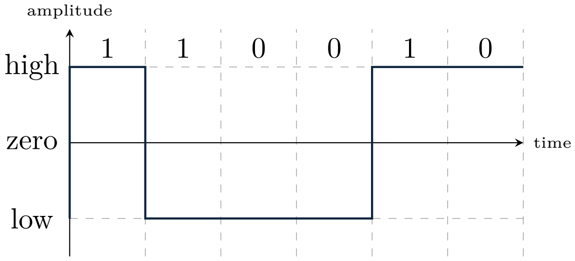

Figure 1.6: Example of NRZ-I Encoding using 110010

1.3.3 Bipolar-AMI

This has three signal levels: high, zero, and low. It works through setting the signal based on the bit to be transitioned to:

- 0 - represented by transmitting zero
- 1 - represented by alternating high and low values, regardless of it there is a 0 value in the middle

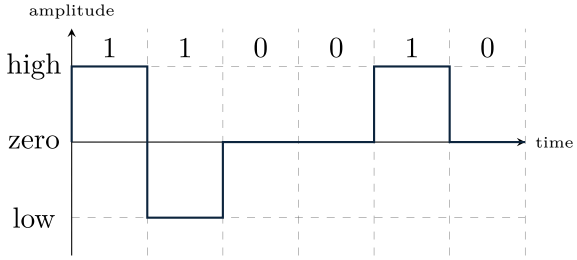

Figure 1.7: Example of Bipolar-AMI Encoding using 110010

1.3.4 Pseudoternary

This has three signal levels: high, zero, low. It works through setting the signal based on the bit to be transmitted:

- 0 - represented by alternating high and low values, regardless of if there is a 1 value in the middle of them
- 1 - represented by transmitting zero value

This is the inverse of Bipolar-AMI.

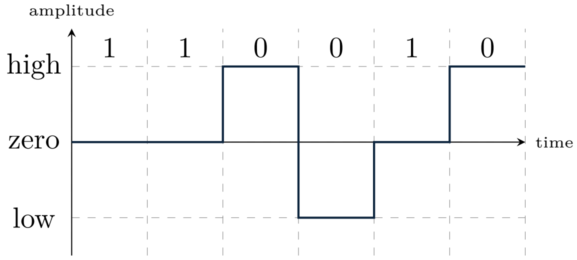

Figure 1.8: Example of Pseudoternary Encoding using 110010

1.3.5 Manchester

This has two signal levels: high and low. It works through having up-to two transitions per interval:

- 0 - represented by a transition from high to low in the middle of the interval (which for successive zeros will require a low-to-high transition at the start of the interval)
- 1 - represented by a transition from low to high in the middle of the interval (which for successive ones will require a high-to-low transition at the start of the interval)

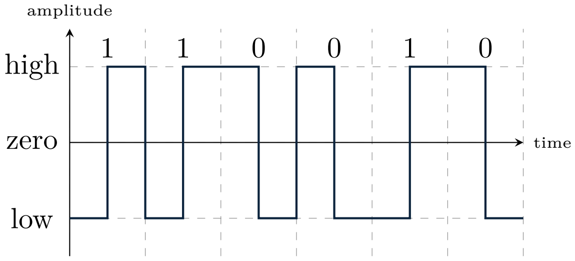

Figure 1.9: Example of Manchester Encoding using 110010

1.3.6 Differential Manchester

This has two signal levels: high and low. It works by always transitioning in the middle of the interval and then examining the transition at the beginning of the interval:

- 0 - represented by a transition at the beginning of the interval
- 1 - represented by no transition at the beginning of the interval

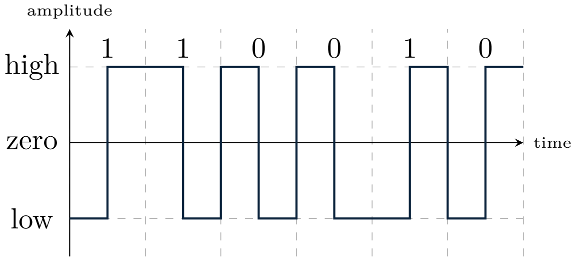

Figure 1.10: Example of Differential Manchester Encoding using 110010

1.3.7 Multi-Level Transmit 3 (MLT-3)

This uses three different signal levels: low, zero and high. It works by cycling through these states based on the bit to be transmitted:

- 0 - remain on the current signal level (regardless of signal level value)
- 1 - move to the next state in the cycle of signal levels (high - zero - low - zero repeat)

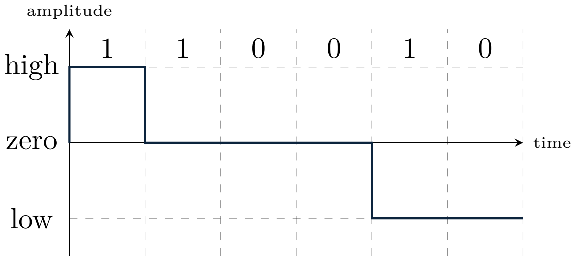

Figure 1.11: Example of MLT-3 Encoding using 110010

1.4 Electromagnetic Waves

Electromagnetic waves are the digital representation of an analog signal. They are considered to be smooth as they don't have fixed values. The key properties of any EM wave are as follows: Amplitude, Phase, Wavelength & Frequency.

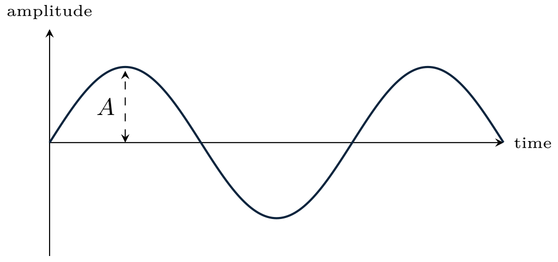

Figure 1.12: Electromagnetic Wave showing Amplitude (A)

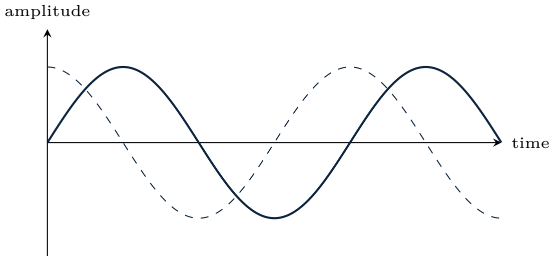

Figure 1.13: Electromagnetic Wave showing Phase (dashed)

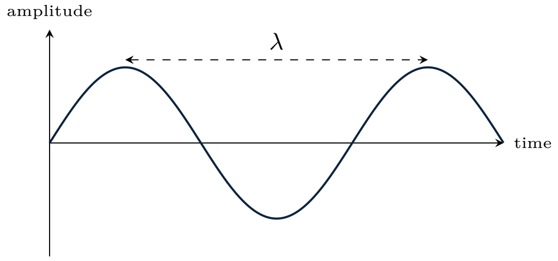

Figure 1.14: Electromagnetic Wave showing Wavelength (λ)

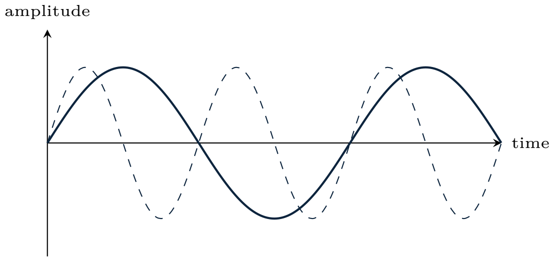

Figure 1.15: Electromagnetic Wave showing increased frequency (dashed)

1.4.1 Carrier Waves & Modulation

Definitions

Carrier Wave a continuous, periodic waveform that carries no information

A *carrier wave* is modified by an information-bearing signal to convey information. The modification can be by either changing its amplitude, frequency, phase, or some combination of the three. The process of modifying a carrier wave is called *modulation*.

1.5 Digital Data, Analog Signals

There are many examples of where digital data has to be transmitted through an analog medium. The most well-known of which being the public telephone network. This is designed to transmit voices, within the frequency of 300 to 3400 Hz; therefore a problem is presented when a digital device is connected to the network. This problem is overcome by connecting the digital device to a Modem (Modulator-Demodulator) which converts digital data to analog signals and the other way around.

There are a number of different methods which can be used to convert digital data onto a analog signal. These methods employ a carrier wave for the digital data to be modulated onto. For the subsequent examples - it's assumed the carrier wave is a sine wave.

1.5.1 Amplitude-Shift Keying

Amplitude-Shift Keying (ASK) modulation works by modulating the digital signal onto the amplitude of the carrier wave. Meaning that the amplitude of the carrier wave is increased for a digital 0 and decreased for a digital 1.

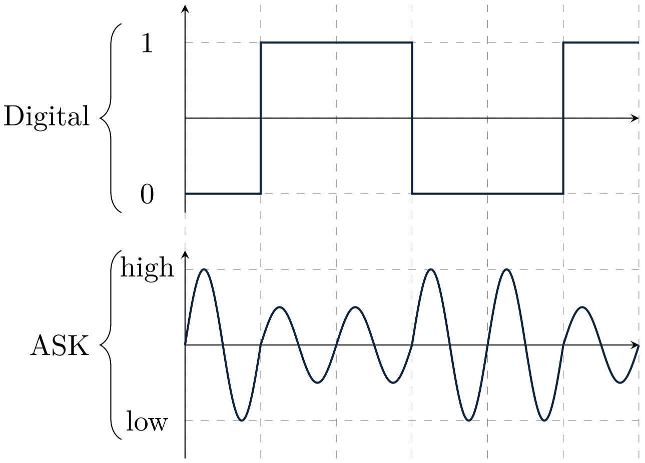

Figure 1.16: Example of ASK Modulation

1.5.2 Frequency-Shift Keying

Frequency-Shift Keying (FSK) modulation works by modulating the digital signal onto the frequency of the carrier wave. This means that the frequency of the carrier wave is increased for a digital 1 and decreased for a digital 0.

The sender and receiver may use different frequencies to allow full-duplex transmissions on the same channel. It is less susceptible to errors than ASK modulation is. FSK can be used for higher frequencies (3 - 30 MHz), radio and Local Area Network transmissions. It can also support Multiple Levels in MFSK.

The *bandwidth* of the transmission is the difference between the frequency of the high frequency (representing a 1) and the low frequency (representing a 0). Different frequency carrier waves will be used for sending and receiving on the same line.

The most common form of FSK is Binary FSK (BFSK) in which the two binary values are represented by two different frequencies near the carrier frequency:

- 0 - represented by $A\cos(2\pi f_1 t)$
- 1 - represented by $A\cos(2\pi f_2 t)$

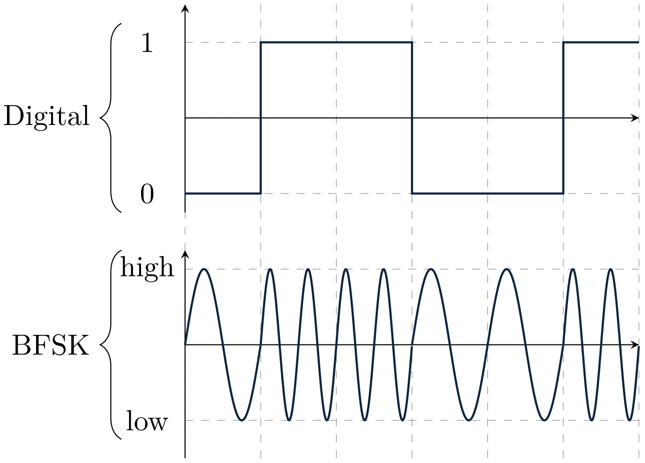

Figure 1.17: Example of BFSK Modulation

1.5.3 Phase Shift Keying

Phase Shift Keying (PSK) Modulation works by modulating the digital signal onto the phase of the carrier wave. This means that the phase of the carrier wave is adjusted to represent digital 0 and digital 1, respectively. There are two different types of PSK studied here.

1.5.3.1 Binary Phase Shift Keying

Binary PSK (BPSK) works by using two phases to represent two different binary digits (0 and 1) and shifting between them:

- 0 - represented by the sine wave $A\cos(2\pi ft + 180)$ which equals the sine wave $-A\cos(2\pi ft)$
- 1 - represented by the sine wave $A\cos(2\pi ft)$

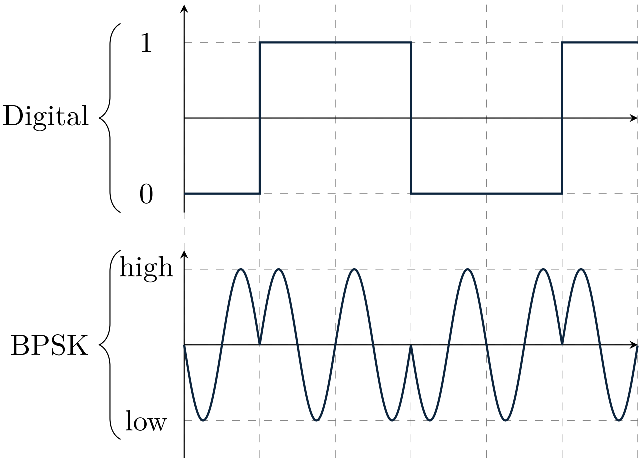

Figure 1.18: Example of BPSK Modulation

1.5.3.2 Differential Phase Shift Keying

Differential PSK (DPSK) works by referencing the previous bit in the current bit transmitted. DPSK removes the need for an accurate local oscillator phase at the receiver which is matched with the transmitter, because so long as the preceding phase is received correctly - the phase reference is accurate.

- 0 - send a signal similar to the previous one
- 1 - send a signal with phase shift as compared to the previous one

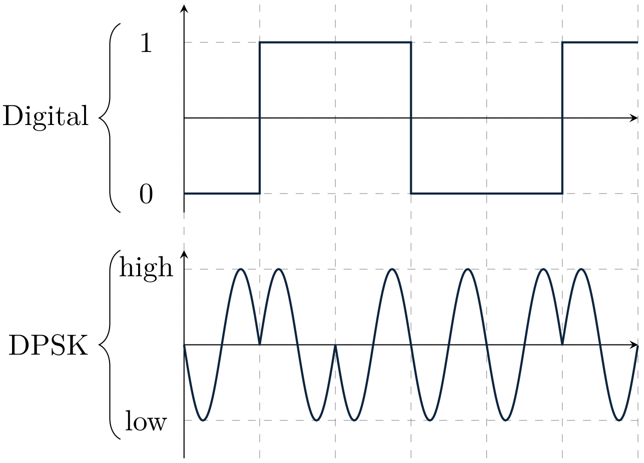

Figure 1.19: Example of DPSK Modulation

Page 2

Lecture - Spread Spectrum and Walsh Codes

📅 2025-10-06 🕚 11:00 👨‍🏫 Asim

2.1 Communication Channels

There are two different types of Communications Channels (also sometimes referred to as *Communication Media* or *Transmission Media*).

Definitions

Guided Media Wired (Bounded) Media (i.e Twisted Pair, Coaxial, Fibre Optic)

Unguided Media Wireless Media (i.e. Microwave, Radio Wave, Cellular, Infrared, Satellite)

Within Guided Media, the electromagnetic waves are guided along a physical path. The medium can be considered to be *point-to-point* if it provides a direct link between two devices and those two devices are the only devices sharing the medium.

Unguided media is the opposite - where the electromagnetic waves are not guided through any physical containment, rather they transmit through air, vacuum or seawater.

The term *Direct Link* is used to refer to a transmission path where the signal is transmitted directly from sender to receiver without any intermediate devices. This can apply to both guided and unguided media. A *multipoint guided configuration* is a configuration such that more than two devices share the same medium.

A transmission may be simplex, half duplex or full duplex.

Definitions

Simplex a transmission in which signals are only transmitted in one direction; one station is the transmitter and the other is the receiver.

Half-Duplex a transmission in which both stations can transmit and receive but only one can transmit at one time therefore

Full-Duplex a transmission in which both stations can transmit and receive at the same time; which requires a medium is required for signals to be transmitted in both directions at the same time

In a full-duplex transmission system there may sometimes be an overlap in the frequency ranges used for each direction of transmission. This is sometimes acceptable and sometimes not - depending on the application. In any case, the overlap would be at the very edges of the frequency range; however this would still cause some interference.

There is a diagram detailing the frequencies used for different unguided transmissions available both in the slides on Moodle & in the textbook on page 109.

2.2 Interference and Noise

Often with transmissions - our signal may be interrupted in some way. This will alter the signal being transmitted which could change the data it represents - therefore garbling the resultant wave. Often this interference will come from *noise*.

Definitions

Interference The combination of two or more electromagnetic waveforms to form a resultant wave in which the displacement is either reinforced or cancelled

Noise An unwanted signal which is combined with desired signal

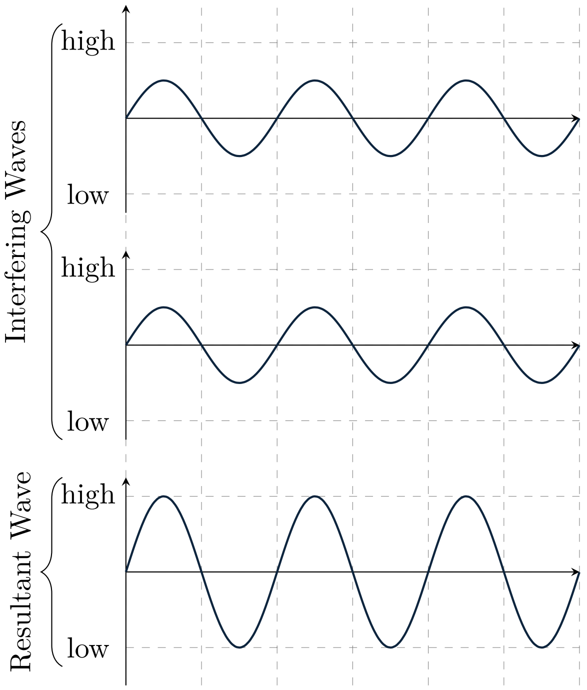

Figure 2.1: Constructive Interference

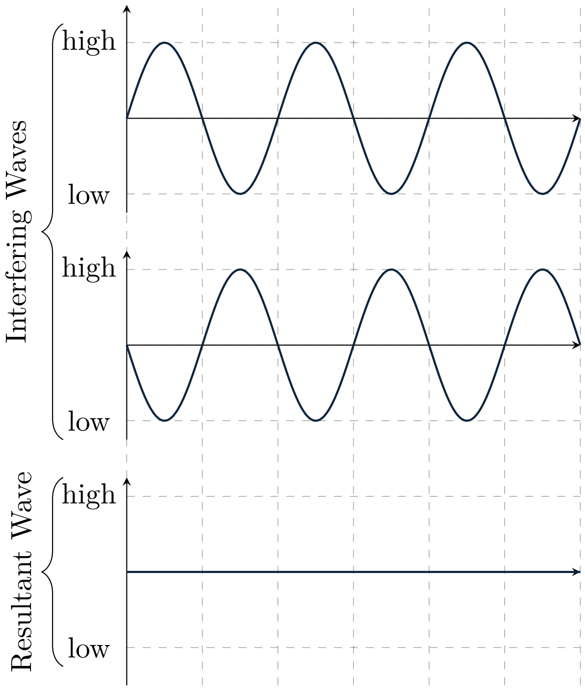

Figure 2.2: Destructive Interference

As we can see in the above figure, where we have two waves which are of the same phase interfering with each other - they will *interfere constructively* to increase the amplitude of the resultant wave. However where two waves in opposite phase interfere with each other - they will *interfere destructively* to effectively cancel each other our and the resultant wave has no amplitude.

For any data transmission, the received signal will consist of the transmitted signal, modified by the various distortions used by the transmission system, plus additional unwanted signals that are inserted somewhere between transmission and reception (which is referred to as *noise*).

Noise can be divided into four categories:

- Thermal Noise
- Intermodulation Noise

- Crosstalk
- Impulse Noise

2.3 Basic Definitions

Definitions

Data Rate The rate, in bits per second (bps), at which data can be communicated

Error The reception of a 1 where 0 was transmitted, or the reception of a 0 when a 1 was transmitted

Error Rate The rate at which errors occur

Frequency Bandwdith The difference between the upper and lower frequencies in a continuous band of frequencies

Channel Capacity The maximum rate at which information can be transmitted through a communication channel

Signal to Noise Ration (SNR) The ratio of the signal power to the noise power, measured in Decibels $10 \log_{10} \frac{\text{signal power}}{\text{noise power}}$

The *Channel Capacity* can be calculated using:

$$C = 2B \log_2 M$$

Where C is the channel capacity; B is the bandwidth; M is the signal or voltage levels.

The Nyquist Bandwidth Theory stipulates that if the rate of signal transmission is $2B$ then a signal with frequencies no greater than B is sufficient to carry the signal rate. The converse is also true. This limitation is due to the effect of intersymbol interference, which is produced by delay distortion. This is in essence based on the Nyquist Sampling Theorem (flashback to A-Level Electronics).

From this we can see that permitting all other things being equal, when we double the bandwidth - we double the error rate. The error rate then only gets worse as we increase the data rate because a higher data rate will mean the bits are shorter so more bits are affected by a given pattern of noise. Mathematician *Claude Shannon* tied these into a formula:

$$C = B \log_2(1 + SNR)$$

Where C is the capacity of the channel in bps and B is the bandwidth of the channel in Hertz.

2.4 Multiplexing

Definitions

Multiplexing A technique that allows the simultaneous transmission of multiple signals through the same channel or link; several signals are combined into a single composite signal

2.4.1 Frequency Division Multiplexing

In Frequency Division Multiplexing (FDM), the different message signals are modulated onto different carrier frequencies. This then means the signals being transmitted are separate from each other in the frequency domain. These modulated signals are then combined together to form the composite signal and this signal is sent over the shared medium or channel. To avoid the interference between the different message signals, a guard band is also kept between the message signals.

On the transmitter end, the signals to be transmitted are modulated onto the different carrier frequencies. Then on the receiver end - the frequencies first pass through a Band Pass Filter (definition not required for this module), then through the demodulator, then finally through a low pass filter (again, definition not needed in this module). The band pass filter is used to separate the specific signal from the composite signal, and the low pass filter is used to filter out the higher frequencies introduced as part of the FDM process.

2.4.2 Time Division Multiplexing

In Time Division Multiplexing (TDM), the channel is divided into several time slots, and each signal allocated during it's time slot. As a result - several signals share the channel without interfering with each other.

2.5 Spread Spectrum

When transmitting our analog signals (whether these originated as digital or analog), we can spread the signal over a wider bandwidth to avoid jamming and frequency interception.

Spread Spectrum is a technique used by military and intelligence applications which is also used in Wireless and Cordless networks. There are a number of different techniques which can be used, we will explore 3 of them.

Definitions

Pseudorandom Noise (PN) is a deterministic sequence of bits which satisfies one or more of the standard tests for statistical randomness while being repeatable after a period

2.5.1 Frequency Hopping Spread Spectrum

In Frequency Hopping Spread Spectrum (FHSS), the signal is broadcast over a number of different radio frequencies, and the frequency used is changed at a fixed intervals which are generally extremely short (i.e. $1ms$). The receiver will hop between the different frequencies used in sync with the transmitter.

If the transmission is compromised, then the attacker would only hear unintelligible blips of the transmission. It would also thwart attempts to jam the signal as the attacker would only be able to block a few bits of the signal.

FHSS transmission systems tend to work with the binary data being fed into a modulator using a digital-to-analog encoding scheme (for example FSK or BPSK). The resultant signal is centred in a frequency. A PN source serves as an index into a table of frequencies; this is the spreading code. Each k bits of the PN source specifies one of the carrier frequencies. At each pre-agreed interval, a new carrier frequency is selected. This frequency is then modulated by the signal produced from the initial modulator to produce a new signal with the same shape, but now centred on the selected carrier frequency.

On reception - the spread spectrum signal is demodulated using the same sequence of frequencies derived from the PN source, and then demodulated to produce the output data.

2.5.2 Direct Sequence Spread Spectrum

Direct Sequence Spread Spectrum (DSSS) works by encoding a single bit to be transmitted (i.e. 1) as a multi-but sequence (i.e. 0110) using a spreading code. The *spreading code* spreads the signal across a wider frequency band in direct proportion to the number of bits used. Which means that a 4-bit spreading code spreads 1-bit of signal across a frequency band which is 4 times greater than a 1-bit spreading code. Of course, it doesn't have to be a 4-bit spreading code; it could be 10-bit or 20-bit.

A common method for encoding DSS is to combine the digital data input signal with a *Pseudorandom Noise (PN)* sequence (the individual bits within are called *chips*). The combined output signal is then referred to as a *chip sequence*.

In reality, this encoding process works by combining the digital data input signal with the Chip Sequence using an Exclusive Or (XOR) operation.

$$\begin{aligned}0 \oplus 0 &= 0\\0 \oplus 1 &= 1\\1 \oplus 0 &= 1\\1 \oplus 1 &= 0\end{aligned}$$

This produces a *combination bit stream* which has the data rate of the spreading code sequence, so therefore has a higher bandwidth than the information stream.

Example: DSSS

If we take the first bit we want to transmit, 0, and the first four bits of PN sequence 0110; we then perform the XOR operation for each of them:

$$\begin{aligned}0 \oplus 0 &= 0\\0 \oplus 1 &= 1\\0 \oplus 1 &= 1\\0 \oplus 0 &= 0\end{aligned}$$

This gives us the transmitted spread-sequence representing our first bit: 0110. This process is then continued for all bits to be transmitted.

On the receiving end, the receiver has the same PN sequence so they can perform an XOR function on the received signal and PN sequence:

$$\begin{aligned}0 \oplus 0 &= 0\\1 \oplus 1 &= 0\\1 \oplus 1 &= 0\\0 \oplus 0 &= 0\end{aligned}$$

The un-spread single bit output is then the output of the XOR. In this case, that would be 0. Obviously, this decoding process is repeated for as many bits as needed.

There is an alternative representation of DSSS using a graph to represent the different signals in the Slides available on Moodle.

2.5.3 Code Division Multiple Access Spread Spectrum

Code Division Multiple Access (CDMA) is a multiplexing technique used with spread spectrum.

To understand how CDMA works, we first have to understand how *Walsh Code* works. Walsh Code is a matrix from which codes can be taken by reading the rows of the matrix. The Walsh matrix starts with the general structure:

$$W_1 = (-1)$$

We can then double n to provide larger matrices, for the events in which we need longer sequences of chip codes.

$$W_{2n} = W_2 = \begin{pmatrix} W_1 & W_1 \\ W_1 & \overline{W_1} \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ -1 & +1 \end{pmatrix}$$

Obviously, this can be taken further to see a 4 × 4 matrix which is a common size we will interact with in this module.

$$W_{2n} = W_4 = \begin{pmatrix} W_2 & W_2 \\ W_2 & \overline{W_2} \end{pmatrix} = \begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & +1 & -1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & +1 & +1 & -1 \end{pmatrix}$$

CDMA works by taking the data signal, with a bit-rate R, and selecting a unique code of n chips according to the Walsh Matrix; for example *row 1 for a; row 2 for b; etc.* Then if a the user, k, sends a 1 - the transmitter sends the chip code c_k; alternatively if the user k sends a 0 - the transmitter sends the chip code $\overline{c_k}$ (which may also be represented in this module as c'_k).

The signal to be transmitted is the built by summing the "columns" of chip codes (i.e. all first indexes summed for A, B, C & D; all second indexes summed for A, B, C & D; etc; in a situation where 4 signals are being transmitted together). The receiver then decodes this signal by performing inner-product-multiplication for the user k. This works by them taking the original Chip Code (not the inverted chip where a -1 has been transmitted) and multiplying each element with the corresponding element in the received signal (i.e. first element of chip code multiplied with first received bit, etc). The output from these multiplications are then summed together, and where the result is the number of bits in the chip code, n: a 1 was transmitted; and where the result is $-n$: a 0 was transmitted.

Example: CDMA

If we take the following users to have the following Chip Codes:

- User A Chip Codes: $(-1, -1, -1, -1)$
- User B Chip Codes: $(-1, +1, -1, +1)$
- User C Chip Codes: $(-1, -1, +1, +1)$
- User D Chip Codes: $(-1, +1, +1, -1)$

Case 1

User A sends 1, user B sends 1, user C sends 1, and user D sends 0 (represented as -1).

We sum their chip codes, for A, B, and C - these are as specified above; however for D we need to invert the chip codes because D is transmitting a 0 (-1).

$$\begin{aligned} A + B + C + D' &= (-1, -1, -1, -1) + (-1, +1, -1, +1) + (-1, -1, +1, +1) + (+1, -1, -1, +1) \\ &= (-2, -2, -2, 2) \end{aligned}$$

The receiver will receive this and perform an inner product multiplication using the Chip Code of A.

$$\begin{aligned} f &= (-2,-2,-2,2) \times (-1,-1,-1,-1) \\ &= (-2 \times -1) + (-2 \times -1) + (-2 \times -1) + (2 \times -1) \\ &= 2+2+2-2 \\ &= 4 \end{aligned}$$

Which therefore confirms that A is sending a bit 1.

Case 2

User A sends 0 (-1), user B sends 1, user C sends 1, and user D sends 0 (-1).

We sum their chip codes, this time inverting A and D because they're transmitting 0s.

$$\begin{aligned} A+B+C+D' &= (+1,+1,+1,+1) + (-1,+1,-1,+1) + (-1,-1,+1,+1) + (+1,-1,-1,+1) \\ &= (0,0,0,4) \end{aligned}$$

Then we can find what the receiver decodes using A's chip code.:

$$\begin{aligned} f &= (0,0,0,4) \times (-1,-1,-1,-1,) \\ &= (0 \times -1) + (0 \times -1) + (0 \times -1) + (4 \times -1) \\ &= 0+0+0+-4 \\ &= -4 \end{aligned}$$

As this results to $-n$ (remembering n is the length of the users chip codes) - we know that A transforms to bit 0.

Page 3

Lecture - Security

📅 2025-10-13 🕚 11:00 Asim

3.1 Security Requirements

Within *Network Security* there are three important requirements. The three requirements work together to ensure that network transmissions are of a good security standard.

Definitions

Confidentiality Ensuring that only authorised parties can read the message; therefore restricting unauthorised third parties from accessing it

Integrity Ensuring that the transmitted message is exactly the same as the received message and that only authorised parties can modify, delete, or reply to the message

Availability Ensuring the message is available on demand to authorised parties only

3.2 Attack Types

There are two main types of attack used.

Definitions

Passive Attacks Adversaries analyse the traffic or read message content

Active Attacks Adversaries modify the message content, deny the service to a request, replay to a message and masquerade (pretend to be a different entity)

We can attempt to overcome the threats using cryptography to encrypt the message on the transmitter's side, and then ensure that only the designated recipient can decode the message.

The message we want to encrypt, known as the *plaintext*, is transformed by a function that is parametrised by a *key*. The output of the encryption process, known as the *ciphertext*, is then transmitted. If this ciphertext is intercepted by the adversary, they are unable to directly interpret it as they do not know what the key used in the encryption process was. However, it may be possible for the adversary to crack the cipher depending on the encryption method used.

There is a standard notation for the plaintext, ciphertext and keys. $C = E_K(P)$ is used to represent the encryption, E of the plaintext, P using the key, K which results in the ciphertext, C. The decryption is defined as $P = D_K(C)$ where the plaintext, P, is the result of applying the decryption, D, with the key, K, on the ciphertext, C.

3.3 Encryption Techniques

There are a number of Encryption Techniques used commonly.

3.3.1 Substitution cipher

In a *substitution cipher*, each letter or group of letters is directly replaced by another letter or group of letters which disguises it.

A well-known example of this is the Ceaser Cipher or a substitution cipher.

Example: Substitution Cipher

If we take the following substitution:

Plaintext:	a	b	c	d	e	f	h	g	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext:	Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

Figure 3.1: Substitution Matrix

We can see how we convert from plaintext to ciphertext.

For example, taking the plaintext *attack* - we get the ciphertext *QZZQEA*; or taking the plaintext *london*, we get the Ciphertext *SGFRGF*.

Using a standard alphabet with 26 characters, assuming we only use lowercase letters, then there are 4×10^{26} possible combinations. Which on a computer with one million CPU cores - can take up to 10,000 years to crack.

3.3.2 Transposition Cipher

In a *transposition cipher*, we take a key and use this key to design a grid which we populate with our plaintext, working row-by-row. We then read the ciphertext from the grid reading column-by-column in alphabetical order of the key's letter.

Example: Transposition Cipher

If we take the plaintext *pleasetransferonemilliondollarstomyswissbankaccountsixtwotwo* and the key *MEGABUCK*.

We start by writing out the key at the top of the grid, and assigning a number to each column. In our case, the numbers are assigned in ascending order of the alphabet.

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6

Figure 3.2: Transposition Cipher Matrix with Key and Number shown

We can then take our plaintext and write that into the matrix working down the rows. We pad the end of the string to fill out the last row entirely, in our case we're using the start characters of the alphabet.

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Figure 3.3: Transposition Cipher Matrix with plaintext added

To get the ciphertext out of the matrix, we read from the matrix vertically in order of the column numbers: *afllsksoselawaiatoossctclnmomantesilyntwrnntsowdpaedobuoeriricxb*.

To decode the message, a similar process is followed: setup the matrix by writing the key, and the column numbers; then write in the ciphertext in columns based on the order of the numbers; then read rows to obtain the plaintext.

3.4 Symmetric Encryption

Symmetric Encryption is an encryption method in which the sender and receiver both use the same key to encode and decode the message. This is where it's alternative name, *shared key encryption* comes from.

Symmetric Encryption works through taking the plaintext input which is then encrypted using the key, K. The encryption algorithm will perform various substitutions or transformations on the plaintext, which will be key-dependent. This means that the same plaintext, with two different keys and the same encryption algorithm will produce two different ciphertexts. The ciphertext is then transmitted to the receiver. The receiver then decrypts the message using that pre-shared-key, K. The decryption algorithm is effectively the encryption algorithm run in reverse. This gives them the plaintext.

To denote a shared public key between A and B, we may see the key notated as K_{AB+}.

We can see an example of a Symmetric Encryption in the DES encryption method.

3.4.1 DES

The *Data Encryption Standard*, or DES, is a symmetric-key encryption algorithm which was developed in the 1970s and has been hugely influential in modern-day cryptography.

The plaintext input for DES is always 64-bits in length, and the key is always 56-bits. If the total plaintext needing to be encrypted using DES is longer than 64-bits then it gets chunked into 64-bit chunks so then the DES algorithm can process its 64-bit blocks.

The DES algorithm works by taking the 56-bit key and generating 16 sub-keys from it. These 16 sub-keys are each used for their own round of encryption, therefore the DES algorithm encrypts the plaintext 16 times through a sequence iterating on the output of the previous as the input to the next. The keys are used in reverse order - meaning K_{16} is used on the first iteration, K_{15} on the second, and so on until K_1 is used on the final iteration.

DES works through the 16 rounds of encryption following a modified *Feistel Network* structure. This works by taking the plaintext and splitting it into two substrings of equal length. One half of the key is passed to the F-Function (Feistel Function) which performs Expansion, Key-Mixing, Substitution and Permutation operations on the input and that round's key. The output from the F-Function then gets XOR'd with the other half of the input string. The next round is then prepared by feeding the output from the XOR operation into the next F-Function, and the input to the current round's F-Function to the next rounds XOR. There are 16 rounds of this in total. The final ciphertext is created by concatenating the output of the final XOR with the output of the penultimate XOR operation (which fed into the F-Function, the output from which fed into the input to the final XOR operation).

> There is a diagram explaining the Feistel Network in the slides on Moodle.

3.4.2 3DES

The *Triple Data Encryption Standard*, or 3DES, is a symmetric-key encryption algorithm which apples the DES algorithm three times to each block.

3DES works with three keys, K_1, K_2, and K_3. It will use K_1 on the first round of DES encrypting the data. Then the second round of DES will decrypt the data using K_2. The third and final round of DES will encrypt the data using K_3. The middle decryption doesn't return the data to the plaintext state, as $K_1 \neq K_2$.

Decryption follows the same encryption process but in reverse: decrypting with K_3, encrypting with K_2 before decrypting with K_1.

3.4.3 AES

The *Advanced Encryption Standard*, or AES, is a symmetric-key encryption algorithm which was designed to replace DES.

AES supports three different key lengths: 128-bit, 192-bit and 256-bit. Similarly to how DES works on 64-bits of data at a time, AES works on 128-bit blocks of data and will perform n rounds depending on the different key-length used. 128-bit keys get 10 rounds, 192-bit keys get 12 rounds and 256-bit keys get 14 rounds.

3.5 Asymmetric Encryption

Asymmetric Encryption is an encryption method where the sender and receiver both use different keys to encode and decode the message. This is where it's alternative name, *Public Key Encryption* comes from.

The sender encrypts the plaintext with the receivers public key to produce the ciphertext. The ciphertext is then transmitted over the network to the receiver. The receiver uses their private key to decrypt the message thus producing the plaintext.

The public key of the receiver is made available to anyone who wants to send the receiver a message. The public key of anyone is restricted to them and them alone - leaking of a public key would mean that an adversary could decode the message. Due to the receivers public keys being required to be able to send the receiver a message - senders will have as many public keys as the number of people they send messages to.

A major flaw with this model is that an adversary could intercept the public-key encoded message and stop it from reaching the receiver. Whilst they couldn't read that message, they could spoof that message by sending their own payload encoded with the receivers public key. The receiver would receive this and decode it.

They keys may be notated as follows:

- Senders public key: K_{A+}

- Senders private key: K_{A-}
- Receivers public key: K_{B+}
- Receivers private Key: K_{B-}

We can see an example of the Asymmetric Encryption in the RSA algorithm.

3.5.1 RSA

The *Rivest-Shamir-Adlerman* cryptosystem, or RSA, is a public-key cryptosystem which was developed in 1973 at GHCQ.

RSA works through taking two large prime numbers where their product is used to form the public key (along with other values used). The original prime numbers are kept secret. Then anyone can use the public key to encrypt the message. The message is transformed using mathematical operations which involve the public key. On the receiving end - only the corresponding private key can decrypt the message. The decryption process involves the original prime factors to reverse the encryption.

The RSA process can be formalised below:

1. Chose two large primes, p and q.
2. $n = p \times q$ and $z = (q - 1) \times (q - 1)$
3. Choose a number relatively prime to z and call it d
4. Find e such that $e \times d = 1 \mod z$
5. To encrypt a message, P, compute $C = P^e(\mod n)$
6. To decrypt the ciphertext, C, compute $P = C^d(\mod n)$
7. The public key is (e, n) and the private key is (d, n).

Example: RSA

We can see the above steps with real numbers substituted in below:

1. Let $p = 3$ and $q = 11$
2. Therefore $n = 33$ and $z = 20$
3. Let $d = 7$
4. We can calculate e using $e \times 7 = \mod 20$ which sets $e = 3$
5. We can encrypt our message, for example 19 by performing $C = 19^3(\mod 33) = 28$
6. We can decrypt the ciphertext, in this example 28 by performing $P = 28^7(\mod 33) = 19$
7. This means we know that the public key was $(3, 33)$ and the private key was $(7, 33)$.

There is a more fleshed-out example of the RSA process for the string "SUZANNE" in the slides on Moodle.

3.6 Ensuring Confidentiality

As we have already seen - we need to ensure confidentiality within the transmission to ensure that adversaries cannot intercept and access the message.

3.6.1 Within a Symmetric Encryption Network

As both the sender and the receiver share the same private key (K_{AB}), the adversary cannot decode the message as they do not have the key.

The sender encodes the plain text message into a cipher text $C = D_{K_{AB}}(M)$. Only the receiver, who has the key, is able to decrypt it using $P = D_{K_{AB}}(E_{K_{AB}}(M))$

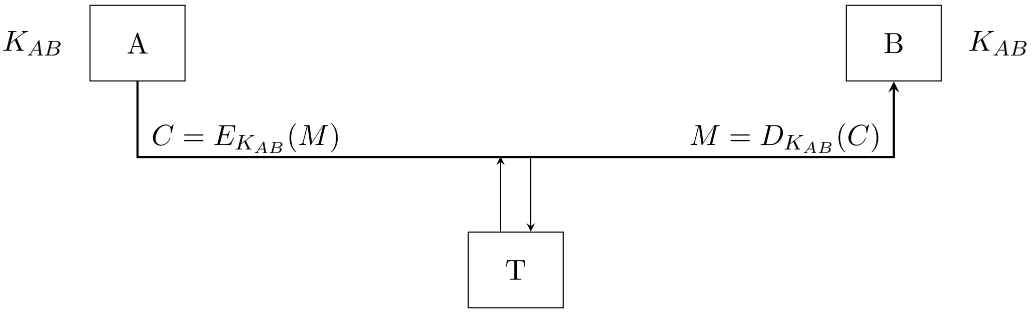

Figure 3.4: Example of Symmetric Key Encryption with a sender, A, receiver, B, and adversary, T showing confidentiality

As we can see in the above example, the adversary, T, can send messages onto the transmission network however these are not encrypted so B knows they have not come from A. T can also consume the ciphertext from the transmission network, however they can't decrypt it as they do not have the key.

3.6.2 Within an Asymmetric Encryption Network

The adversary is not able to read the transmitted message.

Both the sender (A) and receiver (B) compute their own private keys (K_{A-}, K_{B-}) and public keys (K_{A+}, K_{B+}). The public keys are advertised on the network.

When a transmission occurs - the sender will use the receivers public key to encrypt the message $C = E_{K_{B-}}(P)$. Therefore only the receiver can decode the message using their private key $P = D_{K_{B+}}(C)$, alternatively shown as $P = D_{K_{B+}}(E_{K_{B-}}(P))$.

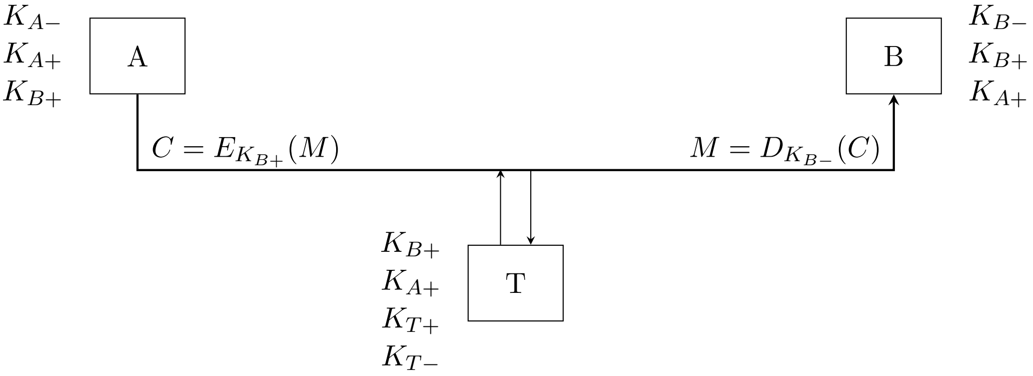

Figure 3.5: Example of Asymmetric Key Encryption with a sender, A, receiver, B, and adversary, T showing confidentiality

As we can see in the above figure, the adversary, T can read whatever is on the transmission network however as it's been encoded using K_{B+} which requires the use of K_{B-} to decode it - they cannot decode it. T can, however, block the message from A (C) from being transmitted and transmit their own message in it's place which is encrypted using K_{B+} which makes it look like A sent it, because T has B's public key in their keyring.

3.7 Ensuring Integrity

We have to ensure integrity of our transmitted messages to ensure that the user who says they transmitted the message are actually the person who transmitted the message. This is for in the event that an adversary intercepts and replaces a message transmitted from Sender to Receiver with a message sent from adversary to Receiver pretending it was from the original sender.

3.7.1 Digital Signatures

The use of a *Digital Signature*, or *Reverse Public Key* is a method of ensuring integrity which removes confidentiality from the encryption process.

Digital Signatures work by the sender encrypting the message using their private key (K_{A-}). The receiver then uses the public key of the sender (K_{A+}) to decrypt the received message. This means that then the sender and only the sender can encrypt messages, however anyone on the network who has the senders public key is also able to decrypt the message. This therefore means that any adversaries on the network cannot change the contents of the message.

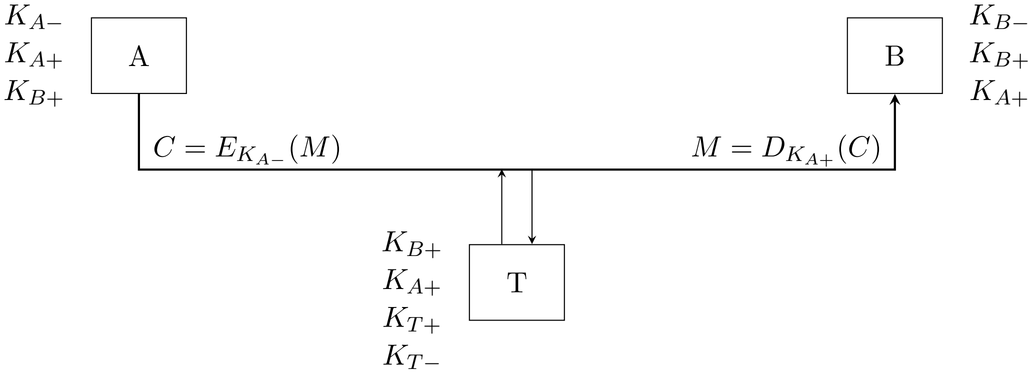

Figure 3.6: Example of Asymmetric Key Encryption with a sender, A, receiver, B, and adversary, T showing Integrity through use of Digital Signatures

3.7.2 Message Digests

The *Message Digests* method works through producing a short fingerprint of the message, which is a hash of the message, represented as $H(M)$. The hash function works by encrypting a small block of the message which is the function of the document. The hash of two different messages will always be different and it is impossible to find M from $H(M)$.

The hash is encrypted with the senders private key, while the message itself is not encrypted. The hash therefore serves as a signature verifying the origin of the document, content and sequencing. This works because the adversary does not have the senders private key, so they cannot spoof a hash value. This method has no confidentiality as the content of the message is transmitted as plaintext, meaning an adversary can read the contents of the message.

On the receiving end - the received message is hashed using the same hashing function, H, and this is compared to the decrypted hash transmitted from the sender. If the two values are the same, then the message is as intended; otherwise it is known that the message has been tampered with.

Popular examples of message digest functions are MD5 and SHA-1.

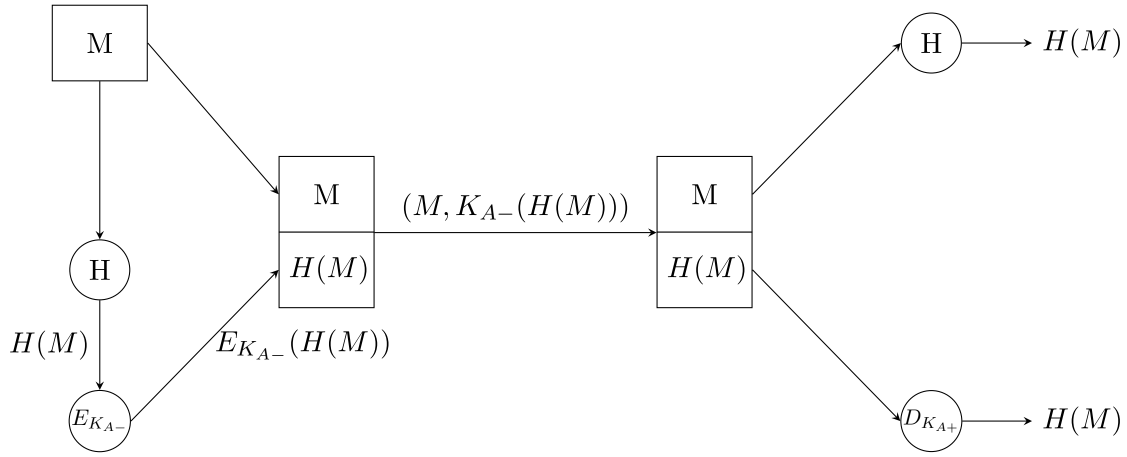

Figure 3.7: Example of Message Digests transmission

3.8 Authentication

Authentication verifies that the messages came from an authentic source. Authentication can be achieved by conventional techniques:

- Private Key for authentication source
- Error detection and sequencing for message alteration
- Timestamp for messages delayed

Authentication is important because it shows that the message has come from a trusted, authentic source. Adversaries may record the messages and then re-play messages at a later time which they have captured. This would look like it's coming form the right person as they are correctly encrypted using the correct keys - however in reality it's a retransmission. This type of attack is called a *Play-Back Attack*

3.8.1 Symmetric Key Authentication

Authentication can be introduced to the symmetric-key encryption model through the following steps:

1. Sender sends their ID (i.e. a password) to receiver
2. Receiver responds with a one-time random number (called a *nonce*). The adversary as well as the sender could pick up the nonce R_B.
3. The sender encrypts using the private shared key, K_{AB} and sends this back to the receiver. The adversary may also pick this up, however they cannot decrypt because they don't have the key. The receiver decrypts the message and now knows they are talking to the sender.
4. The sender sends their own Nonce number, R_A to the receiver. The receiver encrypts this and sends it back to the sender. Only the sender can decrypt this, again because shared keys, and they now know they are talking to the sender.

This prevents playback because of the session established between the sender and the receiver. If the adversary tried to re-send a message later in time - the session would most likely have expired and so the receiver of that message would know to discard because it's not valid any more.

3.8.2 Public Key Authentication

In what can only be described as *the pinnacle of this lecture*, we will now see how Confidentiality, Integrity and Availability are maintained while transmitting secure messages across a communications network.

Public Key Authentication involves three parties: the sender, the receiver and the PKI (*Public Key Infrastructure*). The PKI is a centralised repository for public keys for devices on a given network. The Sender and Receiver's public keys are submitted to the PKI through an authentication-heavy trust-laden process (which we don't need to know about in this module).

The Public Key Authentication process is numbered below (now featuring A and B rather than sender and receiver):

1. A sends a request for B's public key to the PKI
2. PKI returns B's public key to the A (K_{B+})
3. Encrypted with the B's public key, A sends B their identification and a Nonce: $K_{B+}(A, R_A)$
4. B then requests A's public key from the PKI
5. The PKI sends B A's public key (K_{A+})
6. B now sends A their Nonce, A's Nonce, and a shared key to use for this session - all of which gets encrypted with A's public key: $K_{A+}(R_B, R_A, K_S)$
7. A receives this and decodes it using their private key, and then returns just B's nonce encrypted with the shared session key to B: $K_S(R_B)$

A authenticates B when they receive R_A in step 6. This proves it's a fresh message, not playback. B authenticates A when they receive R_B in step 7.

The adversary can fabricate a message at step 3, but as soon as A receives the wrong R_A back in step 6 - the session will be terminated.

The session key will be used as a private key for the duration of the communication; therefore achieving what we know to be the most secure form of communication. The short-lived session keys prevent playback because if a adversary attempts to play back a message, it's session will have expired so the message is discarded.

Page 4

Lecture - Local Area Networks (Ethernet)

🗓 2025-11-03 🕒 11:00 Asim

A *Local Area Network (LAN)* is a network which covers a small geographical area. They are commonly founds in homes and offices, where their function is most apparent: provide personal computers and workstations the ability to easily share data between one another at a high rate of transfer. A LAN also enables users to access other devices such as printers, modems or local servers. LANs are generally privately owned. LANs cover a small geographical area. Within LANs, noise and errors are minimised.

Moving up in scale, we find the *Metropolitan Area Network (MAN)* which is a class of network which serves a large geographical area between 5 and 50 kilometers in range. This could include several buildings such as a University Campus, or even up to a small city. MANs are larger than LANs and generally will provide their communication by fibre optic cable rather then copper. They mostly work at layer 2 (data link) of the OSI model.

At the top of the size scale - we find the *Wide Area Network (WAN)* which is the biggest class of networks. A WAN connects LANs and MANs together. The most common example of a WAN is the *internet*.

4.1 Topologies

Definitions

Topology A representation of the layout of the network

4.1.1 Bus Topology

In the bus topology, all the connected devices attach directly through to the linear transmission medium, known as the bus. There are specialist connectors used at the junction between the branch to device and the bus itself.

A transmission from any station will travel down the length of the medium in both directions and can be received by all other stations. At the end of the bus, there are terminators which absorb any signal, removing it from the bus.

We can see this arrangement in the below figure where the black squares represent the devices connected to the bus and the red squares represent the terminators on the end of the bus.

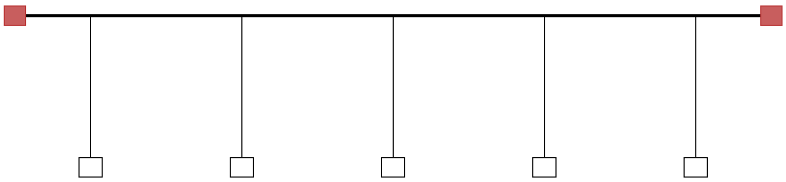

Figure 4.1: Example of a Bus Topology

There are two problems with this arrangement:

- As all transmission will be received by all other stations - there is a need for regulation over demarcating which transmission is relevant to which stations
- There is a need for there to be some regulation over the transmission - to minimise collisions between transmissions where two or more stations are simultaneously transmitting; and to prevent a single station broadcasting for a long period of time preventing other stations from transmitting.

These problems are resolved through breaking the data to be transmitted into *frames*, which contain a header that contains the control information, including a stations unique identifier.

The Carrier Sense Multiple Access Collision Detection (CSMA/CD) protocol can be used to detect and retransmit data lost to a collision.

4.1.2 Star Topology

In the star topology, all the stations connect via a bi-direction connection to a central node. Generally, there are two alternatives for the central node to minimise a single point of failure in the network.

An example of a single-centre-node star topology can be seen in the below diagram. The central switch is represented by the green circle, while the individual workstations are represented by black squares.

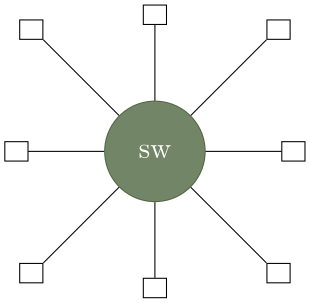

Figure 4.2: Example of a Star Topology

One approach for the operation of the star topology is for the transmission of the frame to be initially from the transmitting station, received by the hub, which then broadcasts to all other connected stations. Only one station will actually want this message to be transmitted to them. In this option - the central node would be a hub.

An alternative operation method for the central node is to act as a *frame switching* device. This is where the frame is buffered in the node and then retransmitted on an outgoing link to the destination station. In this option - the central node would be a switch.

4.2 OSI and IEEE 802.x Protocol Layers

The *Open Systems Interconnection* (OSI) Model is a reference model developed by the ISO that provides a common basis for the coordination of standards development for the purpose of system interconnection. This model underpins all network communications - giving a standard 7-layer stack taking the data to be transmitted from the application to the medium, and vice versa on the receiving end.

The IEEE 802 LAN Committee have developed a sub-architecture to the OSI architecture, focusing on the lower-level LAN transmissions. The IEEE 802 LAN architecture focuses on the lowest two layers of the OSI model: physical and data-link.

4.2.1 Physical Layer

The lowest level of the OSI model maps directly to the lowest level of the IEEE 802 LAN architecture, both calling it the *physical* layer as this is the layer closest to the transmission medium. The physical layer of the IEEE 802 LAN architecture includes the following functions:

- Encoding / decoding of signals
- Preamble generation / removal (used for synchronisation)
- Bit transmission / reception
- Transmission medium specification
- Topology specification

Note that the last two bullet points are not present in the OSI physical layer, as they are recognised in that model as being below the scope of the OSI model. They are included in the IEEE 802 LAN model because they are considered to be of critical importance to the design of the LAN.

4.2.2 Data Link Layer

Working up the OSI model, after the physical layer comes the *data link* layer. This layer provides the following functions:

- Governing access to the LAN transmission medium
- On transmission - assemble the data into a *frame* with address and error-detection fields (more on these later)
- On reception - disassemble the frame and perform address recognition and error detection
- Providing an interface to the higher layers, performing flow and error control

Within the IEEE 802 LAN architecture - the data link layer is subdivided into two different layers: the LLC and MAC. The IEEE has this separation because the OSI model doesn't include the logic required to manage access to the shared-medium at layer 2, as well as that there may be multiple MAC options for the same LLC.

4.2.2.1 LLC Layer

The *Logical Link Control* (LLC) layer is the "higher" of the two layers and it's primary purpose is to interface with the higher layers, performing flow control and error control.

In the bigger picture - data is passed down the protocol stack to the LLC which appends control information as a header. This creates a LLC *protocol data unit* (PDU). The entire LLC PDU is then passed down to the MAC layer.

4.2.2.2 MAC Layer

The *Medium Access Control* (MAC) layer is the "lower" of the two layers. It sits between the LLC and physical layer. The MAC layer handles the following functions:

- On the transmission end - assemble the transmission frame (adding addresses and error detection)
- On the receiving end - disassemble the received frame (removing the addresses, performing error detection using the provided data)
- Perform Medium Access Control (governing access to the transmission medium)

The MAC layer takes the LLC PDU passed down from above and forms a MAC frame with it through appending control information at the front and back of the packet.

4.3 MAC: Ethernet Access Protocol

The *Media Access Control* Protocol is a protocol for sharing the bus or hub.

Only a single station can transmit at the time. If two stations attempt to transmit at the same time, then there will be a collision.

When a collision occurs - a message is transmitted on the bus to state that there is a collision. Then the station will wait a random back off time which will be a value between 0s and the maximum random back off time. The maximum back time can be calculated using the back off algorithm

$$2^n - 1 \times \text{slot time}$$

where n is the number of collisions. After waiting the random back off time, the station then attempts to retransmit. In the event that there is excessive collision, where $n = 15$, the error is reported up the OSI model to the upper layers.

The Slot Time is calculated based on the time it would take for 512 bits to be transmitted. For example on a 10Mbps transmission line, it will take $51.2\mu sec$ to transmit 512 bits.

4.4 Networking Devices

In LANs there are key networking components which make the network function.

4.4.1 Bridge Operations

A bridge will connect two different similar LANs together where they use the same protocols. Due to this, the amount of processing which the bridge does is minimal. Some bridges are capable of mapping from one MAC format to another (for example to interconnect a Ethernet using LAN to a token-ring using LAN).

Bridges increase the reliability of the network, as it subdivides the network into smaller units. This reduces the impact potential if one part of the network fails. It increases performance by having multiple bus topologies so multiple frames can be transmitted at the same time. It can also increase the security of the network, as it may be needed for some segments of the network to be encrypted, but others not. Finally, bridges can interconnect multiple networks in different geographical locations where it would be impractical to run a single Bus topology through.

4.4.2 Routing in LANs

When a bridge receives a frame from the transmitting network - it must decide whether to forward it or not. Furthermore, if the bridge is connected to more than two LANs then it must decide which LAN to forward the frame onto.

Within LANs - fixed routing is where the routing table has manual entries entered for the different destinations on the network. This requires manual updating when there is a change on the network, for example if a station is removed.

Dynamic routing can also be used which is where the routing table is dynamically generated and is automatically updated to reflect changes to the network.

4.4.3 Hubs

Hubs act as a repeater - when a single station transmits, the hub repeats this signal on all other outgoing links to every other station.

If two stations transmit at the same time, there will be a collision. Multiple levels of hub can be configured in a hierarchical configuration, therefore each hub can have a mixture of stations and other hubs attached to it. This fits well with building wiring practices.

4.4.4 Switches

In contrast to Hubs, Switches work by forwarding the frame to the appropriate destination only. They take the frame, review the destination address and forward based on that.

Some switches work at layer 3 and have some routing capabilities, or the ability to work with IP addresses. Other switches work at later 2 and use the MAC address to define destination.

There are two different techniques which can be used by switches for receiving, processing and transmitting the packets. Both introduce small amounts of delay.

Store and Forward works by receiving the packet, then once fully received, it is processed to identify which port the data will be transmitted out of. SF delay is calculated as follows:

$$\text{Store \& Forward Delay} = \frac{L}{R} + \text{Switch Latency (MAC processing)} + \frac{L}{R} + \text{propagation delay of cable} \\ + \text{propagation delay of cable}$$

Within switch delay calculations - there is a common metric which we need to be aware of. This is commonly represented in shorthand as L/R. The full calculation can be seen below:

$$\frac{L}{R} = \frac{\text{length of frame}}{\text{rate of transmission}}$$

Example: Store and Forward Calculations

If we take that the packets have a length $L = 10000$, the transmission rate is 100Mbps, the switch has a propagation time of $3\mu sec$, and a latency of $4.8\mu sec$. We can find the time for transmission.

First we find the L/R value:

$$\frac{L}{R} = \frac{1000}{100} = 100\mu sec$$

We can then substitute this into the above formula.

$$100 + 4.8 + 100 + 3 + 3 = 218.8\mu sec$$

Virtual Cut Through works by reading the first bits of the packet as they are being received, then as soon as it's identified the port to transmit out of - the switch begins transmitting the packet out. This means that virtual cut through is considerably quicker than store and forward as there is no delay while waiting for the entire packet to be received by the switch before it transmits it. This can be seen in the delay algorithm.

$$\text{Virtual Cut Through Delay} = \frac{L}{R} + \text{Switch Latency (MAC processing)} + \text{propagation delay of cable}$$

Example: Virtual Cut Through Calculations

If we take that the packets have a length $L = 10000$, the transmission rate is 100Mbps and a latency of $4.8\mu sec$. We can find the time for transmission.

First we find the L/R value:

$$\frac{L}{R} = \frac{1000}{100} = 100\mu sec$$

We can then substitute this into the above formula.

$$100 + 3 + 4.8 = 107.8\mu sec$$

4.5 Ethernet Frame Format

The Ethernet Frame is the thing which is transmitted on Ethernet-protocol-using networks. It has a number of different fields each with a set size:

Preamble Additional information, for example Manchester Encoding which is used to synchronise the clock of the receiver

FD Frame Delimiter - to signal the start of the frame

Length Data length in bytes.

Pad Additional data inserted to ensure that the frame is the minimum size (46 bytes)

CRC Checksum used to detect errors on the frame

56	8	48	48	16	0-12000	0-368	32
preamble	FD	destination	source	length	data	pad	CRC

Figure 4.3: Ethernet Frame showing field size in bits

Frames have a minimum data length of 46 bytes (368 bits). If the data is shorter than this, for example 100 bits, padding will be added to the frame to make up to the minimum data length. Obviously data can be longer than 368 bits minimum, as long as it's within the 12000 bit maximum.

Frame size can be calculated using the lengths of the fields of the frame:

$$\text{Frame size} = \text{Destination} + \text{Source} + \text{Length} + \text{Data} + \text{Padding} + \text{CRC}$$

As above, the 'frame size' doesn't actually include all the values. It doesn't include the preamble or the FD. These values are added when working out the packet size. Packet size can be calculated as follows:

$$\text{Packet Size} = \text{Frame size} + \text{Preamble} + \text{FD}$$

Example: Frame Length Calculations

For the following payload (data) values, calculate the Padding, Frame size & packet sizes.

Payload	Padding	Frame	Packet
0	368	48 + 48 + 16 + 0 + 368 + 32 = 512	576 = 512 + 56 + 8
100	268	48 + 48 + 16 + 100 + 268 + 32 = 512	576 = 512 + 56 + 8
368	0	48 + 48 + 16 + 368 + 0 + 32 = 512	576 = 512 + 56 + 8
1000	0	48 + 48 + 16 + 1000 + 0 + 32 = 1144	1208 = 1144 + 56 + 8

Table 4.1: Frame & Packet size calculations

4.6 Broadcast Domains

The *broadcast domain* is the set of devices which receive broadcast frames from each other.

A *broadcast* is a special message on a network designated for all devices on that network to receive.

The MAC address in the frame indicates this fact. Broadcasts are generally used for purposes such as network management or transmitting an alert to lots of devices.

Broadcasts are the opposite to a *unicast* message which is destined for a single device on the network - as the intended recipient's MAC address is in the destination field of the frame.

4.7 Virtual LANs

Virtual Local Area Networks (VLANs) are a logical group of devices within a LAN.

VLANs combine workstations and network devices into a single broadcast domain regardless of the physical LAN segment they are attached to. Where traffic needs to travel from one VLAN to another VLAN - routing is required. Routers can be implemented as separate devices so that traffic from one VLAN to another is directed to a router, or the router logic can be implemented as part of the LAN switch.

VLAN membership is not constrained by physical location - as they are entirely logical. This means there is a need to define VLAN membership, for which there are a number of different approaches:

- Membership by port group - each switch in the LAN configuration contains two types of port (trunk to connect switches together, or access port which connects the switch to an end system). This approach is advantageous as it's relatively easy to configure however the network manager must reconfigure the port when it is needed for a different device to connect to it.
- Membership by MAC address - a MAC address is assigned to be a member of a designated VLAN. This is advantageous as the right VLAN will always be used when the same device connects to the network, however in situations where intermediary devices (such as docking stations) are used - these also have to have their MAC address configured into the right VLAN.
- Membership based on protocol information - VLAN membership can be assigned based on IP address, transport protocol information, or even higher-layer protocol information. This is flexible, but does require switches to examine portions of the MAC frame above the MAC layer which can have degrade performance.

4.8 IEEE 802.3 10Mbps

So far in this lecture we've been discussing 10 Mbps speed networks, which are painfully slow and out-dated.

There are a number of alternative physical layer mediums, which can be seen in the below table.

	10BASE5	**10BASE2**	**10BASE-T**	**10BASE-FP**
Transmission Medium	Coaxial Cable (50Ω)	Coaxial cable (50Ω)	Unshielded Twisted pair	850-nm optical fibre pair
Signalling Technique	Baseband (Manchester)	Baseband (Manchester)	Baseband (Manchester)	Manchester / on-off
Topology	Bus	Bus	Star	Star
Maximum Segment Length (m)	500	185	100	500
Nodes per segment	100	30	-	33
Cable diameter (mm)	10	5	0.4-0.6	62.5/125 μm

Table 4.2: Comparison of IEEE 802.3 10Mbps physical layer medium alternatives

Network speed can increase up to 10 Gbps Ethernet, which is seen in the core backbone of networks; or even up to 100 Gbps Ethernet which is seen in the core of datacentres or large server farms.

Page 5

Lecture - Wireless LANs

📅 2025-11-10 🕚 11:00 Asim

5.1 What is a Wireless LAN?

Wireless LANs (WLAN) are an extension of a wired LAN, as we saw last week. Wireless LANs allow the client devices to connect wirelessly to the LAN as the name would suggest.

Within wireless LAN architecture - there is a backbone wired LAN which supports the wired components within the LAN and provides one or more bridges or routers to link with other networks. There is also a *control module* (CM) which acts as an interface to a WLAN, often seen as the “Router” in a small domestic LAN or a Wireless Access Point (AP) in a larger LAN.

The CM includes either bridge or router functionality to link the WLAN to the backbone. They also include some sort of control logic, such as a polling or token-passing scheme to regulate the access from the end-systems.

When discussing WLANs, we refer to *user modules* (UM), which are the end-user devices. They can be stand-alone devices such as Laptops or Smartphones, etc. Alternatively, UMs can also be wireless receiver devices which output a wired LAN signal which allows devices to have a wired connection.

5.1.1 Large WLANs

In larger networks - there can be multiple control modules, each interconnected by a wired LAN. Each control module supports a number of wireless end systems within their transmission systems (often between 100 and 300m for enterprise APs). Where there are multiple control modules on a network - they each have to have their own frequencies assigned, to prevent interference. When moving from one cell, the coverage area of a CM, to another cell and staying connected to the same network - the CMs will hand off the device providing a seamless experience for the end-user.

In multi-cell networks - a single cell is referred to as a *basic service set*. The entire wireless network including the LAN backbone is referred to as the *extended service set*.

5.1.2 Ad-Hoc WLANs

The final type of WLAN we may come across is an *Ad-Hoc Wireless LAN*. In this topology, there is no fixed infrastructure (i.e. CMs, APs, etc). Rather, a collection of stations within a range of each other may dynamically configure themselves into a temporary network of peer-to-peer communications.

A common example of an ad-hoc network is an Independent Basic Service Set (IBSS). This is a BSS where all the stations are mobile stations which have no connection to other BSS’. Within an IBSS - all stations communicate directly, there is no AP involved.

5.2 WLAN Requirements

There are a number of requirements for a WLAN to be effective and efficient.

Throughput The MAC protocol should make efficient use of the wireless medium to maximise capacity

Number of Nodes WLANs may need to support hundreds of nodes across multiple cells

Connection to backbone LAN There should be a good quality connection to the backbone LAN. For infrastructure WLANs - this is accomplished through CMs

Service Area Typically, the coverage area for a WLAN is between 100 and 300m

Battery Power Consumption Often the devices connecting to a WLAN are battery powered - it is important that the battery isn't drained through connecting to a WLAN. Typical WLAN implementations have features to reduce power consumption when not using the network (i.e. a sleep mode)

Transmission Robustness and Security WLANs can be vulnerable to interference and network eavesdropping. The design of WLAN must permit reliable transmission even in a noisy environment and should provide some level of security from eavesdropping.

License Free Operation WLANs should operate within available frequency-bands which do not require the purchase of special licenses

Handoff / Roaming The MAC protocol used in the WLAN should enable mobile stations to move from one cell to another

Dynamic Configuration The MAC addressing and network management aspects of the WLAN should permit dynamic and automated addition, deletion and relocation of end systems without disruption to other users.

5.3 IEEE 802.11

The IEEE 802.11 protocol is the main protocol used for communications within the wireless network. This is the protocol which is commonly known as Wi-Fi. There have been many variations and subsequent discoveries, which are represented by a letter, or multiple letters on the end of the name - such as 802.11ax.

Definitions

Associated Stations Connected devices to a WLAN

Access Point Provides access to the distribution system for associated stations

Basic Service Set (BSS) Set of stations controlled by a single coordinator

Extended Service Set (ESS) A set of one or more connected BSS'

Distribution Systems (DS) A system used to interconnect a set of BSS' and integrated LANs to create an ESS

Frame MAC protocol data uint

5.3.1 Architecture

There is a common prescribed architecture for all WLANs, as stipulated within the IEEE 802.11 specification. This architecture consists one or more basic service sets, which consist of some number of stations executing the same MAC protocol and competing for access to the same shared wireless medium. A BSS may be isolated or it may be connected to a backbone distribution system (DS) through an AP. The AP functions as a bridge and relay point. The DS can be a switch, a wired network, or a wireless network.

In a BSS - clients do not communicate directly with one another. Rather, if one station in the BSS wants to communicate with another station in the same BSS, the MAC frame is first sent from the originating station to the AP and then from the AP to the destination station. Similarly where a station wants to communicate with a station in a different BSS - the MAC frame is first sent from the sending station to the AP, which relays the frame over the DS to the destination station.

It is possible for two BSS to overlap geographically, so that a single station can participate in more than one BSS. The association between a station and a BSS is dynamic - stations may turn off, come within range, and go out of range.

An Extended Service Set consists of two or more BSS' interconnected by a distribution system. Typically the distribution system is a wired backbone LAN but can be any communications network. The ESS appears as a single logical LAN to the Logical Link Control (LLC) Level.

An AP is implemented as part of a station; the AP is the logic within a station that provides access to the DS by providing DS services in addition to acting as a station. To integrate the IEEE 802.11 architecture with a traditional wired LAN, a portal is used. The portal logic is implemented in a device such as a bridge or router that is part of the wired LAN and that is attached to the DS.

5.3.2 Services

The IEEE 802.11 protocol specifies a number of services. The services are divided into two types: those provided by *distribution system* and those provided by the *station* itself.

Distribution system services handle the connection and data transfer across multiple access points and networks:

Association This is a service which establishes a data link-layer connection between a wireless station and an access point (AP)

Disassociation This service terminates an established association with an AP.

Distribution This service handles the distribution of data frames from an AP to other APs or stations.

Intergration This enables the integration of the wireless network stations with other wired networks such as the internet.

Reassociation This allows a station to switch its connection from one AP to another within the same ESS.

Station services provide the authentication, deauthentication and MSDU delivery for individual devices:

Authentication This is the process where a device verifies itself to gain access to the WLAN.

Deauthentication This is the process used to terminate a previously authenticated network connection.

MSDU delivery This is the delivery of the MSDUs which are the units of data exchanged between wireless stations.

Privacy This service implements encryption (like WEP) to secure data transmissions and ensure privacy.

5.4 Access Control

There is a need within WLANs to control who can communicate on the network and when. This is to prevent collisions in the wireless network.

There are two possible modes which can be used to coordinate communications on a Wireless Network. *Distributed Coordination Function* (DCF) provides a distributed approach where there is no centralised controller; while *Point Coordination Function* (PCF) is an alternative paradigm sitting on top of DCF's core functions but also providing centralised control over the communications ensuring contention-free communication. Both of these operate at the MAC layer of the OSI model.

5.4.1 Point Coordination Function

Point Coordination Function (PCF) is a centralised control which provides a contention free service (for example base stations to a backbone). PCF sits on top of the contentious DCF function.

PCF works by the base station polling the other stations asking them if they have frames to transmit, guaranteeing no collisions. The base station sends a beacon frame (between 10 and 100 times a second) which invites stations to sign in. The frame contains information such as hopping frequencies, dwell time, clock synchronisation, etc. When a station is signed in, it is guaranteed a fraction of the bandwidth (therefore making it possible to get quality of service) in a round-robin time-share style. The base station also manages the power; through putting some stations in a standby mode until awakened by a reception.

When polling, the Point Coordinator makes use of *PCF Interframe Space* (PIFS) which is smaller than *DCF Interframe Space* (DIFS). The point coordinator can seize the medium and lock out all asynchronous traffic while it issues polls and receives responses.

5.4.2 Distributed Coordination Function

The *Distributed Coordination Function* (DCF) makes use of the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol. In this protocol - where a station has a MAC frame to transmit, it listens to the medium. If the medium is idle, the station may transmit; otherwise the station must wait for the current transmission to complete before transmitting.

DCF doesn't include a collision detection function (i.e. CSMA/CD) because collision detection isn't practical on a wireless network. The dynamic range of the signals on the medium is very large, meaning that a transmitting station cannot effectively distinguish incoming weak signals from noise and the effects of its own transmission.

DCF makes use of a set of delays that amount to a priority scheme. The DCF transmission algorithm works as follows:

1. A station with a frame to transmit senses the medium. If the medium is idle, it waits to see if the medium remains idle for a time equal to IFS. If so - the station may transmit immediately
2. If the medium is busy - either because the station initially finds the medium busy or because the medium becomes busy during the IFS idle time; the station defers transmission and continues to monitor the medium until the current transmission is over.
3. Once the current transmission is over - the station delays another IFS. If the medium remains idle for this period, then the station backs off a random amount of time and again senses the medium. If the medium is still idle - the station may transmit. During the backoff time, if the medium becomes busy - the backoff timer is paused and resumes with the medium becomes idle.
4. If the transmission is unsuccessful, which is determined by the absence of an acknowledgement, then it is assumed that a collision has occurred. To ensure that the backoff maintains stability - binary exponential backoff is used. This provides a means of handling a heavy load with repeated failed attempts to transmit resulting in longer and longer backoff times, which help to smooth out the load. Without such a backoff - it could occur that two or more stations attempt to transmit at the same time causing a collision, then they attempt to retransmit again immediately, causing another collision.

Where a station wants to transmit to another station - first a *Request To Send* (RTS) is issued. If the destination ins clear to receive the transmission then a *Clear To Send* (CTS) is sent back. The source then transmits the frames, and the destination responds with an *Acknowledgement* (ACK) for each frame received - so the source the knows what has been received and what hasn't.

Example: DCF

We can see an example of DCF in action, albeit without any interframe spacing, as that's too complicated to make happen in TikZ.

In this example, C is a station within the range of A, and D is a station within the range of B. A sends a RTS to B before transmitting the first fragment. C, being within the range of A, hears this and uses the estimation of transmission time included in the RTS to enter NAV (Non-Active Mode). When B replies to A with the CTS, the stations within it's range, D, enter NAV.

A will then begin transmission of the first fragment, and sets a timer; if the timer expires before receiving the ACK frame from B, the whole process is repeated.

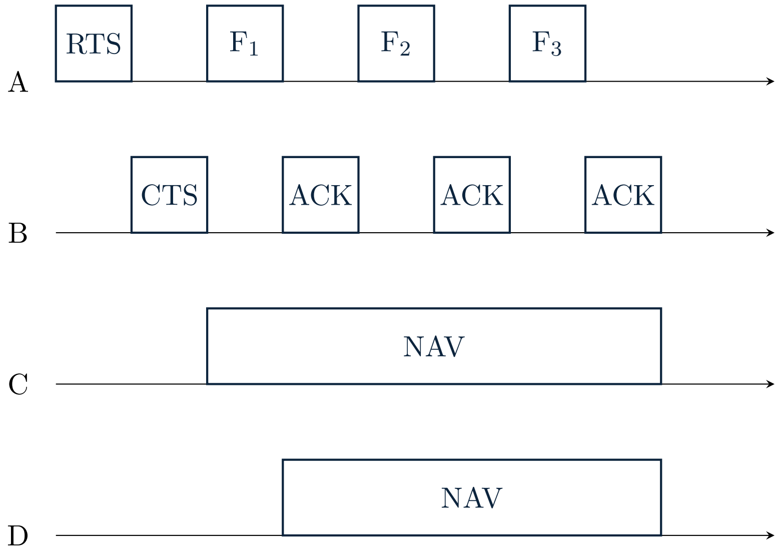

Figure 5.1: DCF Example

There are additional example diagrams of how DCF works in the slides available on Moodle.

5.4.3 Interframe Spaces

InterFrame Spaces (IFS) are used to control when data frames and control frames can be transmitted within the WLAN. There are different durations of IFS which are used by different scenarios - as they are of different priorities.

The *Short InterFrame Space* (SIFS) is used by priority traffic. For example, acknowledgements of transmissions, Clear To Send, or Poll responses within PCF. Multi-frame data units are also transmitted at SIFS intervals, so the multiple frames are kept together rather than them all being transmitted with potentially other frames jumping in front of them. SIFS is generally about $16\mu s$.

The *PCF InterFrame Space* (PIFS) is used by the coordinator when PCF operation is taking place. This is longer than SIFS as it is not the highest priority traffic on the network, but shorter than DIFS as it is higher priority than the general data transmission. PIFS is generally about $25\mu s$.

The *DCF InterFrame Space* (DIFS) is used for normal traffic on the network. DIFS is generally about $34\mu s$.

The InterFrame spacing works such that if the SIFS slot is not used, then the PIFS slot is available to be used. Then if the PIFS slot is not used, the DIFS slot is available to be used.

5.4.4 Super Frame

A super frame is used to prioritise time-sensitive traffic from wireless nodes. It is used to prevent coordinators continually issuing polls, locking out asynchronous traffic.

The first part of the super frame is where the point coordinator issues polls in a round-robin fashion (PCF). The second part is where the point coordinator idles and allows a contention period for asynchronous access (DCF).

5.4.5 Basic Access Method

An alternative access method to the Super Frame is that called 'Basic Access'.

After the medium becomes free - the first space of time is allocated for SIFS, then PIFS, then DIFS. SIFS is used for Acknowledgements, Clear to Send Messages & Poll Responses.

Example: Wireless Communication Between Two Devices

This example will show the communication between two subscribers, A and B, and an access point. Subscriber A wants to send 200 bytes to B; and B wants to send 150 bytes to A. This communication has a fragment size of 150 bits.

1. PIFS: AP → A - Beacon (34 bits)
   PIFS: AP → - B Beacon (34 bits)
2. SIFS: AP → A - CF-Poll (34 bits)
3. SIFS: A → AP - Data (150 bits)
4. SIFS: AP → B - Data (150 bits) + Poll (34 bits)
5. SIFS: B → AP - Data (150 bits) + ACK (34 bits)
6. SIFS: AP → A - Data (150 bits) + ACK (34 bits) + Poll (34 bits)
7. SIFS: A → AP - Data (50 bits) + ACK (34 bits)
8. SIFS: AP → B - Data (50 bits) + ACK (34 bits) + Poll (34 bits)
9. SIFS: B → AP - ACK (34 bits)
10. SIFS: AP → A - ACK (34 bits)
11. SIFS: AP → A - CF-END (34 bits)
    SIFS: AP → B - CF-END (34 bits)

We can then calculate the total communications time, knowing the data rate for this communication is 54MBps.

$$
\begin{aligned}
&= (PIFS \times 1) + (SIFS \times 10) + \frac{(34 \times 12 + (150 + 150 + 150 + 150 + 50 + 50) \times 8)}{54} \\
&= (25 \times 1) + (16 \times 10) + \frac{(34 \times 12 + (150 + 150 + 150 + 150 + 50 + 50) \times 8)}{54} \\
&= 25 + 160 + \frac{(408 + 5600)}{54} \\
&= 185 + \frac{6008}{54} \\
&= 185 + 111.26 \\
&= 296.26\mu s
\end{aligned}
$$

A diagramatic representation of this communication can be seen in the slides for this lecture and it's Tutorial.

5.5 MAC Protocol Data Units

The IEEE 802.11 frame is known as the *MAC Protocol Data Unit* (MPDU). This has a general format which is used for all data and control frames, but not all the fields in it are used in all the contexts.

The fields, their sizes and their uses can be seen in the below table. Fields marked with * are only present only in certain types of frames.

Field	Octets	Content
Frame control	2	Indicates the type of frame (control, management, or data) and provides control information which includes whether the frame is to for from a DS, fragmentation information and privacy information
Duration / Connection ID	2	If used as a duration field - indicates the time (in ms) the channel will be allocated for successful transmission of a MAC frame. In some control frames, this field contains an association or connection identifier
Address 1	6	The number and meaning of the 48-bit address fields depend on context. Transmitter address and receiver address are the MAC addresses of stations joined to the BSS that are transmitting and receiving frames over the WLAN. The service set ID (SSID) identifies the WLAN over which frame is transmitted. The source address and destination addresses are the MAC addresses of stations, wireless or otherwise. The source address may be identical to the transmitter address and the destination address may be identical to the receiver address.
Address 2 *	6	
Address 3 *	6	
Sequence Control *	2	Contains a 4-bit fragment number subfield used for fragmentation and reassembly, and a 12-bit sequence number used to number frames sent between a given transmitter and receiver
Address 4 *	6	
Quality of Service Control *	2	Information relating to the IEEE 802.11 QoS facility.
High throughput control *	4	Control bits related to the operation of 802.11n, 802.11ac and 802.11ad
Frame body *	0-7951	Contains a MSDU or fragment of MSDU
Frame Check Sequence	4	A 32-bit cyclic redundancy check

Table 5.1: MAC Frame Format

Page 6

Lecture - Bluetooth Networks

📅 2025-11-17 🕚 11:00 Asim

A *Bluetooth Network* is another type of Local Area Network. It is similar to an ad-hoc network, in that there is no fixed organisation, or fixed access point.

Bluetooth can be seen in many different applications including the following:

- Wireless headphones
- Wireless, bluetooth, mouse
- Wireless gaming controller

Bluetooth is a universal radio interface for ad-hoc wireless connectivity. It interconnects computers, their peripherals, handheld devices, PDAs and mobile phones. The bluetooth technology is embedded into devices, with a target cost of less than £5/device or can be sold as a separate USB antenna, costing less than £20/device.

Bluetooth has a short range, between 10m and 100m maximum and therefore uses very low power. The license for transmission is free under the Industrial Scientific and Medicine (ISM) band at 2.54GHz, and requires no registration. There are also no regulations which keep the cost low.

Voice and data transmissions are transmitted at about 1Mbps, or for Bluetooth 2.0 and later it reaches about 3Mbps.

6.1 Piconet

A *Piconet* is a collection of devices which are connected with a bluetooth connection. Within a single Piconet, there is a maximum of 8 devices; with one acting as a master and the remaining devices acting as a slave. The devices are connected in an ad-hoc manner and synchronised to the same hopping sequence.

There are 79 different frequencies which are used in the hopping sequence in a random order, which has been determined by the master. The benefit of using this frequency hopping is that it provides security (see frequency hopping from an earlier lecture); and that the changing in frequency should prevent interference between Piconets which are near each other in geographical locations.

Each Piconet has a unique hopping pattern to avoid interference with other Piconets. If a device wants to join a specific Piconet, it must synchronise with the same pattern.

There are a number of different states a *piconet* device can be in:

Park A device that is not actively participating in piconet traffic, but is still synchronised with the master device. The master device can park up to 255 slave devices and bring them back into active service when needed.

Hold A device that retains its synchronisation with the master device and can listen to *Synchronous Connection Oriented* (SCO) link packet transmissions. This device can also participate in or initiate the creation of other piconets.

Sniff A power-saving mode where the device sleeps and only listens for transmissions at a set interval.

Active The regular connected mode - where the device is actively transmitting or receiving data.

The Link Management Protocol (LMP) provides the state diagram for piconet devices.

6.2 Scatternets

A *Scatternet* is a collection of multiple Piconets. This means that a device which is part of a Scatternet - is able to be participate in more than one Piconets at once. A device which is in more than one Piconet is known as a *bridge device*, and can either be master in one, slave in one, master in both, or slave in both.

Slaves transmit once permission has been granted by the master using Time Division Multiplexing (TDM) access. This means that each slave gets a round-robin-like slice of time to broadcast it's message, and after all the other broadcasting devices have had a chance to transmit - it loops around again.[1] There is more on the methods of transmission later in this lecture.

6.3 Bluetooth Protocols

Bluetooth is defined as a layered protocol architecture which consists of a number of different protocols.

A diagram of the protocol stack can be found on slide 6 of the slides on Moodle.

The core elements of the protocols form a five-layer stack, the outline for which can be seen below:

Radio Specifics details of the air interface, including frequency, the use of frequency hopping, modulation scheme and transmit power

Baseband Concerned with connection establishment, addressing, packet forming, timing and power control

Link Manager Protocol (LMP) Responsible for link setup between Bluetooth devices and ongoing link management. Includes security aspects such as authentication and encryption, as well as control and negotiation of baseband packet size

Logical Link Control and Adaption Protocol (L2CAP) Adapts upper-layer protocols to the baseband layer. L2CAP provides both connectionless and connection-oriented services.

Service Discovery Protocol (SDP) Device information, services, and the characteristics of the services can be queried to enable the establishment of a connection between two or more Bluetooth devices.

Sitting on top of the L2CAP is RFCOMM which is the cable replacement protocol. RFCOMM presents a virtual serial port that is designed to make replacement of cable technologies as transparent as possible. Serial cables are the most common types of communications interfaces used within computing and communications devices, hence the decision to emulate this to reduce the amount of modification required in existing devices. RFCOMM provides binary data transport and emulates EIA-232 control signals over the Bluetooth baseband layer. EIA-232 (previously known as RS-232) is a widely used serial port interface.

The *Telephone Control Specification - Binary* (TCS-BIN) is a bit-oriented protocol that defines the call control signalling for the establishment of speech and data calls between Bluetooth devices. In addition - it defines mobility management procedures for handling groups of Bluetooth TCS devices.

The philosophy within the Bluetooth development team is to adopt protocols where they exist already, rather than defining more and more protocols. The adopted protocols include the following:

[1] See more in 'Spectrum Spread and Walsh Codes' lecture

PPP The Point-To-Point Protocol is an internet standard protocol for transporting IP datagrams over a point-to-point link

TCP / UDP / IP These are the foundation of the TCP/IP protocol suite

OBEX The object exchange protocol is a session-level protocol developed by the Infrared Data Association (IrDA) for the exchange of objects. OBEX provides functionality similar to that of HTTP but in a simpler fashiopn. It also provides a model for representing objects and operations - for example, vCard and vCalendar formats are transferred by OBEX

WAE/WAP Bluetooth incorporates the wireless application environment and the wireless application protocol into it's architecture

6.4 Frequency and Time Division Duplex

LTE-M supports both *Frequency Division Duplex* as well as *Time Division Duplex*.

Frequency Division Duplex (FDD) makes use of two different carrier frequencies, one for Download (DL) and one for Upload (UL). If the device supports *Full-Duplex FDD* (FD-FDD) it can perform reception and transmission at the same time. However if the device only supports *Half-Duplex FDD* (HD-FDD), it has to switch back and forth between reception and transmission.

Time Division Duplex (TDD) makes use of a single carrier frequency for both DL and UL transmission. TDD alternates between DL and UL periods, therefore it cannot perform reception and transmission at the same time.

Only a summary has been taken from the notes on the slides as this is talking about LTE-M which appears to not be directly related to Bluetooth networks. There is further information within the notes on the slides about oscillators and subframes.

6.5 Bluetooth Radio Specification

Naturally with all specifications we have seen so far in this module, there is a specification document which was originally part of the IEEE 802.15.1 specification, however this has since been disbanded.

Parameter	Details
Topology	Up to 7 simultaneous links (8 devices) in a logical star
Modulation	Gaussian FSK (with binary 1 represented by positive frequency deviation and binary 0 represented by negative frequency deviation)
Peak Data Rate	1Mbps
RF bandwidth	220kHz (-3dB), 1MHz (-20dB)
RF band	2.4GHz (ISM bnd)
RF carriers	23 / 79 channels
Carrier spacing	1MHz
Transmit Power	0.1W (implemented using LMP between master and slaves in a piconet)
Piconet access	FH-TDD-TDMA
Frequency hop rate	1600 hops/s
Scatternet access	FH-CDMA

Table 6.1: Bluetooth Radio Parameters

Frequency Hopping Spread Spectrum (FHSS) is used to hop between 79 different channels in a pseudorandom sequence. This sequence is determined by a master within a piconet, and all the slaves abide by this. The frequency is hopped 1600 times a second, giving a dwell time of $625\mu s$ per hop (which is also equal to the slot time). This hopping makes it hard to eavesdrop on the transmission.

Piconet access is allocated by the master. The master gets all the even numbered slots and the slaves get the odd numbered slots which they share using TDMA. Slots are labelled from 0 to $2^{27}-1$ and then cycles back to 0 again.

6.6 Bluetooth Packets

Bluetooth packets have three fields:

Access Code Used for timing synchronisation, offset compensation, paging and inquiry

Header Used to identify packet type and to carry protocol control information

Payload If present - contains user voice or data, and in most cases a payload header

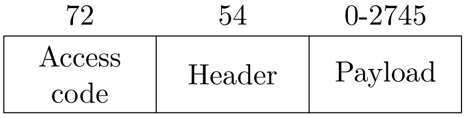

Figure 6.1: Bluetooth Packet showing field size in bits

6.7 Communications between Master and Slaves

As we have already seen - communications always take place between master and slave, never directly between slave and slave. This means if one slave wants to send another slave some information - this has to travel slave - master - slave. There are two different methods of communication - one being connection oriented and the other being connectionless.

The table showing the different Packet Types will be made available in the exam.

6.7.1 Synchronous Connection Oriented

Within Synchronous Connection Oriented (SCO), the master allocates fixed slots for communication between itself and a specific slave. These will come at a regular interval and generally are reserved in pairs so there's a slot for each direction, for example every third pair of slots will be used for communication between Master and Slave A. Payload transmitted may be 80, 160 or 240 bits.

The most reliable variant of SCO is 80 bits where master and slave get 800 slots/second which achieves 64000bps full-duplex which is used for a voice channel. Forward Error Correction (FEC) is used as SCO packets are never retransmitted, rather they use error correction techniques. While maintaining the maximum data rate (64Kbps) - the best quality transmission is where there is only a single slave with 1/3 FEC which means each payload of 80 bits is replicated 3 times. Where there are two slaves and we want to maintain our maximum data rate - we can replicate each payload twice and use 2/3 FEC. For 3 devices, each wanting 64Kbps transmission - no FEC is used.

SCO is used for real-time data, video and audio applications.

6.7.2 Asynchronous Connectionless

Asynchronous Connectionless (ACL) is used in the gaps which exist between reserved SCO blocks. ACL is a link between the master and all the slaves, again in a direct communication method - not in a slave-to-slave communication method.

ACL packets take up either 1, 3 or 5 slots. An ACL is returned by a slave if it was addressed in the preceding master-to-slave slot. This means ACL packets are bigger than SCO packets. Where an ACL packet is greater than 1 slot - the entire packet is transmitted on the same hopping frequency, rather than hopping mid-packet.

ACL packet transmission requires a short ($250\mu s$) settling time before the header can be transmitted - this is to allow for synchronisation between the master and slave.

There is no guaranteed bandwidth for ACL packets; and it does not include error correction - so retransmission is required.

Page 7

Lecture - Cellular Networks

2025-11-24 11:00 Asim

There are two YouTube videos linked on Moodle which are informative about this topic.

Definitions

Bandwidth The maximum amount of data that can be transmitted at a given time.

Frequency Reuse The technique used by cellular networks whereby they use the same frequency in two different geographical areas to increase capacity without requiring a greater spectrum

Cell A specific geographical area covered by a base station

Base Station A fixed-location radio transmitter and receiver which handles wireless communication within a cell

Channel A specific pathway, often a set of frequencies, used for transmitting data between a device and a base station

Control Channel A dedicated channel used for transmitting control data not user data

Forrward Channel The transmission path for signals sent from the base station to the device (also known as *downlink*)

Backward Channel The transmission path for signals sent from device to base station (also known as *uplink*)

MTSO (Mobile Telephone Switching Office) This is the central hub of of a cellular network that connects all the cell towers to each other and to the Public Telephone Switched Network (PTSN)

Hand Off The process of transferring an active call or data session from one cell or channel to another without interrupting service.

7.1 Single Cell Network

A single cell network is comprised of a base station which is comprised of an antenna, controller (which handles calls within the cell) and transceivers to communicate on the chosen channel for the cell.

The end-user devices connect to the base station; or, in the event that the base station is too far away - to a *relay* station which relay traffic back to the base station and vice versa.

In a similar way to that of the Bluetooth network or the WLAN architecture - the end user devices cannot communicate between each other directly, rather all communications must go through the base station.

Each cell has a number of available channels - this makes up the cell's bandwidth.

7.2 Multi-Cell Networks

Multiple single-cell networks are combined together to make a larger Cellular Network. The larger cellular network is comprised of 'hexagonal' cells, each with a base station in the middle of it. Relay stations are used on the borders of the cells to provide a stronger connection strength to each of the devices within that cell.

Each of the transmitters within the network are less than 100W powerful. Each cell is allocated a band of frequencies and is served by a base station consisting of a transmitter, receiver and control unit. Adjacent cells are assigned different frequencies to avoid interference or crosstalk. However, cells sufficiently distant from each other can use the same frequency band.

7.2.1 Geometry of Multi-Cell Networks

The 'hexagonal' patterns enables equidistant antennas within a multi-cell network. The radius of a hexagon is defined to be the radius of the circle that circumbscribes it, or equivalently, the distance from the centre to each vertex, or equal to the length of a side of the hexagon.

For a given radius, R, the distance between the cell centre and each adjacent cell centre is $d = \sqrt{3}R$

The area of cell can be calculated as $A = 2.6 \times R \times R$

7.2.2 Frequency Reuse

As we know - each cell is assigned a set of frequency channels. We cannot reuse these channels on directly adjacent cells as this would cause too much interference. However, we can reuse these channels if we separate the cells using the same frequencies with the same distance, D. The safe distance can be calculated as follows:

$$\frac{D}{d} = \sqrt{N}$$

$$\frac{D}{R} = \sqrt{3N}$$

Where R is the radius of a cell; d is the distance between centres of adjacent cells; D is the distance between centres of cells using the same frequency (safe distance); and N is the number of cells in a repetitious patters.

7.2.3 Network Structure

We saw above that each cell is a logical hexagon. These aren't real hexagons - as that's not how radio frequencies work - but we can think of them as hexagons.

Two cells using the same frequency create a Co-Channel Interference (CCI).

The re-use pattern distance can be evaluated by parameters i and j.

1. Move i steps from the reference cell in any hexagonal chain
2. Turn counter clockwise by 60 deg
3. Move j steps in that direction

So from this we can see that the number of cells in a cluster $N = i^2 + j^2 + i \times j$ with i and j from $\mathbb{N}$.

There is an example of this in the slides on Moodle - I'm not writing that out in TikZ.

Example: Calculating Network Capacity

If we take an example system with 32 cells, each with a radius of 1.6km, a total frequency bandwith of 336 channels and a reuse factor of 7.

Ex. 1 Find the area covered by a single cell

$$A = 2.6 \times R \times R = 2.6 \times 1.6 \times 1.6 = 6.65km^2$$

Ex. 2 Find the area covered by all cells

$$32 \times 6.65 = 213km^2$$

Ex. 3 Find the number of channels available in a single cell

$$\frac{336}{7} = 48$$

Ex. 4 Find the number of channels available in all the cells

$$32 \times 48 = 1536$$

Ex. 5 Find the number of users who can be served within all the cells

$$1536 \times 8 = 12228$$

(The 8 comes from TDMA mentioned in the previous lecture)

7.2.4 Increasing Cell Capacity

It is possible to increase capacity of a single cell when more customers join the cell than the cell can support. There are a few ways this can be achieved:

Adding New Channels Typically, when a system is set up in a region, not all of the channels are used. Therefore growth and expansion can be managed in an orderly fashion by adding new channels from the unused set

Frequency Borrowing Frequencies are taken from adjacent cells by congested cells. The frequencies can also be assigned to cells dynamically

Cell Splitting The distribution of traffic and topographic features is not uniform and this presents opportunities for capacity increases. Cells in areas of high usage can be split into smaller cells. The original cells start out at between 6.5 and 13km in size and these then get split down. The smaller cells can be subdivided yet again into picocells or femtocells. To use a smaller cell, the power level must be reduced to keep the signal within the cell. A radius reduction by a factor, F, reduces the coverage area and increases the required number of base stations by a factor of $F2$.

Cell Sectoring A cell is divided into a number of wedge-shaped sectors, each with its own set of channels. Typically there are three sectors per cell and directional antennas at the base station are used to focus signals on each sector. This can be seen in the triangular shape of typical cellular antenna configurations.

Microcells As the cells become smaller, antennas move from the top of tall buildings or hills to the top of small buildings or lampposts where they form *picocells*. Each decrease in size is accompanied by a reduction in power emitted by the base stations and mobile units. If we need to go smaller than a picocell - we can place a small cell inside buildings where they are known as *femtocells*. Femtocells may be restricted to certain users only, which is known as a *closed subscriber group*.

7.3 Operation of Cellular Networks

At the centre of each cell is a Base Station (BS). The BS includes an antenna, a controller and a number of transceivers for communicating on the channels assigned to that cell. The controller is used to handle the call process between the mobile unit and the rest of the network. At any time, a number of mobile units may be active and moving about within a cell communicating with the BS.

Each BS is connected to a *Mobile Telecommunications Switching Office* (MTSO). One MTSO serves multiple BSs. Typically MTSOs and BSs are connected by a wire, although wireless connection is becoming increasingly popular with technologies like WiMAX. The MTSO connects calls between mobile units. The MTSO is also connected to the public telephone or communications network and can make a connection between a fixed subscriber to the public network and a mobile subscriber to the cellular network. The mobile is also given access to the Internet. The MTSO assigns the voice channel to each call and performs hand-offs and monitors the call for billing information.

When a mobile unit is turned on, it scans and selects the strongest *setup control channel* used for this system. Cells repeatedly broadcast on different setup channels. As part of this process, the mobile unit will automatically select BS antenna of the cell within which it will operate; then a handshake takes place between the mobile unit and the MTSO controlling this cell (through the BS in the cell). The handshake is used to identify the user and register its location. As long as the mobile unit is on - this scanning procedure is repeated periodically to account for the motion of the unit. If the unit enters a new cell, then a new BS is selected.

A mobile-unit originated call starts by sending the number of the unit it wants to call on the preselected setup channel. This is done by first checking that the setup channel is idle by examining information in the forward channel. When an idle state is detected - the mobile unit may transmit on the uplink channel. The BS sends the request to the MTSO.

The MTSO then attempts to complete the connections to the called unit. The MTSO does this by sending a paging message to certain BSs to find the called mobile unit, although this depends on the called mobile unit's number and the latest information on the unit's whereabouts. The MTSO does not always know where every mobile unit is, especially if they have been in idle mode. Each BS transmits the paging signal on its own assigned setup channel.

The called mobile unit recognises its number on the setup channel which it is monitoring and responds to that BS which sends the response to the MTSO. The MTSO sets up a circuit between the calling and called BS. At the same time - the MTSO selects an available traffic channel within each BS's cell and notifies each BS, which notify its mobile unit. The two mobile units tune to their respective assigned channels

While the call is ongoing - the two mobile units exchange voice or data signals, going through their respective BSs and the MTSO.

If a mobile unit moves out of range of one cell and into the range of another during a connection - the traffic channel has ot change to the one assigned to the BS int he new cell. The system makes the change without interrupting the call or alerting the user.

When roaming - the use of the cellular network works in a similar way. The big difference being that the mobile unit is connecting to a different network to it's home network. The mobile unit is referred to as a guest on the roaming network.

7.4 GSM Network

The GSM network is a legacy form of cellular network, also known as 2G. Despite it being classed as a legacy system - it is still in use in many locations to this date. The GSM architecture is comprised of a number of sub-systems which are formally standardised in the GSM specification. This means that it is possible to purchase equipment from different vendors with the expectation that they will successfully interoperate. The GSM network is comprised of three core blocks:

- Mobile station
- Base Station Subsystem
- Network Subsystem

Each *Mobile Station* is comprised of two components:

Mobile Equipment (ME) is the physical terminal, such as a telephone of Personal Communications Service (PCS) device which includes the radio transceiver, digital signal processors and the SIM.

Subscriber Identity Module (SIM) is a portable device in the form of a smart card of plug in module which stores the subscriber's identification number, the networks the subscriber is authorised to use, the encryption keys and other information specific to the subscriber.

The *Base Station Subsystem* is comprised of two components, although there may be multiple base transceiver stations per base station controller.

Base Transceiver Station (BTS) defines a single cell. It includes a radio antenna, a radio transceiver and a link to a base station controller. A GSM cell can have a radius of between 100m and 35km depending on the environment.

Base Station Controller (BSC) may be co-located with a BTS or may control multiple BTS units and therefore multiple cells. The BSC reserves radio frequencies, manages the handoff of a mobile unit from one cell to another within the BSS and controls paging.

The *Network Subsystem* (NS) provides the link between the cellular network and the Public Switched Telecommunications Network (PTSN). The NS controls hand off between cells in different BSSs, authenticates users and validates their accounts, and includes functions for enabling worldwide roaming of mobile users. There are a number of components and, crucially, 4 databases involved in the NS:

Mobile Switching Centre (MSC) coordinates the functions and manages the databases.

Authenticaiton Centre database (AuC) is used for the authentication activities of the system - for example holding the authentication and encryption keys for all the subscribers in both the home and visitor location registers

Equipment Identity Register database (EIR) keeps track of the type of equipment that exists at the mobile station

Home Location Register database (HLR) stores information about each of the subscribers that 'belong' to it - this is both permanent information about the subscribers and temporary information about those subscribers.

Visitor Location Register database (VLR) stores the location of temporary subscribers (for example, those roaming)

7.4.1 Channel Bandwidth

The bandwidth of a channel is often around 25MHz; which would, for example, range from 1094MHz to 1119Mhz. Note that this would only be a channel for either uplink *or* downlink - there would need to be another same-sized channel for communication in the other direction. Between these channels there is a gap to avoid interference. This gap is often about 20MHz.

7.5 History of Cellular Communication

This is not a complete history. This history section covers the start of cellular connectivity through to the inception of 3G. We are currently on 5G connectivity. As of this module being taught, O2 have switched off the UK's 3G network...

7.5.1 First Generation: Analog Connections

The first generation cellular networks did not use encryption - this introduced risk on the control channels to get users identifications. The quality of a call of a first generation network was also far inferior to that of later generations - analog traffic is easily degraded by interferences, and there are practically no control or error control to overcome interference. There were also a number of inefficiencies with the spectrum and frequency allocation within - RF carriers were always allocated to users whether they are active (speaking) or idle within a given call.

The most common 1G service was originally developed in the 1980s and called the *Advanced Mobile Phone Service* (AMPS) was developed by AT&T. AMPS was deployed in Central & South America, Canada, Australia and China.

As goes the ways with technological development - there were a number of other systems developed and implemented at the same time which didn't have any interoperability. In the UK, Italy, Spain, Austria and Ireland - *Total Access Communication System* (TACS) was used; while in France the *Radiocom 200* was used, Germany made use of *C-450* and *Nordic Mobile Telephone* (NMT) was used in several other countries.

7.5.2 Second Generation: Digital, Voice & Data

Initially, D-AMPS was developed as an extension to AMPS which was an overlay to steal carriers from AMPS and convert them to digital signals using CDMA. This system was updated further under the IS-136 name to be fully digital and operate at about 800MHz and use TDMA. This system had a lot in common with GSM.

At the time of the conception of the Global System for Mobile Communication (GSM) - the 1G network in Europe was not compatible, so GSM was developed as the European standard for 2G. GSM had a goal to meet; it wanted to provide good speech quality; low terminal and service costs; international roaming; spectral efficiency; and be ISDN compatible. There are 4 different versions of GSM operating at different frequencies.

The GSM spectral allocation is 25MHz for base transmission (935-960MHz) and 25MHz for mobile transmission (890-915MHz). Other GSM bands have been defined outside of Europe. Users access the network using a combination of FDMA and TDMA, around radio frequency carriers every 200kHz which provide for 125 full-duplex channels. The channels are modulated at a data rate of 270.833kbps. As with AMPS - there are two different types of channels: traffic and control.

GSM uses a complex hierarchy of TDMA frames to define logical channels. Each 200KHz frequency band is divided into 8 logical channels defined by the repetitive occurrence of time slots. Each time slot is allocated 4.15ms.

As we saw above - this second generation makes use of CDMA for the D-AMPS system. CDMA was already proven and had a few advantages: frequency diversity, multi-path resistance and privacy. However, CDMA comes with a number of drawbacks: self-jamming, near-far problem, soft hand-off. In the IS-95 standards - it's defined to use 64 logical channels on the same bandwidth (1250KHz). Each channel has a chip code derived from the Walsh Matrix (sized 64×64). Meaning that 35 users can simultaneously transmit within the same bandwidth. There are 55 traffic channels and additional channels for paging, synchronisation, etc. They transmit at different data rates - ranging from 1200 to 14400bps.

7.5.3 Third Generation: WCDMA

The *International Mobile Telecommunication* IMT-2000 defined the spectrum, services and technologies for the 3rd generation. The spectrum used in most countries is now over 2GHz so the idea is to to define new allocations or re-use the 2nd Generation frequencies.

The third generation came with similar services to the second generation:

- High quality voice transmission
- Messaging (e-mail, fax, SMS, chat, etc)
- Multimedia (Playing music, viewing videos, film, TV, etc)
- Internet Access (web surfing, text, audio, and video)

The voice quality of 3G is comparable to that of the PTSN. A data rate of 144kbps is available to users in high-speed motor vehicles over large areas, or 384kbps is available to pedestrians standing or moving slowly over small areas. There is support for 2.048Mbps for office use as well as symmetrical & asymmetrical data transmission rates. 3G supports both packet-switched and circuit-switched data services and has an adaptive interface to the Internet to reflect efficiently the common asymmetry between inbound and outbound traffic. It makes more efficient use of the available spectrum in general and provides support for a wide variety of mobile equipment. 3G also provides the flexibility to allow the introduction of new services and technologies.

Page 8

Seminar - Cellular Networks Exercises

📅 2025-11-27 🕙 10:00 Asim

Example: Cell Networks Calculations

A mobile phone service provider has leased a spectrum starting from the frequency 1094MHz and has adopted the re-use factor of 4, which is 4 cells per cluster to cover greater Manchester area. The greater Manchester area has a population of approximately 3Millions and a geographical area of 1277km-square. The service provider decides to set up cells of 1.5KM radius each. The encoding system is based on CDMA (64-bits), only and the service provider is planning to offer about 300 KHz of bandwidth to all its customers. The regulation for bandwidth assignment is the standard 25MHz band on the up and down channels and 20MHz of gap to avoid interference. The service provider has done some market research and found out 75% of the population use mobile phones and 30% out of them might be potential clients.

Ex. 1: What are the border frequencies for this spectrum?

Up-link will start at 1094MHz and be 25MHz wide: 1094-1119MHz

Guard Band (gap between up & down) will be 20MHz wide: 1119-1139MHz

Down-link will be 25MHz wide: 1139-1164MHz

Ex. 2: What are the values of i and j for the design of the cluster?

Question says the reuse factor of cells is 4 - therefore plugging 4 into the $N = i^2 + j^2 + i \times j$, we get either $i = 0$ & $j = 2$ or $i = 2$ & $j = 0$.

Ex. 3: How many channels are there per cell?

Taking the size of a frequency band, 25MHz and the offered bandwidth, 300KHz, we can calculate the number of channels in total: $\frac{25000000}{300000} = 83$

We can then use the 83 channels, and our re-use factor of 4 from the question: $\frac{83}{4} = 20.75 \approx 20$

Ex. 4: How many cells are needed to cover the whole Greater Manchester are?

We can calculate the size of a single cell by calculating the area of the hexagonal cell: $2.6 \times 1.5 \times 1.5 = 5.82km^2$.

Knowing that the area of Manchester is 1277km^2, we can divide the two to find the total number of cells needed: $\frac{1277}{5.82} = 218.29 \approx 219$

Ex. 5: How many mobile users can be supported at the same time without co-channel interference in the greater Manchester are? Does the current infrastructure support the predicted number of customers?

We can calculate the number of mobile users supported at the same time for the Greater Manchester Area using the total number of cells needed, 219, the number of channels per cell, 20, and the number of bits used in the encoding system (CDMA), 64: $219 \times 20 \times 64 = 280320$.

We can find the number of mobile users as given in the question: $0.75 \times 3000000 = 2250000$. Then we can calculate the potential clients using $0.3 \times 250000 = 675000$.

As $280320 < 675000$, the current infrastructure does not support the potential number of clients for this network.

There are additional examples of this type of question, well exactly the same question with different numbers, on Moodle.

Page 9

Lecture - Satellite Networks

📅 2025-12-01 🕚 11:00 Asim

A *satellite* is a large reflector which orbits in the sky around the earth. Satellites have several transponders, which are electronics that listen to different portions of the spectrum - each amplifying the incoming signal (the uplink), changing its frequency then broadcasting it to earth (the downlink).

The sky-based satellites are combined with components on the ground which are called *gateway stations*. These gateway stations are equipped with antennas directed towards the satellite, and some will have network control centres and operation control centres which deal with satellite management and orbit control.

A satellite communication system works where two or more stations on or near the earth (gateway stations) communicate via one or more satellites that serve as relay stations in space.

There are a number of different ways of categorising communication satellites:

Coverage Area which could be global, regional or national. The larger the coverage area, the more satellites must be involved in a single networked system

Service Type which could be fixed service satellite (FSS), broadcast service satellite (BSS) and mobile service satellite (MSS).

General Usage which could involve commercial, military, amateur, or experimental

The area of coverage for a satellite communication system exceeds that of a terrestrial system. In the case of a geostationary satellite - a single antenna is visible to about one quarter of the earths surface. Spacecraft power and allocated bandwidth are limited resources that call for careful tradeoffs in gateway stations / satellite design parameters. The conditions between communication between satellites are more time invariant than those between satellite and earth station or between two terrestrial wireless antennas. Therefore, satellite-to-satellite communication links can be designed with great precision. Transmission cost is independent of distance, within the satellites area of coverage. Satellites can transmit in broadcast, multicast and point-to-point mode. Satellites provide very high bandwidth and the quality of transmission is usually extremely high. For a geostationary satellite - there is an earth-satellite-earth propagation delay of about a quarter of a second. However, in many cases - a transmitting earth station can receive its own transmissions.

Satellite can be used for any of the following: telephone communications; television, radio or digital cinema broadcasting; amateur radio; providing internet access; military activities; and disaster management.

9.1 Topology of Satellite Networks

There are two common topologies which satellite networks may adopt.

Firstly, it is possible for a satellite to provide a point-to-point link between two distant ground-based antennas.

Alternatively, the satellite provides communication abilities between one ground-based transmitter and a number of ground based receivers. There exists a variation of this configuration wherein there is two-way communications among earth stations with one central hub and many remote stations.

9.2 Satellite Orbits

Satellite orbits may be classified in a number of ways:

- The orbit may be circular, with the centre of the circle at the centre of the earth, or elliptical - with the earths centre at one of the two foci of the ellipse.
- A satellite may orbit around the earth in different planes. An equatorial orbit is directly above the earth's equator. A polar orbit passes over both poles; while other orbits are referred to as inclined orbits.
- The altitude of communications satellites is classified as geostationary orbit (GEO), medium earth orbit (MEO), and low earth orbit (LEO) - more on this in a moment...

The *orbital period* is the time it takes for the satellite to go around the earth and can be calculated as follows:

$$\text{Orbital Period} = \frac{1}{1000} \times \text{(Radius of the earth + Altitude of a satellite)}^{1.5}$$

Example: Orbital Periods

Ex. 1: The Moon

$$\text{Orbital Period} = \frac{1}{1000} \times (6378Km + 384000Km)^{1.5} = 2439090sec \approx 1month$$

Ex. 2: GEO Satellite

$$\text{Orbital Period} = \frac{1}{1000} \times (6378Km + 35786Km)^{1.5} = 86579sec \approx 24hours$$

Satellite orbits are within certain altitude ranges, based on distance from earth. There are also two *Van Allen Belts*, one between 2000 and 5000km and the second between 15000 and 20000km. These belts are layers containing charged particles so therefore any satellite falling into that layer will be destroyed by the particles.

9.2.1 Geosynchronous (Geostationary) Satellites

Geosynchronous, or Geostationary, (GEO) satellites orbit at about 35786Km above the earth's surface moving at the same speed as the earth. They were introduced in 1945. The Geostationary orbit has a number of advantages to it:

- The satellite is stationary relative to the earth, meaning there is no problem with frequency change due to the relative motion of the satellite and antennas on earth (Doppler Effect)
- Tracking of the satellite by its earth stations is simplified
- At 35,863km above the earth - the satellite can communicate with roughly a quarter of the earth; meaning that three satellites in geostationary orbit separated by 120° covers most of the inhabited portions of the entire earth excluding only areas near the north and south poles.

However, in saying this - there are problems with Geostationary satellites:

- The signal can get quite weak after travelling over 35000km
- The polar regions and the far northern and southern hemispheres are poorly served
- Even at the speed of light (about 300000km/s), the delay in sending a signal from a point on the equator beneath the satellite to the satellite is in fact 0.24s. For other locations not directly under the satellite - the delay is even longer.

If the satellite link is used for telephone communication, the added delay between when a person speaks and when that person receives a response is almost 0.5s. This is definitely a noticeable delay and that's not good.

Geostationary satellites use their assigned frequencies over a very large area. This can be a desirable feature for Point-to-Multipoint (PMP) applications such as broadcasting TV programmes; however for Point-to-Point communication it is a very wasteful use of the spectrum. Some special sports and steered beam antennas, which restrict the area covered by the satellite's signal, can be used to control the footprint or signalling area.

To launch a GEO satellite - there is a high cost to send rockets and a high delay to get a rocket launched. Despite this, the satellites - once in orbit in the right place, can last for a long time as there is no atmospheric friction.

Up-to 180 satellites can exist in the GEO orbit, each with 2° between them.

To solve some of these problems - orbits other than geostationary have been designed for satellites. An alternative is *Low-Earth-Orbiting* (LEO) and *Medium-Earth-Orbiting* (MEO) satellites which will be covered subsequently.

9.2.2 Low Earth Orbit Satellites

Low Earth Orbit (LEO) satellites orbit under 1500km. The orbit period (time to do a complete lap of the earth) ranges from 1.5 to 2 hours, with a diameter or coverage at about 8000km.

The round-trip signal propagation delays is typically less than 20ms, with a single satellite visible for about 20 minutes. The system must cope with large Doppler shifts.

In the lower atmosphere than a GEO satellite, LEO satellites see more atmospheric drag which results in orbital deterioration. They will eventually fall back towards earth and they have a short life. Usually they are put back on course with space boosters, which require fuel and cannot use solar power; or use of a space shuttle as in the Hubble Telescope. Practical applications of this system require multiple orbital planes to be used, each with multiple satellites in orbit.

Communication between two earth stations will typically involve handing off the signal from one satellite to another.

LEO satellites have a number of advantages over GEO satellites:

- LEO signals are much stronger than GEO signals for the same transmission power
- LEO coverage can be better for localised so that the spectrum can be better conserved. For this reason - this technology is currently being proposed for communicating with mobile terminals and personal terminals that need stronger signals to function. However, to provide broad covereage over 24 hours - many satellites are needed.

There have been a number of commerical proposals to use clusters of LEOs to provide communication services. These proposals can be divided into two categories.

A *Little LEO* is intended to work at communication frequencies below 1GHz using no more than 5MHz of bandwidth and supporting data rates of up to 10kbps. It's aimed for these systems to be used for paging, tracking and low-rate messaging.

A *Big LEO* works at frequencies above 1GHz and supports data rates up to a few megabits per second. These systems tend to offer the same services as those of small LEOs with the addition of voice and positioning services.

> There is some more information about Little LEO and Big LEO in the notes on the slides on Moodle. Not included here as it wasn't discussed in the Lecture.

9.2.3 Medium Earth Orbit Satellites

Medium Earth Orbit (MEO) satellites orbit in the altitude range of 5000 to 15000km. The orbit period is approximately 6 hours, and they have a far longer life from LEO satellites due to the fact that they are further away from the earth atmosphere and its friction. They have a moderate round-trip delay of 50ms.

A common example of MEO satellites is GPS (Global Positioning System), which is comprised of 24 satellites orbiting with the MEO range, at 18000km. GPS uses the triangulation principle to compute the position of an object on earth where three circles intersect on one point, or 4 spheres (4 satellites signals) intersect on 1 point. The GPS receiver sends signals to 4 MEO satellites and measures how long it takes for the signals to return, then calculates the position on the earth and also the location on the map.

Few MEO satellites orbit above 10000km as the deployment cost and propagation delay are significant without any additional advantages.

9.3 Frequency Bands

There are a number of frequency bands available for satellite communications. As the frequency range increases, their bandwidth also increases. However, generally the higher the frequency is, it's found that there is a greater effect of transmission impairments.

The Mobile Satellite Service (MSS) is allocated frequencies in the L and S bands which, when compared to higher frequencies, there is found to be a greater degree of refraction and greater penetration of physical obstacles such as foliage and nonmetallic structures. These characteristics are desirable for mobile services; however, the L and S bands are also heavily used for terrestrial applications. Therefore there is competition amongst the various microwave services for L and S band capacity.

For any given frequency allocation for a service - there is always an allocation of an uplink band and a downlink band, with the uplink band always of a higher frequency. The higher frequency suffers greater spreading (also known as free space loss) than the lower frequency counterpart. This is compensated by the earth station where the earth station is of higher power than the satellite.

Band	Frequency Range	Total Bandwidth	General Application
L	1 - 2GHz	1GHz	Mobile Satellite Service (MSS)
S	2 - 4GHz	2GHz	MSS, NASA, Deep Space Research
C	4 - 8GHz	4GHz	Fixed Satellite Service (FSS)
X	8 - 12.5GHz	4.5GHz	FSS military, terrestrial earth exploration and meteorological studies
Ku	12.5 - 18GHz	5.5GHz	FSS, broadcast satellite service (BSS)
K	18 - 26.5GHz	8.5GHz	BSS, FSS
Ka	26.5 - 40GHz	13.3GHz	FSS

Table 9.1: Frequency Band Allocation

9.4 Transmission Impairments

As we can expect - satellite transmissions can be impaired. The performance of a satellite link depends on three factors:

- Distance between earth station antenna and satellite antenna
- (For downlink only) Terrestrial distance between the earth station antenna and the “aim point” of the satellite
- Atmospheric Attenuation

Distance is perhaps the easiest to comprehend - with the higher the frequency, the greater the loss. Losses seen at points on the surface of the earth away from the equator but still visible from the satellite will be somewhat higher.

There is an equation in the notes on the slides which can be uses to calculate the *Free Space Loss*.

The footprint of a satellite’s downlink signal is reasonably fixed. The centre of this area will receive the highest radiated power, with this value reducing as you move away form the centre point in any direction.

Atmospheric Attenuation refers to the weather impacting the signals as they’re transmitted between earth station and satellite. The biggest impacting factor here is oxygen and water. Attenuation due to water is commonly found when it’s humid and more pronounced with fog and rain. Atmospheric attenuation can also depend on the frequency - with a higher frequency causing a greater effect.

9.5 Satellite Capacity Allocation

As with other communication systems - there has to be some governance as to what can communicate and when, with which encoding to ensure that all devices in the network get fair and equal access to the medium. Satellite communication is no exception to this and the communication techniques used will fall into one of the three following categories:

- Frequency Division Multiple Access (FDMA)
- Time Division Multiple Access (TDMA)
- Code Division Multiple Access (CDMA) - not covered in detail in this section

9.5.1 Frequency Division Multiple Access

Frequency Division Multiple Access (FDMA) works similarly to that where we’ve seen it before - in that the signal wave is modulated onto a carrier wave to position it differently in the frequency spectrum. The whole allocated frequency spectrum is divided into a series of smaller bands, each 36MHz with a guard band before the next band, which in this case are 4MHz. These smaller bands may contain one or more transmitted signals - depending on the required bandwidth for a given signal.

There are two types of *polarisation* which we see signals modulated onto - vertical and horizontal. Vertical Polarisation is where the signal moves sideways in a horizontal plane, while with Horizontal Polarisation - the signal oscillates up and down in the vertical plane. Signals modulated with FDMA make use of both of these polarisations - with the channels alternating between horizontal and vertical.

Example: Polarisation

We can see how Polarisation works if we take a set of channels each 40MHz with inclusive guard-band of 4MHz (so this means we don’t need to write the guard bands separate). The following list of channels shows their channel number, direction, frequency range as well as their carrier frequency.

- C1 (Horizontal): 3700MHz - 3740MHz (CF 3720MHz)

- C2 (Vertical): 3720MHz - 3760MHz (CF 3740MHz)
- C3 (Horizontal): 3740MHz - 3780MHz (CF 3760MHz)
- C4 (Vertical): 3760MHz - 3800MHz (CF 3780MHz)
- etc...

The channels used will repeat alternating between Horizontal and Vertical until they reach the end of the allocated bandwidth. Note the overlap between horizontal and vertical channels.

9.5.2 Time Division Multiple Access

Time Division Multiple Access (TDMA) works similarly to that where we've seen it before. The uplink of TDMA works in that stations take it in turns to use the uplink channel and may put a burst of data in their assigned time slot. Between each satellite's time allocation, handled in a round-robin method, there is a short guard time. The downlink works in much the same way with each signal to be transmitted being taken in turn.

The TDMA frame begins with two reference bursts to define the beginning of the frame. Two bursts are used, each provided by a different earth station, so that even if one of the reference stations are lost - the system can continue to function. Each reference burst begins with a carrier and bit timing recovery pattern, which is a unique pattern that allows all stations to synchronise to a master clock. Each of the N stations are then assigned one or more slots in the frame, where the stations use an assigned slot to transmit a burst of data. The bursts of data consist of a preamble, and the user information. The preamble contains control and timing information as well as the identification of the destination station. Each burst within a frame is separated by guard times to ensure that there is no overlap which prevents signal garbling.

It is beginning to be more common to see TDM used and FDM not used for a number of reasons:

- The continuing drop in the cost of digital components
- The advantages of digital techniques including the use of error correction
- The increased efficiency of TDM due to the lack of intermodulation noise

Page 10

Lecture - Software Defined Networking

📅 2025-12-08 🕚 11:00 👨‍🏫 Asim

Software Defined Networking (SDN) is an alternative paradigm of networking which separates the network control from the data forwarding by using software. This allows for centralised, programmable management of the entire network.

10.1 Planes

No, not the things flying in the sky...

10.1.1 Data Plane

The data plane transmits and receives data packets. It focuses on forwarding data packets based on routing destinations; and uses protocols such as TCP/IP and UDP.

10.1.2 Control Plane

The control plane is concerned with routing of data, naming of devices, declaring policies and performing security checks. It is responsible with managing and controlling the network. The control plane uses protocols such as OSPF, BGP and MPLS.

10.1.3 Management Plane

The management plane configures and monitors network devices. It uses protocols such as SSH, Telnet and SNMP.

10.2 Server Virtualisation

Traditionally, servers were used in a single server per application paradigm. Taking into account testing, development, pre-production, production, and disaster recovery - a single application may require 3 to 5 servers. However the traditional workload of a single one of these servers would be very low which is obviously bad.

Server Virtualisation is a way to consolidate servers by allowing a single physical host to run multiple workloads. A ‘virtual server’ is a software implementation that executes programs like a real server. Multiple virtual servers can work simultaneously on one single physical host server server. Therefore instead of operating many servers at low utilisation, virtualisation combines the processing power onto fewer servers that operate at a higher total utilisation.

Virtualisation improves scalability, reduces downtime and enables faster deployments. It can also speed up disaster recovery efforts because virtual servers can restart applications much faster than physical servers. With virtualisation, entire systems can be moved from one physical host to another in just a few seconds - to optimise workloads or to perform maintenance without causing downtime.

Some virtualisation solutions also have built-in resiliency features such as high availability, load balancing and failover capabilities. Virtualisation has become commonplace in large data centres; however is less common (as of 8 years ago when these slides were written) in smaller data centres.

The application running directly on a physical host server is called a Hypervisor, specifically a type 1 hypervisor.

Server virtualisation masks server resources (both hardware and software) from users. Virtual server hosts can host multiple VM servers.

However, virtual servers introduce the need for virtual LANs to be configured. The physical server and the VM need to use the same switch port. The network needs to be reconfigured whenever the VM migrates.

10.3 Dynamic Configuration of Network Resources

Especially when using Server Virtualisaiton, but also more generally in modern networks - there is a need for dynamic configuration of network resources.

This is needed for management of traffic amongst virtual servers for the following:

- To maintain consistent database images
- Invoke security functions such as access control
- Server-to-server traffic changes with location and time

This is also needed for performance management of enterprise networks:

- Users access enterprise resources through mobile devices
- Load changes dynamically with location and time
- Manual configuration of network servers take long time due to different vendors and different interfaces of devices
- Managers need to configure equipment from different vendors separately on per session per application basis

10.3.1 Traditional Network Configuration

Network managers must be able to respond to changing resources, quality of service needs, and security requirements by providing differentiated QoS levels and security levels for individual traffic flows. The network manager must configure each vendor's equipment separately and adjust security parameters on a per-session, per-application basis.

In a large enterprise network, every time a new VM is brought up, it can take hours or even days for network managers to do the necessary reconfiguration.

10.4 SDN Architecture

The architecture of SDN centres around a central controller who performs all complex functions, including routing, naming, policy declaration and security checks. This controller is the SDN control plane and consists of one or more SDN servers. The SDN controller defines the data flow that occurs in the SDN data plane.

Each flow through the network has to first get permission from the controller, which verifies the communication is permissible by the network policy. If the controller allows a flow, it computes a route for the flow to take and adds an entry for that flow in each of the switches along the path. With all complex functions handled by the controller, the switches are able to solely manage the flow tables.

Communication between the controller and the switches uses a standardised protocol and API. This is most commonly in line with the OpenFlow specification.

10.4.1 OpenFlow

OpenFlow is the protocol used for communication between the SDN Controller and switches. This communication takes place over SSL.

Each OpenFlow switch will connect to other OpenFlow switches and sometimes to end-user devices. Within each switch, there are a series of tables which are typically implemented in hardware or firmware, that are used to manage the flows of the packets through the switch. The OpenFlow specification defines a number of types of tables in the logical switch architecture:

Flow Table matches incoming packets to a particular flow and specifies what functions are to be performed on the packets. There may be multiple flow tables that operate in a pipeline fashion. A flow table may direct a flow to a group table which may trigger a variety of actions that affect one or more flows.

Meter Table can trigger a variety of performance related actions on a flow.

A *flow* in this context refers to a sequence of packets traversing a network that share a set of header field values (source IP, destination IP, VLAN identifier, etc).

An OpenFlow switch encapsulates and forwards a flow's first packet to a SDN controller. The switch forwards incoming packets out of the appropriate port based on the flow table. The switch can drop packets on a particular flow, either temporarily or permanently as dictated by the controller.

There are three different communication classes defined in the OpenFlow protocol. *Controller to Switch* is used by the controller to program, configure and retrieve information from the switch. *Asynchronous (switch to controller)* is initiated by the OpenFlow switch to and sends to the controller to report packet arrivals, state changes, errors, etc. *Symmetric (switch to controller) and Controller to switch* allows either to side to send message without solicitation; for example hello or echo messages to identify if the control channel is still alive and available.

10.4.2 Domains

It is possible for multiple SDNs to exist in proximity to one another. The existence of multiple domains creates a requirements for individual controllers to communicate with each other via a standardised protocol and to exchange routing information. A protocol, SDNi, is being developed for interfacing between SDN domain controllers. It's functionality includes the following:

- coordinate flow setup originated by applications containing information such as path requirement, QoS and SLA
- Exchange reachability informaiton to facilitate inter-SDN routing; allowing a single flow to traveerse multiple SDNs and have each controller select the most appropriate path when multiple such paths are available

As of when these slides were written, SDNi is discussed as something coming in the future. It doesn't appear SDNi has gone anywhere, with the IETF noting the draft as 'no longer active'.

10.4.3 Scope and Scale

Typically a physically centralised control plane is adapted for SDN. It can be implemented in a single server but this is a single point of failure and a potential bottleneck. There may also be reliability or scalability issues.

The alternative to this is a logically centralised distributed control plane. This comprises of physically distributed elements that interface to each other through east & west bound interfaces. The data plane and the applications only see a single logical controller.

10.5 SDN Flows

As we already know, the OpenFlow protocol uses tables to route packet sequences (flows) from the sender to the receiver. When a flow arrives to a OpenFlow switch, a table lookup is performed. If no matching route is found, a request to the controller is made; the reply from the controller is handled in one of three different modes - Proactive, Reactive or Hybrid.

10.5.1 Reactive Mode

In reactive mode, the controller is queried when packets arrive and there is no entry or an expired timer has expired for the required destination. The controller creates and installs a rule in the corresponding switch if necessary. Importantly, the rule is only installed on switches which will be implicated in the forwarding of this flow.

Example: Reactive Mode

In this example Host 1 (H1) sends a packet to Host 2 (H2) which is split into two fragments: P1.F1 and P1.F2. H1 also sends a packet to Host 3 (H3) which is split into three fragments: P2.F1, P2.F2 and P2.F3.

The network in this example is made up of the following: H1 connects to Switch 1 (S1) which is connected to Switch 4 (S4). S4 is connected to both Switch 2 (S2) (connected to H2) and Switch 3 (S3) (connected to H3). All Switches are connected to the SDN Controller (C).

The reactive data flow works as follows:

- H1 (P1.F1) → S1 (Req) → C (Entry) → (S1, S4, S2)
- S1 (P1.F1) → S4 (P1.F1) → S2 (P1.F1) → H2
- H1 (P1.F2) → S1 (P1.F2) → S4 (P1.F2) → S2 (P1.F2) → H2
- H1 (P2.F1) → S1 (Req) → C (Entry) → (S1, S4, S3)
- S1 (P2.F1) → S4 (P2.F1) → S3 (P2.F1) → H3
- H1 (P2.F2) → S1 (P2.F2) → S4 (P2.F2) → S3 (P2.F2) → H3
- H1 (P2.F3) → S1 (P2.F3) → S4 (P2.F3) → S3 (P2.F3) → H3

10.5.2 Proactive Mode

In proactive mode, the controller populates flow tables for all possible traffic matches. These are static entries and they are installed ahead of time of the transmission. Following this - no request is sent to the controller since all incoming flows will find a matching entry. This is advantageous (over reactive mode) as all packets are forwarded at line rate so there is no delay; however it is at a disadvantage because the flow table when implemented in associative memory is very expensive for large tables.

Example: Proactive Mode

In this example Host 1 (H1) sends a packet to Host 2 (H2) which is split into two fragments: P1.F1 and P1.F2. H1 also sends a packet to Host 3 (H3) which is split into three fragments: P2.F1, P2.F2 and P2.F3.

The network in this example is made up of the following: H1 connects to Switch 1 (S1) which is connected to Switch 4 (S4). S4 is connected to both Switch 2 (S2) (connected to H2) and Switch 3 (S3) (connected to H3). All Switches are connected to the SDN Controller (C).

The reactive data flow works as follows:

- H1 (P1.F1) → S1 (Req) → C (Table) → (S1, S2, S3, S4)
- S1 (P1.F1) → S4 (P1.F1) → S2 (P1.F1) → H2
- H1 (P1.F2) → S1 (P1.F2) → S4 (P1.F2) → S2 (P1.F2) → H2
- H1 (P2.F1) → S1 (P2.F1) → S4 (P2.F1) → S3 (P2.F1) → H3
- H1 (P2.F2) → S1 (P2.F2) → S4 (P2.F2) → S3 (P2.F2) → H3
- H1 (P2.F3) → S1 (P2.F3) → S4 (P2.F3) → S3 (P2.F3) → H3

Freely Given, Highly Questionable

These notes are a byproduct of my learning process - mistakes, bad jokes and snark included. If you take an error of mine as gospel and ruin your perfect 1st, that is a *you* problem boo. I am a student, not a prophet, use a textbook if you want certainty.

Yes, they have been typeset in LaTeX; I'd rather typeset in stone with a blunt chisel than use Word Online for another minute. If the formatting looks obsessive, it's because it was either this or a breakdown. You can find the source on my GitHub.

Thomas Boxall is a Computer Science Student studying at the University of Portsmouth, UK. In all honesty, he should probably be revising or doing some work towards his dissertation, not fondling the template of his notes again. You can find Thomas online at thomasboxall.net