Introduction to Computer Networks and the Internet COSC 264

Physical Layer

Dr. Andreas Willig

Dept. of Computer Science and Software Engineering
University of Canterbury, Christchurch

UoC, 2019

Scope and Resources

- Introduction to communications and physical layer concepts
- Introducing some major problems to be solved in the physical layer
- Topics:
  - Communications fundamentals
  - Baseband transmission and signal impairments
  - Broadband / Passband transmission
  - Frame synchronization methods
- Resources:
  - Several of the major computer networking textbooks have chapters on the physical layer, e.g. [25], [27]
  - There is furthermore a large collection of communications textbooks for engineers, e.g. Gallager [9], Proakis [20], Benedetto/Biglieri [1]

Outline

Outline

1. Fundamentals
   - Digital and Analog Transmission
   - Structure of Communication Systems
2. Baseband Transmission
3. Passband Transmission
4. Synchronization

Digital Data

- We will frequently refer to "digital" and "analog" data and need to understand these notions
- Digital/discrete data refers to a (finite or countably infinite) sequence of **discrete symbols**
- A symbol is a member of a **finite** set which is also known as the **alphabet**
- Example: An english word is made up of a sequence of letters, alphabet is $\{a, \ldots, z\}$
- Example: A byte is made up of a sequence of bits, each taken from $\{0, 1\}$
- Example: The Morse code is a sequence of symbols, alphabet is $\{\text{dot}, \text{dash}, \text{pause}\}$
- Nowadays many further types of data are represented digitally, e.g. audio, video, images
- Example: an image is made up of pixels, each pixel is represented by a 24 bit number encoding its colour
- Since it is possible to represent all members of a finite alphabet by groups of bits (for example, the seven- or eight-bit representation of ASCII characters), we assume that **all digital data is just a sequence of bits**

Analog Data

- Analog data can take on an uncountable number of values (e.g. real numbers)
- Examples: voltages on a cable, amplitude of sound signal, position of particle in 3D space
- Some fundamentally analog transmission systems include: human speech, the old FM radio and TV broadcast systems, analog cameras
- In the last few decades our world became more and more digitized, many of the old analog signal types are nowadays converted into digital form and processed by computers
- Within computers and networks, all the data of interest (e.g. data or instructions processed by the processor, messages / packets in networks) is represented as digital data, perhaps after some conversion from analog to digital (A/D conversion)

Transmission of Digital and Analog Data

- In networking we are interested in transmission of analog or digital data
- There is a fundamental difference in the expectations for the "quality" of the data transfer
- When transmitting digital data, we want to recover the transmitted sequence of bits **exactly**, despite any distortions and errors introduced by the channel
- When the receiver receives a signal, it chooses one out of a finite number of alternatives (e.g. 0 or 1 in the case of bits), as the transmitted data must have been one of these
- With some luck the receiver makes the correct choice and the system design centers fundamentally around **maximizing the probability that this actually happens**
- When transmitting analog data again the channel might introduce distortions, but the receiver cannot exactly recover the transmitted data, as it is faced with an (uncountably) infinite number of alternatives for the transmitted signal
- Hence, in analog system design the goal is not to try to recover the transmitted signal exactly, but to extract an as-good-as-possible **approximation** to it

Transmitting Digital Data Using Analog Signals (1)

- In computers and communication networks, we deal fundamentally only with binary data
- However, most transmission media only allow transmission of an **analog physical signal**, taking on a continuous range of values over a continuous time interval
- In the following, we use the words **signal** or **waveform** to denote the evolution of some physical quantity (e.g. a voltage) over time
- We use signals / waveforms for data transmission by modifying them according to the data to transmit and the properties of the channel, this is also called **modulation**
- Electrical signals: voltages, phases or currents over time
- Wireless signals: amplitudes, frequencies or phases of a sinusoidal radio wave over time
- Modulation is a key function of the physical layer!

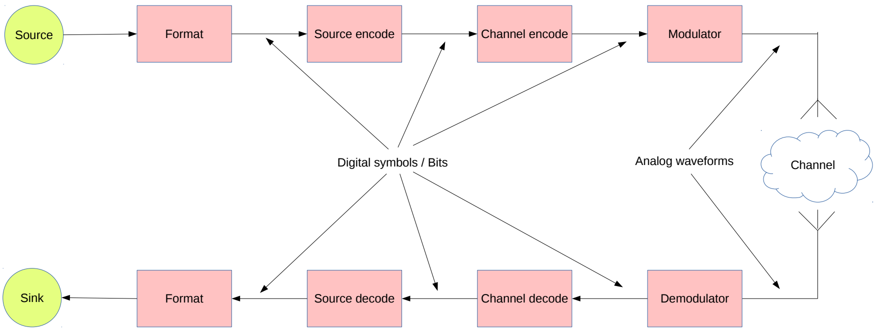
Transmitting Digital Data Using Analog Signals (2)

- In other words: the physical layer transports bits by mapping them to and from analog signals suitable for the given channel
- Existing physical layer standards deal with all things related to this task, including specification of cable types and connectors, electrical specifications (voltages, currents, resistance etc), frequency bands for wireless, shapes of transmitted waveforms etc

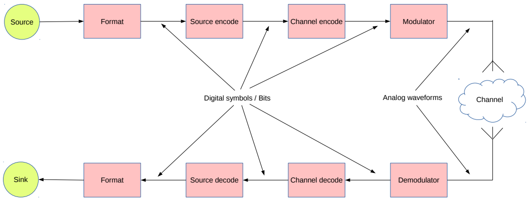
Outline

1. Fundamentals
   - Digital and Analog Transmission
   - Structure of Communication Systems
2. Baseband Transmission
3. Passband Transmission
4. Synchronization

Structure of Communication Systems (1)

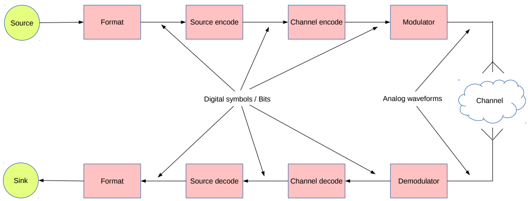


Structure of Communication Systems (2)



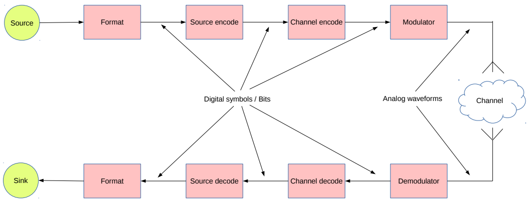
- In the OSI and TCP/IP reference models the lowest layer is the PHY, which maps streams of bits to physical signals (or waveforms) for transmission and back
- Communications engineers have developed own view on structure of communication systems or networks, which centers around the signal / data streams that are processed and the various transformations between these
- Figure shows this view – some authors (e.g. [24]) add further steps
- What we think of as a protocol stack is hidden within the "Channel encode" and "Channel decode" steps, but in a communications engineers mind "Channel encode" is mainly about error-correction coding

Formatting



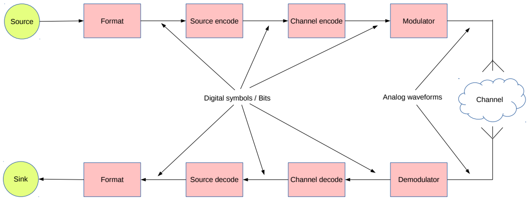
- The formatting step transforms a source signal into digital data
- For example, a video camera captures light and directs it onto an image sensor which generates a raw digital image. The raw image is represented as a finite number of pixels, arranged in a two-dimensional array, such that each pixel consists of a finite number of bits
- Another example is the sampling and subsequent A/D conversion of speech or sound signals. In older ISDN-based telephony speech signals are sampled at 8 kHz, and the amplitude of each sample is represented as an eight-bit number.

Source Coding



- **Source coding** (or **compression**) is the encoding of digital data (text, digital video / voice / audio, web pages, software) with the goal of reducing size
- With **lossless coding** (or "redundancy reduction") the digital data is encoded such that it can be perfectly restored. This is the preferred method for text, web pages, or software.
- With **lossy coding** (or "relevancy reduction") the decoded data may differ from the original data, but the aim is that humans do not perceive a large difference. This method allows for better compression and is applied to data like video, audio or speech, where human perception has some tolerance to imperfections.

Source Coding (2)

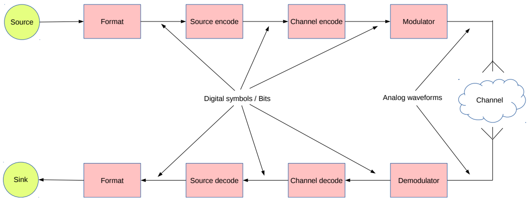


- Example: streaming of HDTV ("High Definition TeleVision"), which offers resolutions of 1280×720 or 1920×1080 pixels in the progressive scan method (no interlacing) at rates of either 24 or 30 frames per second (fps)
- Assuming 24 bit color information (RGB) per pixel, uncompressed rates are:

	24 fps	**30 fps**
1280×720	506 Mb/s	632 Mb/s
1920×1080	1,139 Mb/s	1,423 Mb/s

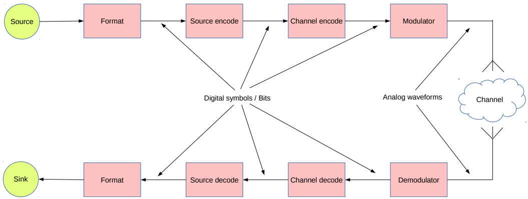
- Compression is needed to reduce this to manageable rates (video streaming).
- Source coding is a wide and established field, some references: [12], [2], [17], [18], [30], [23], [28], [11], [10], [29], [31], [19], [5], [14], [26], [8], [15]

Channel Coding



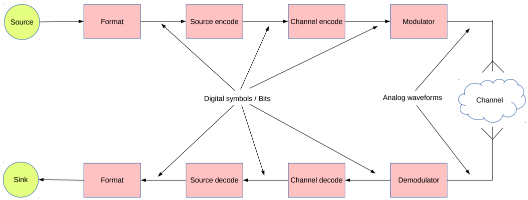
- Channel encoding takes a stream of digital symbols / bits and adds **redundant symbols/bits** to them to counter the occurence of transmission errors (or the loss of individual symbols or even whole packets) in the channel
- Adding redundancy can improve the receivers chance to guess the correct data even when the channel adds distortions
- Main goal of channel encoding is to reduce the **bit error probability**
- Coding theory has been created together with information theory in Shannons 1948 landmark paper [22], see also [3], [21], [6], [16], [7], [4]

Channel Coding (2)



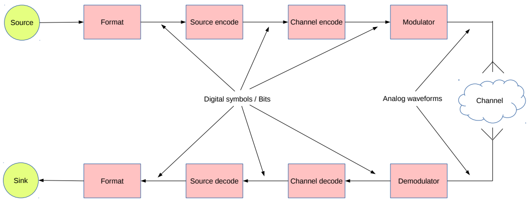
- In the source coding step we compress data by removing redundancy, whereas in the channel encoding step we add redundancy – this sounds contradictory and it is natural to ask whether something can be gained by integrating these
- Maintaining a clear separation means that each of these functions can be engineered separately. Any advances made in video coding can be made available in any system, irrespective of channel coding method
- Source-channel separation theorems: for point-to-point channels (with one transmitter and one receiver) it is optimal (e.g. with respect to quality of the signal) to separate these two functions. This is not generally true in other scenarios, for example broadcast systems

Modulator and Channel



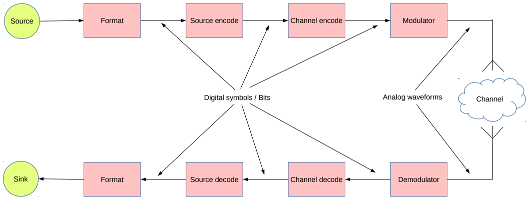
- Modulator takes a sequence of digital symbols and maps these to a physical signal or waveform of a certain duration (the **symbol time**) suitable for transmission over the given channel, and the waveforms of subsequent digital symbols are transmitted back to back
- When we modulate bits, the zero bit will be mapped to some waveform $s_0(t)$ and the one bit will be mapped to another waveform $s_1(t)$
- Example: modulators could generate radio waves for wireless channels, electrical signals for copper cables, acoustic waveforms for underwater channels
- The channel guides the signal / waveform from the transmitter to the receiver. Along the way it introduces attenuation and various distortions

Demodulator and Channel Decoder



- The demodulator processes incoming (and distorted) waveforms and produces a **guess** of what could have been the transmitted waveform (and thus the transmitted symbol), e.g. either $s_0(t)$ or $s_1(t)$ in case of bits. This is possible since there is only a finite number of transmitted waveforms.
- A key performance criterion is the probability that this guess is wrong, which is called the **probability of error** or the **symbol/bit error rate**
- Channel decoder takes output of the demodulator (which is a sequence of digital symbols/bits, possibly different from the symbols/bits at the input of the encoder) and tries to use the redundancy introduced by channel encoder to correct some or all of the errors.

Source Decoder and Formatter



- The source decoder interprets the output of the channel decoder as compressed source data (e.g. compressed video) and uncompresses it, for example producing raw digital images for display
- The final formatting step then converts the raw / uncompressed data for output to the user, e.g. D/A conversion, playout on a speaker

Outline

Baseband Transmission

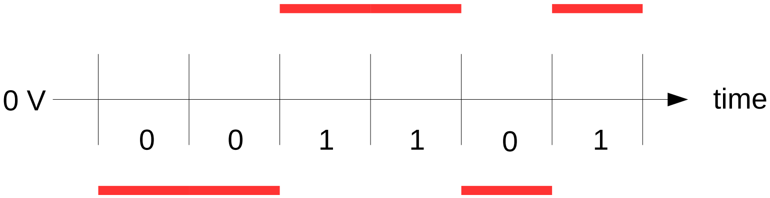
- We discuss simple baseband transmission schemes
- We will also investigate their behaviour over realistic channels and see how propagation phenomena impact the received signal
- Caveat: All of the following is simplified for illustration purposes

Outline

NRZ Encoding

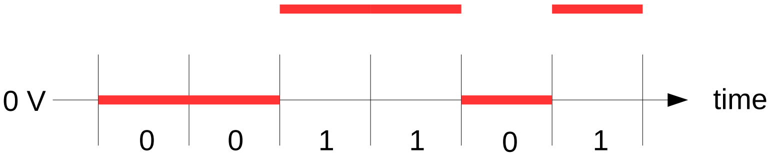
- NRZ methods are concerned with transmission of bits, for example over an electrical cable
- To each bit value (0 or 1) a voltage level is associated that is transmitted for a fixed amount of time called the **symbol time**
- The inverse of the symbol time is the **symbol rate** or **baud rate**
- As we transmit only one bit of information during one symbol time, the symbol rate actually equals the **bit rate**

Bipolar NRZ Encoding



- In **bipolar NRZ** the one bit is mapped to some positive voltage V and the zero bit is mapped to $-V$
- As an example, the figure shows the bipolar-NRZ encoding of the bit sequence 001101

Unipolar NRZ Encoding



- In **unipolar NRZ** (or on-off-keying) the one bit is mapped to some positive voltage V (e.g. 1 V), and the zero bit is mapped to a zero voltage
- As an example the figure shows the unipolar-NRZ encoding of the bit sequence 001101

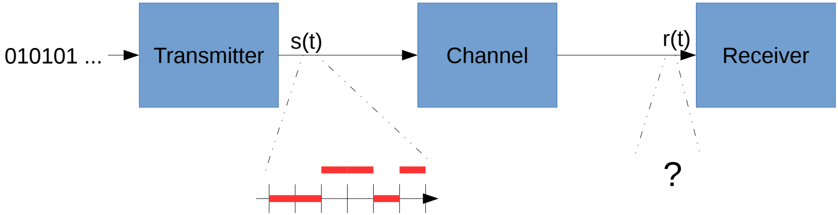
M-ary Signaling

- There is no reason to restrict baseband transmission to NRZ and just using two levels
- In M-ary transmission groups of k bits are mapped to $M = 2^k$ different voltage levels
- Due to transmit power restrictions these voltage levels will generally be closer together than with just two levels, which makes it harder for the receiver to distinguish levels and will increase the symbol error rate

Outline

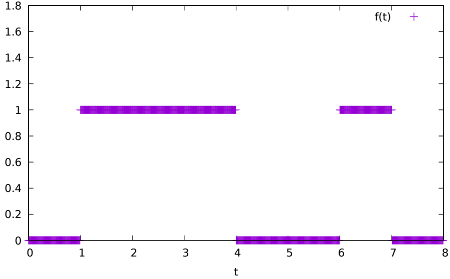
1. Fundamentals
2. Baseband Transmission
   - Non-Return-to-Zero (NRZ) Encoding
   - Signal Impairments
3. Passband Transmission
4. Synchronization

Signal Impairments



- We will use unipolar NRZ as a simple transmission method for a sequence of data bits
- We consider a transmitter sending a sequence of bits encoded according to unipolar NRZ (with output waveform $s(t)$), a channel which acts on the waveform by "adding" various impairments, and a receiver getting impaired waveform $r(t)$ and trying to make sense of it
- The list of impairments is incomplete and reflects wired channels, wireless channels have a wider range of behaviour and are much more complicated

Signal Impairments (2)



- We will look at one particular sequence of eight bits (01110010) defined over the time interval $[0, 8]$, see the figure
- We denote the actually transmitted signal after encoding with unipolar NRZ and taken at the output of the transmitter as $s(t)$, the received signal at the input of the receiver is $r(t)$

Signal Impairments – Attenuation (1)

- For various physical reasons, electrical signals in wires or electromagnetic signals in wireless systems undergo **attenuation** in the transmission medium
- Examples: light is attenuated in water or dark glass, acoustic signals and wireless signals are attenuated over distance (with further attenuation from rain, fog, etc), electrical signals are attenuated in cables
- Attenuation refers to loss of signal power and in a communications context it most often refers to the loss of signal power for **varying distance** (geographical distance in wireless systems, cable length in wired systems) between transmitter and receiver
- Attenuation is measured as the ratio of the signal power P_{tx} observed at the output of the transmitter and the signal power P_{rx} observed at the input of the receiver:

$$\eta = \frac{P_{tx}}{P_{rx}}$$

so that:

$$P_{rx} = \frac{P_{tx}}{\eta}$$

Signal Impairments – Attenuation (2)

- In many applications these attenuation factors η assume ridiculously large values, so it is common to use a logarithmic scale instead of natural scale
- Communications engineers have adopted the convention of **decibels** (dB), and the conversion of a natural value η to its decibel value η_{dB} is

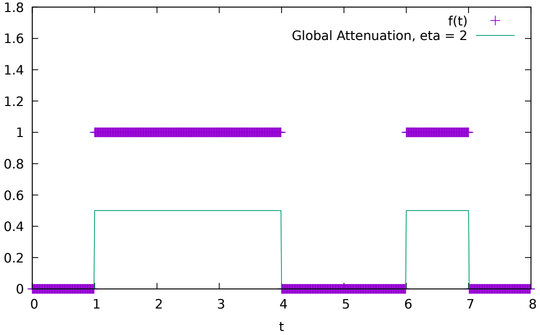
  $$\eta_{dB} = 10 \cdot \log_{10} \eta$$

  where $\log_{10}$ denotes the base-10 logarithm
- $\eta = 10$: we get $\eta_{dB} = 10$
- $\eta = 100$: we get $\eta_{dB} = 20$
- $\eta = 1000$: we get $\eta_{dB} = 30$
- $\eta = 10,000$: we get $\eta_{dB} = 40$
- An example involving "ridiculously large" numbers: A certain vendor of wireless circuitry promises that its receiver should be able to successfully decode signals received with a strength of -95 dB of a signal transmitted with one mW. Writing this number out in the “natural” domain is not very readable.

Signal Impairments – Attenuation (3)

- A signal can be decomposed (through a process called Fourier analysis) into a weighted sum of sine- and cosine-waves of different frequencies
- A channel can act on different frequencies in different ways, e.g. through frequency-dependent attenuation
- The changes introduced by the channel will then also change what the receiver receives
- Important question: does a medium attenuate all frequencies in the same way or are there some frequencies with higher attenuation?
- Example: coloured glass only lets pass light of a certain colour and blocks out (attenuates strongly) all other frequencies

Signal Impairments – Attenuation (4)

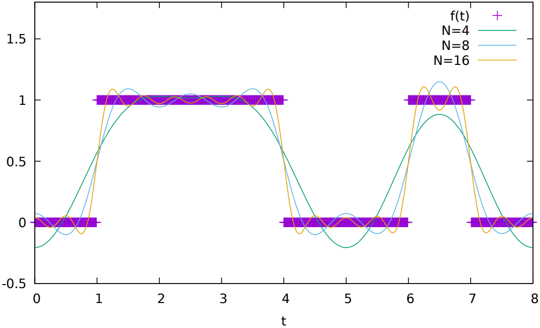


- When the channel attenuates all frequencies by the same factor η, then attenuation translates into a simple division of the signal by the attenuation factor, i.e.

$$r(t) = \frac{s(t)}{\eta}$$

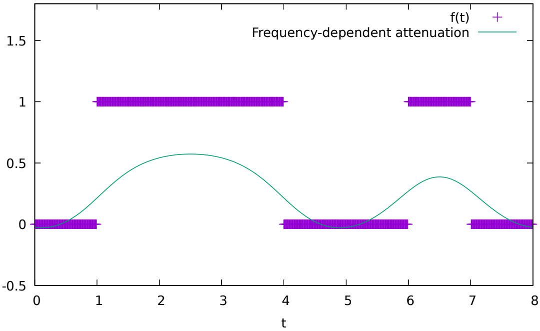
- Figure shows effect of this for $\eta = 2$

Signal Impairments – Attenuation (5)



- Now we assume that there is a cutoff frequency below which all frequencies pass without being attenuated and above which all frequencies are attenuated completely
- This is a mathematical idealization of the realistic observation that communications channels tend to pass some frequencies and block others completely – for example, a copper cable will block visible light
- Figure shows the resulting signals for different values of the cutoff frequencies

Signal Impairments – Attenuation (6)



- The model with the cutoff frequency is more realistic, but this sharp cutoff behaviour cannot be physically realized
- A more realistic model is a channel attenuating different frequencies differently
- Figure shows output of a channel which does not attenuate first three harmonics, and which attenuates further harmonics by factors of 2, 4, 8, 16 etc

Signal Impairments – Thermal Noise

- Even in the absence of a transmitted signal a receiver can see effects of random pertubations, which can either be created in the channel itself (for example: infrared channels subjected to sunlight) or in the receiver circuitry (temperature-dependent movements of electrons generate random signal fluctuations)
- These effects can have a wide range of sources and are collectively referred to as **noise**
- A standard model for thermal noise is **additive white Gaussian noise** (AWGN), in which the noise signal $n(t)$ at time t is modeled as a zero-mean normally distributed (or Gaussian) random variable with variance σ^2, which has the probability density function

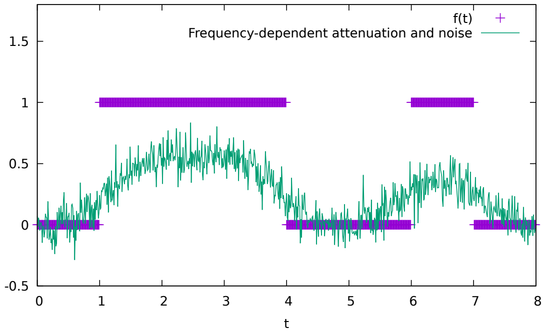
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right] \tag{1}$$

- With thermal noise the receiver will observe the signal

$$r(t) + n(t)$$

UC UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha CHRISTCHURCH NEW ZEALAND

Signal Impairments – Thermal Noise (2)



- Figure shows combined effects of frequency-dependent attenuation and noise
- Receiver has to guess transmitted data from this received waveform!

Outline

Passband Transmission

- A fundamental problem with baseband transmission is that the medium can not be shared
- For example, if you use a cable, then only one baseband transmission can be active at any time, otherwise we have a **collision**
- With passband transmission the digital data is transmitted by modulating a **sinusoidal carrier** of a certain **center frequency**
- The resulting signal occupies a certain bandwidth around the center frequency
- By carefully choosing center frequencies of different signals with sufficient separation, **it becomes possible to transmit several data signals in parallel on the same channel** without collision / overlap of frequencies used
- This is also known as **frequency-division multiplexing**
- Example: FM radio stations sharing the radio band between 87.5 and 108 MHz

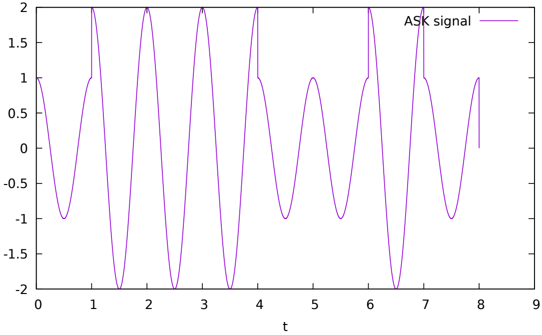
Passband Transmission (2)

- In **digital bandpass modulation** digital information is modulated onto a sinusoidal carrier of duration T, the **symbol duration**
- General form of the transmitted signal:

$$s(t) = A(t) \cdot \cos\left[(f_c + f(t)) \cdot t + \phi(t)\right] \qquad , \qquad t \in [0, T]$$

  where $A(t)$ is the (possibly time-dependent) amplitude, $f(t)$ is the frequency offset, $\phi(t)$ is the phase, and f_c is the center frequency
- There are three "pure" forms of digital modulation
- **Amplitude modulation** or amplitude shift keying (ASK): only $A(t)$ is varied according to the digital data, $f(t)$ and $\phi(t)$ are held constant
- **Frequency modulation** or frequency shift keying (FSK): only $f(t)$ is varied, $A(t)$ and $\phi(t)$ are held constant.
- **Phase modulation** or phase shift keying (PSK): only $\phi(t)$ is varied, $A(t)$ and $f(t)$ are held constant

Amplitude Shift Keying



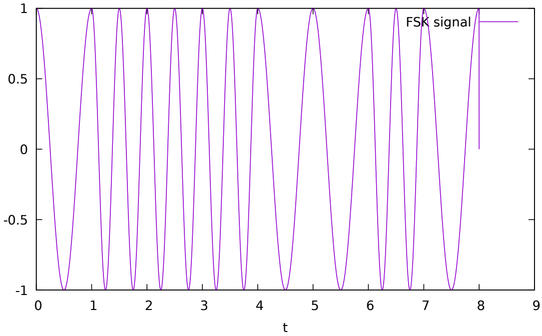
- General expression of ASK modulation:

$$s_i(t) = A_i \cos(f_c \cdot t) \qquad (0 \leq t \leq T)$$

where f_c is the center frequency, $s_i(t)$ is the waveform for the i-th member of the alphabet ($i \in \{0, \ldots, M-1\}$ for M-ary modulation) and A_i is the corresponding amplitude level

- Figure shows ASK-modulated signal for bit string 01110010, $f_c = 2\pi$ and with two amplitudes: $A_0 = 1$ and $A_1 = 2$

Frequency Shift Keying

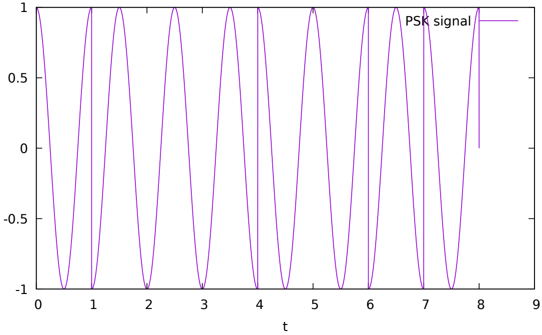


- General expression of FSK modulation:

$$s_i(t) = A\cos((f_c + f_i) \cdot t) \qquad (0 \leq t \leq T)$$

  where f_c is the center frequency, $s_i(t)$ is the waveform for the i-th member of the alphabet ($i \in \{0, \ldots, M-1\}$ for M-ary modulation) and f_i is the corresponding frequency offset
- Figure shows FSK-modulated signal for bit string 01110010, $f_c = 2\pi$ and with two frequency offsets: $f_0 = 0$ and $f_1 = 2\pi$

Phase Shift Keying

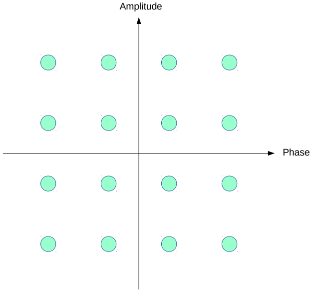


- General expression of PSK modulation:

$$s_i(t) = A\cos(f_c \cdot t + \phi_i) \qquad (0 \le t \le T)$$

  where f_c is the center frequency, $s_i(t)$ is the waveform for the i-th member of the alphabet ($i \in \{0, \ldots, M-1\}$ for M-ary modulation) and ϕ_i is the corresponding phase shift
- Figure shows PSK-modulated signal for bit string 01110010, $f_c = 2\pi$ and with two phase shifts: $\phi_0 = 0$ and $\phi_1 = \pi$

Quadrature Amplitude Modulation (QAM)



- QAM can be regarded as a combination of ASK and PSK, it fixes amplitudes and phase values
- QAM is usually applied with 16, 64 or 256 different constellation points, so that groups of 4, 6, or 8 bits are maped to one QAM symbol
- Figure shows a constellation diagram for 16-QAM, one axis is amplitude, other axis is phase – to each group of four bits one such point is allocated

Outline

Synchronization

- Sender and receiver need to agree on a range of things:
  - How long a unit of time is and how long the symbol period is
  - The exact center frequency
  - Where packets start and end
- Sender and receiver have usually no access to a common time or frequency reference
- Instead, both sender and receiver have their own local time and frequency references, provided through **crystal oscillators**
- But: oscillator frequency depends on time (oscillator aging) and environment (temperature, pressure, supply voltage, . . .) and therefore is somewhat different between transmitter and receiver
- Receiver must **learn** time/frequency reference of transmitter, it:
  - needs to **acquire synchronization** initially, and
  - needs to **track/maintain it** during packet reception

Frequency and Carrier Synchronization

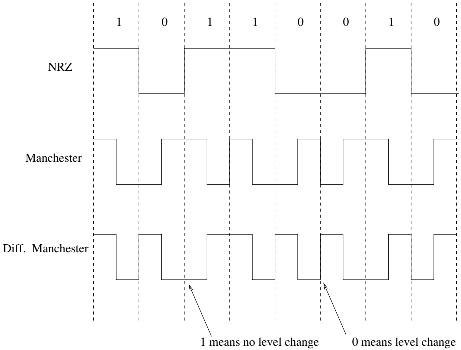
- In passband transmission receiver need to agree on the center frequency
- Potential reasons for frequency mismatch:
  - Oscillator mismatch
  - Doppler shift (through mobility)
- With PSK and QAM modulation, we also need to track the phase of a signal
- The combination of frequency- and phase synchronization is also known as **carrier synchronization**
- The receiver has to do this despite channel distortions
- We will not discuss this any further

UC UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha CHRISTCHURCH NEW ZEALAND

Symbol-/Bit Synchronization

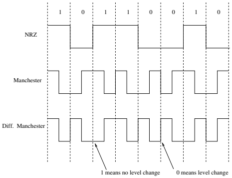
- The transmitter transmits (binary or M-ary) symbols at its own pace, using its local clock to generate symbol duration
- Receiver needs to extract information about symbol duration from the received signal so that it knows where symbols start and end
- For longer streams of information bits the receiver clock must be synchronized continuously
- In encoding schemes like NRZ or On-Off-Keying long runs of the same bit value would lead to long durations with the same voltage / amplitude level, which can cause the receiver to lose synchronization
- Therefore, we need to ensure that the signal level changes sufficiently often

Bit Synchronization – Manchester Encoding



- The Manchester encoding (shown in the second row of Figure and used in Ethernet) ensures that there is at least one signal level change per bit
- Every logical "1" is represented by a signal change from one to zero in the middle of a symbol duration, whereas a logical "0" shows the opposite signal change

Bit Synchronization – Manchester Encoding (2)

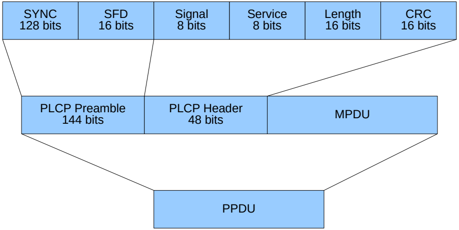


- The internal clock of the receiver samples the incoming signal with a much higher frequency, for instance 16 times per bit
- For a logical "0" bit that is arriving exactly in time, the receiver gets a sample pattern of 0000000011111111
- If the transition between the "0" and "1" samples is not exactly in the middle of the bit but rather left or right of it, the local clock has to be re-adjusted to run faster or slower, respectively
- For initial synchronization, in Ethernet a 64 bit long *preamble* is transmitted ahead of each frame, consisting of alternating "0" and "1" bits

Frame Synchronization

- The receiver does not only need to know where individual bits / symbols start and end, it also needs to know where entire packets start and ends
- On the physical and the link layer, packets are also often referred to as **frames**
- There are several methods to mark the start and end of frames, real world technologies often use a combination

Frame Synchronization – Example Frame Structure



(from IEEE 802.11, [13, Fig. 15-1])

- This is the physical layer frame structure used in wireless LANs
- `SYNC` field is preamble
- Start-frame-delimiter (`SFD`) indicates start of PHY frame
- **Question**: Why is `SFD` needed?

Frame Synchronization Methods

- **Time gap**: between frames the medium is kept idle, i.e. at zero voltage / amplitude
- **Code violations**: introduce deliberate violations in encoding methods, e.g. have symbol durations *without* signal change in Manchester
- **Length field**: use a dedicated field to indicate the length of a frame and thus specify where it ends – but additional methods are needed to deal with bit errors in this field

Frame Synchronization Methods – Start/End Flags

- Some protocols use special *flags* to indicate the frame boundaries
- In HDLC a sequence of 01111110 marks the beginning and the end of a frame
- Since data payload is arbitrary, it is possible that the flag is contained in it
- To avoid misinterpretation of a piece of payload data as the end of a frame, sender has to make sure that it only transmits the flag pattern if it is meant as a flag; any flag-like data has to be altered in a consistent way to allow the receiver to recover the original payload
- **Bit stuffing approach**: sender inserts a zero bit after each sequence of 5 consecutive "1" bits. The receiver checks if the sixth bit that follows five ones is a zero or one bit. If it detects a zero, it is removed from the sequence of bits. If it detects a "1", it can be sure that this is a frame boundary
- Unfortunately, it might happen that a transmission error modifies the data sequence 01111100 into 01111110 and thus creates a flag inadvertently.
- Therefore, additional mechanisms like time gaps are needed to remove the following bits and detect the actual end of the frame.

UC UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

[1] Sergio Benedetto and Ezio Biglieri.
Principles of Digital Transmission – With Wireless Applications.
Kluwer Academic / Plenum Publishers, New York, 1999.

[2] Toby Berger and Jerry D. Gibson.
Lossy source coding.
IEEE Transactions on Information Theory, 44(6):2693–2723, October 1998.

[3] Elwyn R. Berlekamp.
Algebraic Coding Theory.
McGraw-Hill, New York, 1968.

[4] A. R. Calderbank.
The art of signaling: Fifty years of coding theory.
IEEE Transactions on Information Theory, 44(6):2561–2595, October 1998.

[5] Shih-Fu Chang and Anthony Vetro.
Video Adaptation: Concepts, Technologies. and Open Issues.
Proceedings of the IEEE, 93(1):148–158, January 2005.

[6] Daniel J. Costello and G. David Forney.
Channel Coding: The Road to Channel Capacity.
Proceedings of the IEEE, 95(6):1150–1177, June 2007.

[7] Daniel J. Costello, Joachim Hagenauer, Hideki Imai, and Stephen B. Wicker.
Applications of error-control coding.
IEEE Transactions on Information Theory, 44(6):2531–2560, October 1998.

[8] Minoru Etoh and Takeshi Yoshimura.
Advances in Wireless Video Delivery.
Proceedings of the IEEE, 93(1):111–122, January 2005.

[9] Robert G. Gallager.
Principles of Digital Communication.
Cambridge University Press, Cambridge, UK, 2008.

[10] Aditya Ganjam and Hui Zhang.
Internet Multicast Video Delivery.
Proceedings of the IEEE, 93(1):159–170, January 2005.

[11] Bernd Girod, Anne Margot Aaron, Shantanu Rane, and David Rebollo-Monedero.
Distributed Video Coding.
Proceedings of the IEEE, 93(1):71–83, January 2005.

[12] Jenq-Neng Hwang.
Multimedia Networking – From Theory to Practice.
Cambridge University Press, Cambridge, UK, 2009.

[13] IEEE Computer Society, sponsored by the LAN/MAN Standards Committee.
IEEE Standard for Information technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.

[14] Aggelos K. Katsaggelos, Yiftach Eisenberg, Fan Zhai, Randall Berry, and Thrasyvoulos N. Pappas.

Advances in Efficient Resource Allocation for Packet-Based Real-Time Video Transmission.
Proceedings of the IEEE, 93(1):135–147, January 2005.

[15] Daniel T. Lee.
JPEG 2000: Retrospective and New Developments.
Proceedings of the IEEE, 93(1):32–41, January 2005.

[16] Shu Lin and Daniel J. Costello.
Error Control Coding.
Prentice-Hall, Englewood Cliffs, New Jersey, second edition, 2004.

[17] Peter Noll.
Wideband speech and audio coding.
IEEE Communications Magazine, 31(11), November 1993.

[18] Peter Noll.
Digital Audio Coding for Visual Communications.
Proceedings of the IEEE, 83(6):925–943, June 1995.

[19] Jens-Rainer Ohm.
Advances in Scalable Video Coding.
Proceedings of the IEEE, 93(1):42–56, January 2005.

[20] John G. Proakis.
Digital Communications.
McGraw-Hill, Boston, MA, fourth edition, 2001.
International edition.

[21] Tom Richardson and Ruediger Urbanke.

Modern Coding Theory.
Cambridge University Press, Cambridge, Massachusetts, 2008.

[22] Claude E. Shannon.
A mathematical theory of communication.
Bell Systems Technical Journal, 27:379–423, 623–656, July, October 1948.

[23] Thomas Sikora.
Trends and Perspectives in Image and Video Coding.
Proceedings of the IEEE, 93(1):6–17, January 2005.

[24] Bernard Sklar.
Digital Communications – Fundamentals and Applications.
Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[25] William Stallings.
Data and Computer Communications.
Pearson, tenth edition, 2013.

[26] Gary J. Sullivan and Thomas Wiegand.
Video Compression – From Concepts to the H.264/AVC Standard.
Proceedings of the IEEE, 93(1):18–31, January 2005.

[27] Andrew S. Tanenbaum and David J. Wetherall.
Computer Networks.
Prentice-Hall, Englewood Cliffs, New Jersey, fifth edition, 2010.

[28] David S. Taubman and Michael W. Marcellin.

JPEG2000: Standard for Interactive Imaging.
Proceedings of the IEEE, 90(8):1336–1357, August 2002.

[29] Yao Wang, Amy R. Reibman, and Shunan Lin.
Multiple Description Coding for Video Delivery.
Proceedings of the IEEE, 93(1):57–70, January 2005.

[30] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra.
Overview of the h.264/avc video coding standard.
IEEE Transactions on Circuits and Systems for Video Technology, 13(7):560–576, July 2003.

[31] Qian Zhang, Wenwu Zhu, and Ya-Qin Zhang.
End-to-End QoS for Video Delivery Over Wireless Internet.
Proceedings of the IEEE, 93(1):123–133, January 2005.