Communication engineering comprehensive exam

Tamas Rudner
IBYG8Z

November 2016

Contents

1 Topics

1.1 Information and Coding Theory

1. Describe the discrete memoryless source model (sampling, optimal Lloyd Max quantization ... etc.)
2. Derive the Nyquist criterion for ISI free communication over band limited channels
3. Describe the memoryless channel model (AWGN channel and BSC), derive the bit error probability as a function of the signal-to-noise ratio
4. Define and describe the properties of entropy, joint entropy, conditional entropy and mutual information
5. Define the typical set of an IT source (AEP) and derive its properties
6. Define the properties of uniquely decodable codes
7. Discuss the source coding theorem
8. Describe the Shannon-Fano, Huffman, and arithmetic coding and discuss there performance
9. Describe the LZ based compression algorithm
10. Define the channel capacity and elaborate on its calculation for symmetric channels
11. Describe the channel coding theorem
12. Define and explain the relationship between the following properties and parameters of error correcting codes: minimum code distance, code-length and message-length versus performance (Singleton and Hamming bounds), general algorithmic complexity of coding with tables
13. Introduce the concept of linear block coding and explain the meaning of: systematic codes, generator matrix and parity check matrix (and their relationship), algorithmic complexity of linear block coding (including detection)
14. Give the construction of binary Hamming codes (define the corresponding matrices and the error correcting capability)
15. Describe the Reed Solomon codes (generator matrix, parity check matrix, performance)
16. Describe the steps of the PGZ algorithm for detection
17. Summarize the different description of convolution encoders (architecture, state graph and transfer function) and compare the performance with linear block coding
18. Briefly summarize the Viterbi algorithm and its complexity for decoding convolutional codes
19. Describe the CDMA/FH system
20. Describe the CDMA/DS system and the Walsh-Hadamard codes
21. Describe the CDMA/DS system with random codes

1.2 Infocommunication systems

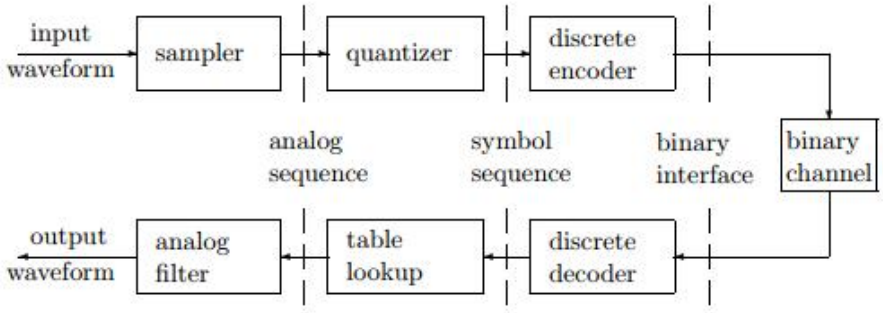
1. Main applications of twisted pair cables (telephone, data, ADSL)
2. Main characteristics of optical fiber cables. Main applications of fiber cables
3. Main radio wave propagation modes, transmission characteristics of radio connections
4. Main functions of multiplexing and switching nodes in the networks, the main features of circuit switching, packet switching and cell switching
5. Main elements and characteristics of PDH, SDH and ATM systems
6. Main elements of a GSM network (MSC, BSC, BTS, HLR, VLR LA, MS. . . ..)
7. Basic services, supplementary services, service quality requirements in different services
8. Digital modulation systems (BPSK, QPSK, QAM)
9. Typical structures and technologies in PSTN networks
10. CaTV, private (academic and university) networks
11. Main functions and characteristics of terminals, interfaces, regulation of terminals
12. Wireless LAN principles, IEEE802.11 standard
13. IPTV, MPEG, TS, multimedia program distribution
14. VoIP, SÍP, ADSL

2 Information and Coding Theory

2.1 1. topic

Description: Describe the discrete memoryless source model (sampling, optimal Lloyd Max quantization . . . etc.)

The basic communication model between two parties over a binary channel:



If the source is memoryless, it means that the consecutive messages coming from it are independent and identically distributed random variables. It is often called stochastic source as well.

Furthermore, if all of the messages are independent, then after sampling and quantisation we can encode them independently as well.

Basically in the field of infocommuncation, we have analog signals – waveforms –, such as audio or video signals. Unfortunately, we are not able to transmit analog signals over our communication channels, but discrete – binary – ones, so we need some discretisation, in order to obtain a discrete signal from an analog one. This is can be done with two easy steps. First we have to get samples from the analog signal – it is called sampling –, and after that we should assign a "new" value for a given sample in order to reduce complexity – it is called quantisation –, but we have to be careful with this, because we do not want to lose or change the information provided by the sample, so we have to keep the quantisation error low.

Let us break down sampling and quantisation a bit more.

2.1.1 Sampling

Sampling, in general, can be done on every function varying in time, space or in any given dimension, and even on multidimensional functions as well.

Sampling a function means that we take the value of it every T_s units of time (mostly seconds). T_s is often called the sampling rate, and coming from basic physics, f_s derived from T_s is called the sampling frequency, and the correspondence between T_s and f_s is the following:

$$f_s = \frac{1}{T_s}.$$

The reconstructing of the original waveform is out of the scope of this work, but it can be done with Whittaker-Shannon interpolation formula (given below), which is a combining algorithm, that takes Dirac-deltas as inputs, modulated by the sampled values, and adding them up to get – nearly – the original analog function.

If we want to reconstruct the original signal from the sampled one without any loss, then we can derive a condition, and if it is satisfied, then the reconstructing can be done without loss.

Let us consider that we have an $x(t)$ analog signal, and take a look at how it can be derived from its Fourier-transform:

$$x(t) = \int_{-B}^{B} x(f) e^{j2\pi f t} df,$$

where B is the bandwidth of the Fourier-transform. We can get the x_k sampled, discrete signal from the original one with $t = kT_s$ substitution:

$$x_k = x(t)\Big|_{t=kT_s} = \int_{-B}^{B} x(f) e^{j2\pi f k T_s} df.$$

Let us denote the sampled signal in the frequency domain with $x_m(f)$, given as follows:

$$x_m(f) = \sum_{k=-\infty}^{\infty} x\Big(f + \frac{k}{T_s}\Big),$$

for which $x_m(f) = x(f)$ is satisfied if $\frac{1}{T_s} \geq 2B$ and $|f| \leq B$. Thus we can take the Fourier-transform of $x_m(f)$:

$$x_m(f) = \sum_{k=-\infty}^{\infty} c_k e^{-2jk\pi f T_s},$$

where

$$c_k = \frac{1}{\frac{1}{T_s}} \int_{-\frac{1}{2T_s}}^{\frac{1}{2T_s}} x_m(f) e^{2jk\pi f T_s} df = T_s \int_{-\frac{1}{2T_s}}^{\frac{1}{2T_s}} x_m(f) e^{2jk\pi f T_s} df = T_s x_k.$$

Now, we can use that $x_m(f) = x(f)$, so $x(f) = \sum_{k=-\infty}^{\infty} T_s x_k e^{-2jk\pi f T_s}$, hence

$$\begin{aligned} x(t) &= \int_{-B}^{B} x(f) e^{j2\pi f t} df = \\ &= T_s \int_{-B}^{B} \left(\sum_{k=-\infty}^{\infty} x_k e^{-2jk\pi f T_s} \right) e^{j2\pi f t} df = \\ &= T_s \sum_{k=-\infty}^{\infty} x_k \int_{-B}^{B} e^{j2\pi f (t - kT_s} df, \end{aligned}$$

where

$$\int_{-B}^{B} e^{j2\pi f (t-kT_s} df = h(t - kT_s).$$

So the analog signal can be constructed by the following infinite summation:

$$x(t) = T_s \sum_{k=-\infty}^{\infty} x_k h(t - kT_s).$$

Assuming that we have an $x(t)$ analog signal, and if the $f_s \geq 2B$, then the original $x(t)$ can be reconstructed without loss from x_k sampled, discrete signal, with the expression above, where

$$h(t - kT_S) = \frac{\sin 2B\pi t}{2B\pi t}$$

is a low-pass filter.

This theorem is often called Nyquist-Shannon sampling theorem.

In practice, we cannot get just one value, but we get a really small interval and then we assign the mean of this interval to the actual value of the sample.

If we are doing oversampling, such that $f_s > 2B$, then we generates too many data, which needs more bandwidth in order to transmit it over a channel, but if we want to reach the theoretic minimum bandwidth, when $f_s = 2B$, then we should construct an ideal filter, which is impossible in practice.

2.1.2 Quantisation

We use quantisation in order to make the sampled signals to be easily represented on digital machines. Easier phrasing: assigning a value from a discrete set to every sampled value.

The main metric of a quantisation algorithm is the $S(Q)NR$, which stands for Signal-(Quality)-to-Noise Ratio. It is used to indicates whether a quantisations algorithm is good – high SNR –, or bad – low SNR. It takes the fraction of two power-like quantities (nominator is the signal, denominator is the noise). There are two major kinds of SNR:

1. Normal:
$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2$$
2. Decibel:
$$10lg\frac{P_{signal}}{P_{noise}} = 20lg\frac{A_{signal}}{A_{noise}}$$

where P is the mean-power, while A is the quadratic mean of the amplitude. In many cases, the SNR in decibal is more informative.

Type of quantisation algorihtms:

1. Equidistant
2. Logarithmic
3. Differential
4. Lloyd-Max

Let us see the properties of these algorithms.

Equidistant quantisation algorithm

Assuming that our signal takes values from the $[-C, C]$. Let us divide this interval, where the distance between two divider point is Δ. It means that we have $N = \frac{2C}{\Delta}$ quantisation level. Let us denote the quantised signal with $\tilde{x}$. Having said that, the error of the quantisation can be described with a random variable, denoted by ε, where $\varepsilon = x - \tilde{x} \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$. Because the quantisation error is like a noise, added to the original value, it makes SNR a very important metric in order to carry out the performance of the algorithm, which can be calculated in case of equidistant quantisation as follows:

$$P_{signal} = \frac{C^2}{2}$$
$$P_{noise} = \mathbb{E}(\varepsilon^2) = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} u^2 P(u) du = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} u^2 \frac{1}{\Delta} du = \frac{\Delta^2}{12}.$$

Using the two power of the signal and noise, we can calculate the SNR:

$$SNR = \frac{\frac{C^2}{2}}{\frac{\Delta^2}{12}} = \frac{3}{2}\frac{4C^2}{\Delta^2} = \frac{3}{2}N^2 = \frac{3}{2}2^{2n}.$$

Logarithmic quantisation algorithm If we are using equidistant quantisation algorithm then in some cases (e.g. human voice) the SNR can be varying regarding to the belonging range of the voice. Having said that, using the same equidistant quantisation the SNR of a signal in $[-C, C]$ would be $\frac{3}{2}\frac{4C^2}{\Delta^2}$, while if the signal is in $[-\frac{C}{2}, \frac{C}{2}]$, then the SNR would be $\frac{12}{2\Delta^2}\frac{C^2}{4}$, which is just the quarter of the previous one, and its a huge loss.

To resolve this issue, we have to transform the divider points of the equidistant quantisation algorithm from x_i to $y_i = l(x_i)$, where l is a function. The optimal differences between x_i and x_j can be calculated as follows:

$$\frac{d}{dx}l(x) \approx \frac{\Delta y}{\Delta x} = \frac{2C}{N}\frac{1}{\Delta x_i} \rightarrow \Delta x_i = \frac{1}{l'(x_i)}\frac{2C}{N}.$$

Finding the optimal $l(x)$ regarding to an unknown $p(x)$ distribution is quite hard. It is the field of Variational Calculus, and its out of the scope of this work. However, we can come up with a suboptimal solution for the l function, where l_{subopt} is independent of $p(x)$:

$$x^2 \sim \frac{1}{l'^2(x)} \rightarrow l'(x) \sim \frac{1}{x} \rightarrow l_{subopt} \sim ln(x).$$

For the SNR calculation, we should calculate the power of the signal and the noise:

$$P_{signal} = \mathbb{E}(x^2) = \int_{-C}^{C} x^2 p(x) dx$$

$$P_{noise} = \mathbb{E}(\varepsilon^2) = \sum_{i=1}^{N} \mathbb{E}(\varepsilon^2 | x \in \Delta x_i) p(x \in \Delta x_i) =$$

$$= \sum_{i=1}^{N} \frac{\Delta x_i}{12} p(x_i) \Delta x_i = K \int_{-C}^{C} \frac{1}{l'^2(x)} p(x) dx,$$

where K is some constant.
From now, we can calculate the SNR as follows:

$$SNR = \frac{1}{K} \frac{\int_{-C}^{C} x^2 p(x) dx}{\int_{-C}^{C} \frac{1}{l'^2(x)} p(x) dx}.$$

Differential quantisation algorithm In the case of differential quantisation, we use the fact that the current sample is derived from the previous one. It can be done via several methods, e.g. if the quantisation algorithm indicates that the signal is increased or decreased from one sample to another, then it is a differential quantisation.

The other type of differential quantisation when when assign variable length codes for the samples regarding to their possibility.

Lloyd-Max algoritmus We have seen that finding the optimal l quantisation function is quite hard. In order to find it iteratively, we can use the Llyd-Max algorithm, which approximates l_{opt} for which, the following statement is true:

$$l_{opt} : max_{l(x)} SNR-$$

The algorithm itself starts from two sets:

1. $\Delta = \Delta_1, ..., \Delta_N$, the differences between the quantisation levels,
2. $Q = q_1, ..., q_N$, divider points of the quantization algorithm.

The algorithm runs until a predefined value of the trustworthiness criterion function, which is:

$$J(\Delta, Q) := \sum_{i=1}^{N} \int_{\Delta_i} (x - q_i)^2 p(x) dx.$$

The steps of the algorithm are the followings:

1. If at a given step the optimal Q is known, then

$$\Delta_{l,opt} := x : (x - q_l)^2 < (x - q_i)^2, \forall i \neq l$$

2. If at a given step the optimal Δ is known, then

$$q_{l,opt} := \frac{\int_{\Delta_l} x p(x) dx}{\int_{\Delta_l} p(x) dx} = \mathbb{E}(x | x \in \Delta_l) \forall l.$$

This is a recursive algorithm, which goes on the path below:

$$\Delta(0), Q(0) \rightarrow \Delta_{opt}(1), Q(0) \rightarrow \Delta_{opt}(1), Q_{opt}(1) \rightarrow$$
$$\rightarrow \Delta_{opt}(2), Q_{opt}(1) \rightarrow \Delta_{opt}(2), Q_{opt}(2) \rightarrow ...$$

This algorithm is often called Voronoi-iteration or relaxation.

2.2 2. topic

Description: Derive the Nyquist criterion for ISI free communication over band limited channels

---

In communications, the Nyquist ISI criterion describes the conditions which, when satisfied by a communication channel (including responses of transmit and receive filters), result in no intersymbol interference or ISI.

When consecutive symbols are transmitted over a channel by a linear modulation (such as ASK, QAM, etc.), the impulse response (or equivalently the frequency response) of the channel causes a transmitted symbol to be spread in the time domain. This causes intersymbol interference because the previously transmitted symbols affect the currently received symbol, thus reducing tolerance for noise. The Nyquist theorem relates this time-domain condition to an equivalent frequency-domain condition.

The Nyquist criterion is closely related to the Nyquist-Shannon sampling theorem, with only a differing point of view.

2.2.1 The Nyquist criterion

If we denote the channel impulse response as $h(t)$, then the condition for an ISI-free response can be expressed as:

$$h(kT_s) = \begin{cases} 1; & \text{if } k = 0 \\ 0; & \text{if } k \neq 0 \end{cases}$$

for all integers k, and where T_s is the symbol period. The Nyquist theorem says that this is equivalent to $(\forall f)$:

$$\frac{1}{T_s} \sum_{k=-\infty}^{\infty} H\Big(f - \frac{k}{T_s}\Big) = 1,$$

where $H(f)$ is the Fourier-transform of $h(t)$. This is the Nyquiest ISI criterion.

This criterion can be intuitively understood in the following way: frequency-shifted replicas of $H(f)$ must add up to a constant value.

In practice this criterion is applied to baseband filtering by regarding the symbol sequence as weighted impulses (Dirac delta function). When the baseband filters in the communication system satisfy the Nyquist criterion, symbols can be transmitted over a channel with flat response within a limited frequency band, without ISI. Examples of such baseband filters are the raised-cosine filter, or the sinc filter as the ideal case.

2.2.2 Derivation of the criterion

To derive the criterion, we first express the received signal in terms of the transmitted symbol and the channel response. Let the function $h(t)$ be the channel impulse response, x_k the symbols to be sent, with a symbol period of T_s; the received signal $y(t)$ will be in the form (where noise has been ignored for simplicity):

$$y(t) = \sum_{k=-\infty}^{\infty} x_k h(t - kT_s).$$

Sampling this signal at intervals of T_s, we can express $y(t)$ as a discrete-time equation:

$$y_l = y(lT_s) = \sum_{k=-\infty}^{\infty} x_k h_{l-k}.$$

If we write the h_0 term of the sum separately, we can express this as:

$$y_l = y(lT_s) = x_l h_0 + \sum_{k \neq l} x_k h_{l-k},$$

and from this we can conclude that if a response h_n satisfies

$$h(k) = \begin{cases} 1; & \text{if } k = 0 \\ 0; & \text{if } k \neq 0 \end{cases}$$

only one transmitted symbol has an effect on the received y_l at sampling instants, thus removing any ISI. This is the time-domain condition for an ISI-free channel. Now we find a frequency-domain equivalent for it. We start by expressing this condition in continuous time:

$$h(kT_s) = \begin{cases} 1; & \text{if } k = 0 \\ 0; & \text{if } k \neq 0 \end{cases}$$

for all integer k. We multiply such a $h(t)$ by a sum of Dirac delta function (impulses) $\delta(t)$ separated by intervals T_s This is equivalent of sampling the response as above but using a continuous time expression. The right side of the condition can then be expressed as one impulse in the origin:

$$\delta(t) = h(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s).$$

Fourier transforming both side of the equation we obtain

$$H(f) * \frac{1}{T_s} \sum_{k=-\infty}^{\infty} \delta\Big(f - \frac{k}{T_s}\Big) = 1,$$

and hence, using the properties of convolution, we get:

$$\frac{1}{T_s} \sum_{k=-\infty}^{\infty} H\Big(f - \frac{k}{T_s}\Big) = 1.$$

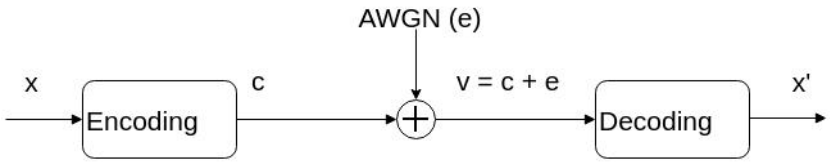
This is the Nyquist ISI criterion and, if a channel response satisfies it, then there is no ISI between the different samples.

2.3 3. topic

Description: Describe the memoryless channel model (AWGN channel and BSC), derive the bit error probability as a function of the signal-to-noise ratio

---

The basic BSC (Binary Symmetric Channel) model look like the following:



The $\oplus$ sign stands for the channel itself, because transmitting messages (denoted by c) over BSC can be modelled with adding additive, Gaussian white noise (AWGN) to the c message which was obtained by doing some encoding algorithm on the x message vector.

The memoryless channel model means that the components of the e vectors, which are added to the message each time we transmit message from one party to another, are independent and identically distributed random variables with Gaussian white noise, so $e_i \sim \mathcal{N}(0, \sigma_i^2)$.

The channel has p probability to produce a bit error, so for every bit of the c transmitted message, we have p probability that it changed from 1 to 0 or vice versa, and $1 - p$ probability to keep the value of the bit.

This model is really useful, hence many problems in communication theory can be reduced to a BSC.

2.3.1 Derivation of the BER as a function of the SNR

The $e(t)$ noise is an additive, Gaussian white noise, with zero expected value and some σ^2 variance. Let us denote the digitization threshold with v_t. It means that if the received signal is greater than v_t then we will assign 1 to the signal, otherwise we assign 0 to the signal. Let us assume that $v_t = 0.5$. Further on, let v is the received symbol, while c is the transmitted symbol. The probability of the bit error can be expressed as:

$$P_b = P(c=1)P(v=0|c=1) + P(c=0)P(v=1|c=0) = 0.5\Phi\left(\frac{-0.5}{\sigma}\right) + 0.5\Phi\left(\frac{-0.5}{\sigma}\right) = \Phi\left(\frac{-0.5}{\sigma}\right).$$

Let us denote the power of the noise-free signal with P_{signal}, then we can write the SNR (Signal-to-Noise ratio) in the following form:

$$SNR(db) = 10 \lg \left(\frac{P_{signal}}{\sigma^2}\right).$$

Given an SNR, we can use this formula to calculate the variance of the noise

$$\sigma = \sqrt{\frac{P_{signal}}{10^{\frac{SNR}{10}}}}.$$

Using the last result, we can calculate the bit error probability as a function of SNR

$$P_b = \Phi\left(\frac{-0.5 10^{\frac{SNR}{10}}}{P_{signal}}\right),$$

which is the expected value of the bit error rate.

2.4 4. topic

Description: Define and describe the properties of entropy, joint entropy, conditional entropy and mutual information

---

The following entropy definitions have been made by Shannon.
The entropy of the X discrete random variable can be defined with the following summation:

$$H(X) = \mathbb{E}(-\log_2 p(X)) = -\sum_{k=1}^{n} p(X_i) \log_2 p(X_i).$$

Please note, that the entropy is not dependent on the value of X, but the probability distribution of X.
The properties of entropy are the followings:

1. $0 \leq H(X) \leq \log_2(n)$, and $H(X) = 0$ if and only if $P(X) = 1$, and $H(X) = \log_2(n)$ if X is equally distributed random variable
2. For any given X random variable, and $g(X)$ function, $H\big(g(X)\big) \leq H(X)$, and the sufficient and necessary condition for equality is the invertibility of g.

To understand it a bit more, we can try to define an information function, I, in terms of an event i with probability p_i. I indicates how much information could be acquired due to the observation of event i. The properties of the I function:

1. $I(p)$ is anti-monotonic, so if the probability of an event increases or decreases then it produce a decrease and increase in its information, respectively
2. $I(0)$ is undefined
3. $I(p) \geq 0$ – information is a non-negative quantity
4. $I(1) = 0$ – events that always occur do not communicate information
5. $I(p_1, p_2) = I(p_1) + I(p_2)$ if p_1 and p_2 are independent

The proper choice of function to quantify information, preserving these properties (especially the last one) is logarithmic, i.e.,

$$I(p) = log_2(\frac{1}{p}).$$

The joint entropy of X and Y discrete random variables can be defined as follows:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) log_2\big(P(x_i, y_j)\big),$$

where $P(x_i, y_j)$ is the joint probability of the X being x_i and Y being y_i.
Properties of the joint entropy are the following:

1. $H(X,Y) \geq \max\big(H(X), H(Y)\big)$
2. $H(X_1, ..., X_n) \geq \max\big(H(X_1), ..., (X_n)\big)$
3. $H(X,Y) \leq H(X) + H(Y)$
4. $H(X_1, ..., X_n) \leq H(X_1) + ... + H(X_n)$

The conditional entropy of X and Y discrete random variables can be defined as follows:

$$H(X|Y) = H(Y,X) - H(Y).$$

This statement needs a proof, which is quite simple:

$$\begin{aligned}
H(Y|X) &= \sum_{x\in\mathcal{X}} p(x) H(Y|X=x) = \\
&= -\sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x) \log_2\big(p(y|x)\big) = \\
&= -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log_2\big(p(y|x)\big) = \\
&= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y) \log_2\big(p(y|x)\big) = \\
&= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y) \log_2\left(\frac{p(x,y)}{p(x)}\right) = \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y) \log_2\left(\frac{p(x)}{p(x,y)}\right) = \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y) \log_2\left(\frac{1}{p(x,y)}\right) - \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y) \log_2\left(\frac{1}{p(x)}\right) = \\
&= H(Y,X) - H(X).
\end{aligned}$$

The value of $H(Y|X) = 0$ if and only if the value of Y is completely determined by the value of X. Coversely, $H(Y|X) = H(Y)$ if and only if Y and X are independent random variable.

The other properties of conditional entropy:

1. $H(Y|X) \leq H(Y)$
2. $H(X,Y) = H(X|Y) + H(Y|X) + I(X;Y)$
3. $H(X,Y) = H(X) + H(Y) - I(X;Y)$
4. $I(X;Y) \leq H(X)$

where $I(X;Y)$ is the mutual information between X and Y.

Using the law of total probability, we can express the joint entropy of n discrete random variable with the sum of the conditional entropies:

$$H(X_1, ..., X_n) = \sum_{k=1}^{N} H(X_k|X_{k-1}, ..., X_1).$$

It is often called chain rule.

There is a Bayes' rule for the conditional entropy, which states

$$H(Y|X) = H(X|Y) - H(X) + H(Y).$$

Its proof is quite straightforward, because if we take both conditional entropy of X and Y

$$\begin{aligned}
H(Y|X) &= H(X,Y) - H(X) \\
H(X|Y) &= H(Y,X) - H(Y),
\end{aligned}$$

and we use the fact that the mutual information is symmetric, such that $H(X,Y) = H(Y,X)$, and we substract the two equation, we get the Bayes' rule.

The mutual information of the X and Y discrete random variables can be given as follows:

$$I(X;Y) = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x,y)\log_2\left(\frac{p(x,y)}{p(x)p(y)}\right).$$

Some basic property of mutual information:

1. $I(X;Y) = 0$ if and only if X and Y are independent random variables
2. $I(X;Y) \geq 0$
3. $I(X;Y) = I(Y;X)$

Mutual information can be equivalently expressed as:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) = \\ &= H(Y) - H(Y|X) = \\ &= H(X) + H(Y) - H(X,Y) = \\ &= H(X,Y) - H(X|Y) - H(Y|X). \end{aligned}$$

Using Jensen's inequality on the definition of mutual information we can show that $I(X;Y)$ is non-negative, consequently, $H(X) \geq H(X|Y)$. Here we give the detailed deduction of $I(X;Y) = H(Y) - H(Y|X)$:

$$\begin{aligned} I(X;Y) &= \sum_{x,y} p(x,y)\log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) = \\ &= \sum_{x,y} p(x,y)\log_2\left(\frac{p(x,y)}{p(x)}\right) - \sum_{x,y} p(x,y)\log_2\big(p(y)\big) = \\ &= \sum_{x,y} p(x)p(y|x)\log_2\big(p(y|x)\big) - \sum_{x,y} p(x,y)\log_2\big(p(y)\big) = \\ &= \sum_{x} p(x)\left(\sum_{y} p(y|x)\log_2\big(p(y|x)\big)\right) - \sum_{y}\log_2\big(p(y)\big)\left(\sum_{x} p(x,y)\right) = \\ &= -\sum_{x} p(x)H(Y|X=x) - \sum_{y}\log_2\big(p(y)\big)p(y) = \\ &= -H(Y|X) + H(Y) = \\ &= H(Y) - H(Y|X). \end{aligned}$$

The proof of the other identities above are similar.

Intuitively, if entropy $H(Y)$ is regarded as a measure of uncertainty about a random variable, then $H(Y|X)$ is a measure of what X does not say about Y.

This is "the amount of uncertainty remaining about Y after X is known", and thus the right side of the first of these equalities can be read as "the amount of uncertainty in Y, minus the amount of uncertainty in Y which remains after X is known", which is equivalent to "the amount of uncertainty in Y which is removed by knowing X". This corroborates the intuitive meaning of mutual information as the amount of information (that is, reduction in uncertainty) that knowing either variable provides about the other.

Note that in the discrete case $H(X|X) = 0$ and therefore $H(X) = I(X;X)$. Thus $I(X;X) \geq I(X;Y)$, and one can formulate the basic principle that a variable contains at least as much information about itself as any other variable can provide.

For discrete random variables X and Y, the Kullback–Leibler divergence from $P(X)$ to $P(Y)$ is defined to be

$$D_{KL}\big(P(X) \parallel P(Y)\big) = \sum_{i} P(i)\log_2\left(\frac{P(X_i)}{P(Y_i)}\right).$$

In words, it is the expectation of the logarithmic difference between the probabilities $P(X)$ and $P(Y)$, where the expectation is taken using the probabilities $P(X)$. The Kullback–Leibler divergence is defined only if $P(Y_i) = 0$ implies $PXY_i) = 0$, for all i (absolute continuity).

Mutual information can also be expressed as a Kullback–Leibler divergence of the product $p(x)p(y)$ of the marginal distributions of the two random variables X and Y, from $p(x, y)$ the random variables' joint distribution:

$$I(X;Y) = D_{KL}\big(p(x,y) \parallel p(x)p(y)\big).$$

Furthermore, let $p(x|y) = \frac{p(x,y)}{p(y)}$. Then

$$\begin{aligned} I(X;Y) &= \sum_y p(y) \sum_x p(x|y) \log_2 \left(\frac{p(x|y)}{p(x)} \right) = \\ &= \sum_y p(y) D_{KL}\big(p(x|y) \parallel p(x)\big) = \\ &= \mathbb{E}_Y \Big(D_{KL}\big(p(x|y) \parallel p(x)\big) \Big). \end{aligned}$$

Note that here the Kullback-Leibler divergence involves integration with respect to the random variable X only and the expression $D_{KL}\big(p(x|y) \parallel p(x)\big)$ is now a random variable in Y. Thus mutual information can also be understood as the expectation of the Kullback–Leibler divergence of the univariate distribution $p(x)$ of X from the conditional distribution $p(x|y)$ of X given Y: the more different the distributions $p(x|y)$ and $p(x)$ are on average, the greater the information gain.

2.5 5. topic

Description: Define the typical set of an IT source (AEP) and derive its properties

---

In information theory, the typical set is a set of sequences whose probability is close to two raised to the negative power of the entropy of their source distribution. That this set has total probability close to one is a consequence of the asymptotic equipartition property (AEP) which is a kind of law of large numbers. The notion of typicality is only concerned with the probability of a sequence and not the actual sequence itself.

This has great use in compression theory as it provides a theoretical means for compressing data, allowing us to represent any sequence X_k using $kH(X)$ bits on average, and, hence, justifying the use of entropy as a measure of information from a source.

The AEP can also be proven for a large class of stationary ergodic processes, allowing typical set to be defined in more general cases.

Given a discrete-time stationary ergodic stochastic process X on the probability space (Ω, B, p), AEP is an assertion that

$$-\frac{1}{n}\log_2\big(p(X_1, X_2, ..., X_n)\big) \to H(X),$$

if n tends to infinity, where $H(X)$ is the entropy of X.

If we have an X random variable, which is an independent and identically distributed source, then its time series $X_1, X_2, ..., X_n$ is independent and identically distributed as well, with entropy $H(X)$. Then the weak law of large numbers gives the AEP with convergence in probability,

$$\lim_{n\to\infty} P\left(\Big| -\frac{1}{n}\log_2\big(p(X_1, X-2, ..., X_n)\big) - H(X)\Big| > \varepsilon\right) = 0,$$

for all $\varepsilon > 0$, since the entropy is equal to the expectation of

$$-\frac{1}{n}\big(p(X_1, X_2, ..., X_n)\big).$$

The strong law of large numbers asserts the stronger almost sure convegence,

$$P\left(\lim_{n\to\infty} -\frac{1}{n}\log_2\big(p(X_1, X-2, ..., X_n)\big) = H(X)\right) = 1.$$

2.5.1 Definition of (weakly) typical sequences

If a sequence $x_1, ..., x_n$ is drawn from an independent and identically distributed distribution X defined over a finite alphabet $\mathcal{X}$, then the typical set, $A_\varepsilon(n) \in \mathcal{X}$ is defined as those sequences which satisfy the following inequalities:

$$2^{-n(H(X)+\varepsilon)} \leq p(x_1, x_2, ..., x_n) \leq 2^{-n(H(X)-\varepsilon)},$$

whre $H(X)$ is the entropy of the random variable X. The probability above need only be within a factor of $2^{n\varepsilon}$.

It has the following properties if n is sufficiently large, $\varepsilon > 0$ can be chosen arbitrarily small so that:

1. The probability of a sequence from X being drawn from $A_\varepsilon^{(n)}$ is greater than $1-\varepsilon$, i.e. $P\big(x^{(n)} \in A_\varepsilon^{(n)}\big) \geq 1-\varepsilon$
2. $|A_\varepsilon^{(n)}| \leq 2^{n(H(X)+\varepsilon)}$
3. $|A_\varepsilon^{(n)}| \geq (1-\varepsilon)2^{n(H(X)-\varepsilon)}$

4. Most sequences are not typical. If the distribution over $\mathcal{X}$ is not uniform, then the fraction of sequences that are typical is

$$\frac{|A_\varepsilon^{(n)}|}{|\mathcal{X}^{(n)}|} = \frac{2^{nH(X)}}{2^{n\log_2(|\mathcal{X}|)}} = 2^{-n\left(log_2|\mathcal{X}|-H(X)\right)} \to 0,$$

as n becomes very large, since $H(X) < \log_2|\mathcal{X}|$.

For a general stochastic process $X(t)$ with AEP, the (weakly) typical set can be defined similarly with $p(x_1, x_2, ..., x_n)$ replaced by $p(x_0\tau)$ (i.e. the probability of the sample limited to the time interval $[0, \tau]$), n being the degree of freedom of the process in the time interval and $H(X)$ being the entropy rate. If the process is continuous-valued, differential entropy is used instead.

Please note that, counter-intuitively, most likely sequence is often not a member of the typical set.

It is often called weak typicality or entropy typicality as well.

2.5.2 Definition of strongly typical sequences

If a sequence $x_1, ..., x_n$ is drawn from some specified joint distribution defined over a finite or an infinite alphabet $\mathcal{X}$, then the strongly typical set, $A_{\varepsilon,strong}^{(n)} \in \mathcal{X}$ is defined as the set of sequences which satisfy the following inequality:

$$\left|\frac{N(x_i)}{n} - p(x_i)\right| < \frac{\varepsilon}{||\mathcal{X}||}.$$

where $N(x_i)$ is the number of occurrences of a specific symbol in the sequence.

It can be shown that strongly typical sequences are also weakly typical (with a different constant ε), and hence the name. The two forms, however, are not equivalent. Strong typicality is often easier to work with in proving theorems for memoryless channels. However, as is apparent from the definition, this form of typicality is only defined for random variables having finite support.

2.6 6. topic

Description: Define the properties of uniquely decodable codes

---

Uniquely decodable codes have an extereme usage in the field of coding theory, because it makes the decoding easier, because for every coded symbol sequence there is just one real symbol sequence, and vice versa.

The $f : \mathcal{X} \to \mathcal{Y}^*$ code is uniquely decodable if every code sequence is coding just one message sequence, more precisely, if $u, v \in \mathcal{X}^*$, where $u = u_1u_2..u_k$ and $v = v_1v_2...v_m$, then $f(u_1)f(u_2)...f(u_k) = f(v_1)f(v_2)...f(v_m)$ if and only if $u = v$.

Please note, that it is more than invertibility, because it might happen that the code is invertible, but not uniquely decodable!

The $g : \mathcal{X} \to \mathcal{Y}^*$ is prefix code if $!\exists u \in \mathcal{X}^*$, $u = u_1u_2...u_n$, for $\forall v \in \mathcal{X}^*$, $v = v_1v_2...v_k$, where $n \geq k$, if we take the first k symbol of u ($u' = u_1u_2...u_k$), then $u' = v$ if and only if $u = u' = v$.

The set of prefix codes is a subset of uniquely decodable codes.

There is an ultimate critera which has to be satisfied by a code in order to be uniquely decodable code, which is the Kraft-McMillan inequality. It is given as follows:

$$\sum_{i=1}^{n} r^{-l_i} \leq 1, \tag{1}$$

where r is the size of the code's alphabet, and $\forall i : l_i$ is the length of the coded word of the corresponding s_i source symbols.

????

2.7 7. topic

Description: Discuss the source coding theorem

In information theory, Shannon's source coding theorem (or noiseless coding theorem) establishes the limits to possible data compression, and the operational meaning of the Shannon entropy.

The source coding theorem shows that (in the limit, as the length of a stream of independent and identically-distributed random variable (i.i.d.) data tends to infinity) it is impossible to compress the data such that the code rate (average number of bits per symbol) is less than the Shannon entropy of the source, without it being virtually certain that information will be lost. However it is possible to get the code rate arbitrarily close to the Shannon entropy, with negligible probability of loss.

The source coding theorem for symbol codes places an upper and a lower bound on the minimal possible expected length of codewords as a function of the entropy of the input word (which is viewed as a random variable) and of the size of the target alphabet.

Source coding is a mapping from (a sequence of) symbols from an information source to a sequence of alphabet symbols (usually bits) such that the source symbols can be exactly recovered from the binary bits (lossless source coding) or recovered within some distortion (lossy source coding). This is the concept behind data compression.

In information theory, the source coding theorem informally states that N independent and identically distributed random variables each with entropy $H(X)$ can be compressed into more than $NH(X)$ bits with negligible risk of information loss, as $N \to \infty$ but conversely, if they are compressed into fewer than $NH(X)$ bits it is virtually certain that information will be lost.

2.7.1 The source coding theorem for symbol codes

Let Σ_1, Σ_2 denote two finite alphabets and let Σ_1^*, Σ_2^* the set of all finite words from those alphabets (respectively).

Suppose that X is a random variable taking values from Σ_1 and f be a uniquely decodable code from Σ_1^* to Σ_2^*, where $\left|\Sigma_2\right| = a$. Let S denote the random variable given by the word length $f(X)$.

If f is optimal in the sense that is has the minimal expected word length for X, then

$$H(X) \leq \mathbb{E}S.$$

2.7.2 Proof of the theorem

First, we are going to prove the source coding theorem generally, then we will provide a proof for it with symbol codes as well.

Given X is an independent and identically distributed source, its time series $X_1, ..., X_n$ is independent and identically distributed as well with entropy $H(X)$. The theorem states that for any $\varepsilon > 0$ for any rate larger then the entropy of the source, there is a large enough N and an encoder that takes N independent and identically distributed repetition of the source, X^N and maps it to $N\big(H(X) + \varepsilon\big)$ binary bits such that the source symbols X^N are recoverable from the binary bits with probability at least $1 - \varepsilon$.

Proof of Achievability

Fix some $\varepsilon > 0$, and let

$$p(x_1, ..., x_N) = P(X_1 = x_1, ..., X_N = x_N).$$

The typical set, $A_\varepsilon^{(N)}$ is defined as follows:

$$A_\varepsilon^{(N)} = \left\{(x_1, ..., x_N) : \left| -\frac{1}{N}\log_2\big(p(x_1, ..., x_N)\big) - H(X)\right| < \varepsilon\right\}.$$

The asymptotic equipartition property (AEP) shows that for large enough N, the probability that a sequence generated by the source lies in the typical set, $A_\varepsilon^{(N)}$, as defined approaches one. In particular,

for sufficiently large N, $P\big((X_1, X_2, ..., X_N) \in A_\varepsilon^{(N)}\big)$ can be made arbitrarily close to 1, and specifically, greater than $1 - \varepsilon$.

The definition of typical sets implies that those sequences that lie in the typical set satisfy the following inequalities:

$$2^{-N(H(X)+\varepsilon)} \leq p(x_1, x_2, ..., x_N) \leq 2^{-N(H(X)-\varepsilon)}.$$

Note that:

1. The probability of sequence $(X_1, X_2, ..., X_N)$ being draw from $A_\varepsilon^{(N)}$ is greater than $1 - \varepsilon$.
2. $\left|A_\varepsilon^{(N)}\right| \leq 2^{N(H(X)+\varepsilon)}$, which follows from the left hands side (lower bound) for $p(x_1, x_2, ..., x_N)$
3. $\left|A_\varepsilon^{(N)}\right| \geq (1-\varepsilon)2^{N(H(X)-\varepsilon)}$, which follows from upper bound for $p(x_1, x_2, ..., x_N)$ and the lower bound on the total probability of the whole set $A_\varepsilon^{(N)}$.

Since $\left|A_\varepsilon^{(N)}\right| \leq 2^{N(H(X)+\varepsilon)}$, $N(H(X)+\varepsilon$ bits are enough to point to any string in this set.

The encoding algorithm: the encoder checks if the input sequence lies within the typical set; if yes, it outputs the index of the input sequence within the typical set; if not, the encoder outputs an arbitrary $N(H(X)+\varepsilon)$ digit number. As long as the input sequence lies within the typical set (with probability at least $1-\varepsilon$), the encoder doesn't make any error. So, the probability of error of the encoder is bounded above by ε.

Proof of Converse

The converse is proved by showing that any set of size smaller than $A_\varepsilon^{(N)}$ (in the sense of exponent) would cover a set of probability bounded away from 1.

Proof of the source coding theorem for symbol codes

For $1 \leq i \leq n$ let l_i denote the world length of each possible x_i. Define $q_i = \frac{2^{-l_i}}{C}$, where C is chosen so that$\sum_{i=1}^{n} q_i = 1$. Then

$$\begin{aligned}
H(X) &= -\sum_{i=1}^{n} p(x_i)\log_2\big(p(x_i)\big) \leq \\
&\leq -\sum_{i=1}^{n} p(x_i)\log_2\big(q(x_i)\big) = \\
&= -\sum_{i=1}^{n} p(x_i)\log_2\big(2^{-l_i}\big) + \sum_{i=1}^{n} p_i \log_2(C) = \\
&= -\sum_{i=1}^{n} p(x_i)\log_2\big(2^{-l_i}\big) + \log_2(C) \leq \\
&\leq -\sum_{i=1}^{n} -l_i p(x_i) = \\
&= \mathbb{E}L,
\end{aligned}$$

where the second line follows from Gibbs' inequality, and the fifth follows from the Kraft's inequality:

$$C = \sum_{i=1}^{n} 2^{-l_i} \leq 1,$$

so $\log_2 C \leq 0$.

In practice, it is impossible to get the exact value of $P(X)$, which indicates an estimation ($Q(X)$), for which the source coding theorem takes the following form:

$$H(X) + D_{KL}\big(P(X)||Q(X)\big) \leq L_Q < H(X) + D_{KL}\big(P(X)||Q(X)\big) + \frac{1}{N}.$$

2.8 8. topic

Description: Describe the Shannon-Fano, Huffman, and arithmetic coding and discuss there performance.

In the history of information and coding theory there have been several algorithms developed for constructing uniquely decodable codes. The 3 (actually 4) main codes are the following:

1. Shannon-Fano
2. Huffman
3. Shannon-Fano-Elias
4. Arithmetic

2.8.1 Shannon-Fano codes

To construct a Shannon-Fano code, we can assume the following:

$$p(x_1) \geq p(x_2) \geq ... \geq p(x_n) > 0,$$

because we can achieve this even if it is not the case because we just change the indices.

In the first step, we have to divide the whole, descending ordered set into 2 subset, for which the following satisfies:

$$j = min_j \sum_{i=1}^{j} p(x_i) - \sum_{i=j+1}^{n} p(x_i),$$

where j is the index of the last element which goes to the first subset besides of every $p(x_i)$ for $i < j$, while the others goes to the second subset.

Due to its recursive nature, the steps of the algorithm are the same, and it is goes until in every set there is just one element.

For Shannon-Feno codes, we can set the length of the encoding messages, such that

$$l_i = \lceil -\log_2 \big(p(x_i)\big) \rceil,$$

so that

$$-\log_2 \big(p(x_i)\big) \leq s_i < -\log_2 \big(p(x_i)\big) + 1$$

and so

$$2^{-l_i} \leq p(x_i)$$

and

$$\sum_{i=1}^{n} 2^{-l_i} \leq \sum_{i=1}^{n} p(x_i) = 1$$

and so by Kraft's inequality there exists a prefix-free code having those word lengths (actually it is the Shannon-Feno code which can be constructed with the algorithm above). Thus the minimal L satisfies

$$\begin{aligned}
\mathbb{E}L &= \sum_{i=1}^{n} p(x_i) l_i < \\
&< \sum_{i=1}^{n} p(x_i) \Big(-\log_2 \big(p(x_i)\big) + 1 \Big) = \\
&= \sum_{i=1}^{n} -p(x_i) \log_2 \big(p(x_i)\big) + 1 = \\
&= H(X) + 1.
\end{aligned}$$

So the performance of the Shannon-Feno codes is the following:

$$H(X) \leq L < H(X) + 1.$$

To obtain a Shannon-Feno code, we should construct the binary tree, then read every symbols code to build the look-up table.

2.8.2 Huffman codes

Unfortunately, the Shannon-Fano algorithm does not always generate an optimal code. To resolve this problem, a new algorithm has been given, which always produces an optimal tree for any given symbol weights (probabilites).

While the Shannon–Fano tree is created from the root to the leaves, the Huffman algorithm works in the opposite direction, from the leaves to the root.

The steps of the algorithm:

1. Create a leaf node for each symbol and add it to a priority queue, using its frequency of occurrence as the priority.
2. While there is more than one node in the queue:
   (a) Remove the two nodes of lowest probability or frequency from the queue
   (b) Prepend 0 and 1 respectively to any code already assigned to these nodes
   (c) Create a new internal node with these two nodes as children and with probability equal to the sum of the two nodes' probabilities
   (d) Add the new node to the queue
3. The remaining node is the root node and the tree is complete

It can reach lower value for the expected code length than the Shannon-Fano coding, however it is more complex than the Shannon-Fano, which is quite fair trade-off.

The other advantage of Huffman codes over Shannon-Fano is the optimality.

2.8.3 Shannon-Fano-Elias codes

Given a discrete random variables X of ordered values to be encoded, let $p(x_i)$ be the probability for x_i. Define a function

$$\bar{F}(x_i) = \sum_{j=1}^{i-1} p(x_j) + \frac{1}{2} p(x_i).$$

For each x_i in X, let Z be the binary expansion of $\bar{F}(x_i)$. Choose the length of the encoding of x_i, $L(x_i)$, to be the integer $\lceil \log_2 \left(\frac{1}{p(x_i)} \right) \rceil + 1$. Choose the encoding of x_i to be the first $L(x_i)$ most significant bits after the decimal point of Z.

Unfortunately the performance of the code is really poor, because the average code length is

$$\mathbb{E}L = \sum_{i=1}^{n} p(x_i) L(x_i) = \sum_{i=1}^{n} p(x_i) \left\lceil \log_2 \left(\frac{1}{p(x_i)} \right) \right\rceil + 1.$$

Thus for $H(x)$, the entropy of the random variable X:

$$H(X) + 1 \leq \mathbb{E}L < H(X) + 2.$$

Hence it is not widely used in practice.

2.8.4 Arithmetic coding

Arithmetic coding is the natural extension of the Shannon-Feno codes. The probabilities are coming from intervals, so the probabilities have to be calculated very precisely.

Apart from the other 3 algorithm, it encodes blocks instead of symbols. Every code word maps to a $[0, 1)$ interval, and it is given by as many bits as needed in order to distinguish the interval from other sub-intervals. Shorter code words get larger intervals, so they are coding more likely blocks.

In practice the resolution is getting smoother and smoother until it gets small enough.

Ideally, it is done with infinity precision arithmetic, but in practice we can use just finite precision arithmetic, of course.

The coding method is the following:

1. At the first step we choose the interval $[E_0, V_0)$ to be $[0, 1)$.
2. Every symbol is being coded in two steps:
   (a) First, we divide the actual $[E_k, V_k)$ interval into distinct sub-intervals, and each of them should stands for the possible values of the new symbol. The length of the intervals are coming from the conditional probabilities of the symbols.
   (b) Then we choose the interval which contains the value of the new symbol's probability, and after that, this will be the $[E_{k+1}, V_{k+1})$ (actual) interval.

To calculate the intervals, we should introduce the cumulative probabilities, which can be gives for the k-th symbol's i-th possible value as follows:

$$w_i^k = \begin{cases} 0; & \text{if } i = 0 \\ \sum_{j=1}^{i-1} p(x_j | x_{\alpha_1}, ..., x_{\alpha_{k-1}}); & \text{if } 1 \leq i \leq n \end{cases}$$

where $x_{\alpha_1}, ..., x_{\alpha_{k-1}}$ are the processed symbols. If the i-th possible symbol value comes from the input, and for a given $[E_{k-1}, V_{k-1})$ actual interval, then the new $[E_k, V_k)$ interval can be given as follows:

$$E_k = E_{k-1} + w_i^k (V_{k-1} - E_{k-1})$$
$$V_K = E_{k-1} + \big(w_i^k + p(x_i)\big)(V_{k-1} - E_{k-1}).$$

It can be easily seen that the last interval uniquely defines the whole nested sequence of interval, so it can be decoded easily.

Every interval can be coded with the value from the interval which can be coded by the least bits.

It can be seen that for an Y block, the value of the interval which further encodes the Y block should be coded by

$$\left\lceil \log_2 \left(\frac{1}{p(x_i)} \right) \right\rceil + 1$$

bits. It is actually the Shannon-Fano-Elias coding.

Assuming that the components of X message are independent and identically distributed, then w_i^k are independent from k, so their calculation of them does not require the conditional probabilities to be calculated with:

$$w_i^k = \begin{cases} 0; & \text{if } i = 0 \\ \sum_{j=1}^{i-1} p(x_j); & \text{if } 1 \leq i \leq n \end{cases}$$

Then using the additive property of independent random variables joint entropy, $H(x_1, x_2, ..., x_n) = nH(x_1)$, we get for the average code length of symbols the following:

$$\frac{1}{n} \mathbb{E}L \leq H(x_1) + 2/n,$$

so if we increase the n length of the block, then we can get arbitrarily near to the lower bound of the average code length.

It is more rewarding then the block coding version of the Huffman-codes because the space complexity does not vary if we increase the length of the codes. It is performed real-time, so there is not much delay, regardless of the length of the blocks. Unfortunately, we gets $O(1/n)$ decrease in the average code length, while we lost $O(K^n)$ time in the execution time of the algorithm, where K is the size of the symbol set.

2.9 9. topic

Description: Describe the LZ based compression algorithm.

The Lempel-Ziv codes are very different from their predecessors, because it is not using probabilistic models.

There were two initial form of it, the LZ77 and LZ78 (usually cold LZ1 and LZ2). After that, based on these algorithms, several variations has been developed, including LZW (Lempel-Ziv-Welch), LZSS (Lempel–Ziv–Storer–Szymanski), LZMA (Lempel–Ziv–Markov chain algorithm) and many others.

Besides their academic influence, these algorithms formed the basis of several ubiquitous compression schemes, including GIF and the DEFLATE algorithm used in PNG.

They are both theoretically dictionary coders. LZ77 maintains a sliding window during compression. This was later shown to be equivalent to the explicit dictionary constructed by LZ78 – however, they are only equivalent when the entire data is intended to be decompressed. LZ78 decompression allows random access to the input as long as the entire dictionary is available,[dubious – discuss] while LZ77 decompression must always start at the beginning of the input.

In the second of the two papers that introduced these algorithms they are analyzed as encoders defined by finite-state machines. A measure analogous to information entropy is developed for individual sequences (as opposed to probabilistic ensembles). This measure gives a bound on the data compression ratio that can be achieved. It is then shown that there exist finite lossless encoders for every sequence that achieve this bound as the length of the sequence grows to infinity. In this sense an algorithm based on this scheme produces asymptotically optimal encodings.

These codes are called adaptive codes.

2.9.1 LZ77 algorithm

The coder uses a h_a length window to analyze the input stream. The window has 2 parts:

1. Searching buffer, which contains the previously coded h_k number of source symbols
2. Forward-looking buffer, which contains the next h_e source symbols, which is needed to be coded. (In practice, $h_k \gg h_e$

The algorithm looks for the longest sequence in the searching buffer which is in the forward-looking buffer as well, and then it sends a tuple, made from t, h, c, where t is the distance between the found sequence in the search buffer and the forward-looking buffer (offset), h is the matching symbols largest length, and c is first not matching symbol's code word in the forward-looking buffer. We send the first non-matching symbol's code because with this we ensures that we handles the case when none of the forward-looking buffer exists in the searching buffer. In this case t and h are zero. Coding the tuple with fixed-length coding, we should have $\lceil \log_2(h_k) \rceil + \lceil \log_2(h_e) \rceil + \lceil \log_2 |\mathcal{X}| \rceil$, where $|\mathcal{X}|$ is the size of the symbol set. Note that, transmitting the length of the matching symbols needs $\lceil \log_2(h_e) \rceil$, bits not $\lceil \log_2(h_k) \rceil$. The reason behind this is the "aliasing", such that, the matching string can be longer than the length of the searching buffer, so it might contains symbols in the forward-looking buffer as well. After the coding the windows moves forward for $h + 1$ places.

It can be shown that the efficiency of the algorithm is asymptotically approaches the optimal algorithm's, which knows the input stream's distribution.

We are using the recently coded symbols, so we are assuming that the "periodicity" of the coded symbols are smaller then the length of the searching buffer.

However, in very rare cases, when this "periodicity" is larger, then we cannot compress the data.

To solve this problem, they implemented an upgraded version of LZ1, and it became the LZ2.

2.9.2 LZ78 algorithm

The base idea is the same as the LZ77, but it does not have window for coding.

During encoding and decoding, we build up a dictionary from the previously encoded/decoded symbol sequences.

The coder finds the longest match from the actual position in the dictionary, and sends the (i, c) pair, where i is the index of the matching string in the dictionary, while c is the first non-matching symbols code word, and after that it pushes a new word to the dictionary which is the originally (matched) string concatenated with c word and it gives the next free index to it. If it does not find matching string in the dictionary, then it sends $(0, c)$ over the channel.

It can be shown that the LZ78 or LZ2 is asymptotically optimal as well. One of the common problems of LZ78 is the unbounded dictionary. In practice, we should define an upper bound for the longest word in the dictionary with either deleting the not used entries or after a given time, we are using the algorithm with fixed-length dictionary.

2.9.3 LZW algorithm

After the LZ2 algorithm, Terry Welch derived another version of the LZ codes, which drops the c from the (i, c) pair. It is necessary to have all the symbols with 1 length coded in the dictionary in order to execute the algorithm.

The algorithm is simple: we are reading the s sequence as long as we have it in the dictionary. If we get to a symbol a, for which the sa sequence is not in our dictionary, then we push it to the dictionary, and we start the reading once again from a.

2.10 10. topic

Description: Define the channel capacity and elaborate on its calculation for symmetric channels

---

One of the most important metrics of a (binary symmetric) channel is the capacity of it, because it limits the speed of the transmission.

Mathematically, the theorem can be stated as follows: for every discrete memoryless channel, the channel capacity is

$$C = maxI(X;Y).$$

In case of an IT source, which produces k-bit long binary messages, which is encoded into n-bit long binary messages then we can derive the measure of the encoding by the following fraction:

$$R = \frac{k}{n}.$$

Usually we have limited bandwidth channels and long messages. In order to send them over a limited bandwidth channel, we should compress the data, and we do not want to lose information.

The channel capacity C satisfies the following inequality:

$$R = \frac{k}{n} \leq C.$$

This is the so called Channel coding theorem.

So, it does not matter if we can compress our k-bit long binary message into very short code words, but if we do not satisfy this inequality, then we will not be able to transmit our data over the channel without errors.

2.10.1 Channel capacity of the binary symmetric channels

If we have a BSC, then it means the bit error probability is P_b, so an error might occur with P_b probability, while with $1 - P_b$ probability, the good symbols will be transmitted over the channel.

Calculating the entropy of the P_b, we get:

$$H(P_b) = P_b \log_2 \big(P_b\big) + (1 - P_b) \log_2 \big(1 - P_b\big).$$

From this, we can write an expression for b, given as follows:

$$k = k + kH(P_b) + kH^2(P_b) + ... = k \sum_{i=0}^{\infty} H^i(P_b) = k \frac{1}{1 - H(P_b)}.$$

If we restructure the equation, we get

$$\frac{k}{n} = 1 - H(P_b) = 1 - P_b \log_2 \big(P_b\big) - (1 - P_b) \log_2 \big(1 - P_b\big) \leq C.$$

From this, it can be easily shown that the minimum value of the channel capacity is occurs if $P_b = 0.5$, because then the sent and received messages mutual information would be zero, and they would be totally independent from each other.

2.11 11. topic

Description: Describe the channel coding theorem

The theorem describes the maximum possible efficiency of error-correcting methods versus levels of noise interference and data corruption.

The Shannon theorem states that given a noisy channel with channel capacity C and information transmitted at a rate R, then $R < C$ there exist codes that allow the probability of error at the receiver to be made arbitrarily small.

R can be calculated as follows:

$$R = \frac{k}{n},$$

where k is the length of the message vector, while n is the length of the code word.

his means that, theoretically, it is possible to transmit information nearly without error at any rate below a limiting rate, C.

The converse is also important. If $R > C$, an arbitrarily small probability of error is not achievable. All codes will have a probability of error greater than a certain positive minimal level, and this level increases as the rate increases. So, information cannot be guaranteed to be transmitted reliably across a channel at rates beyond the channel capacity. The theorem does not address the rare situation in which rate and capacity are equal.

Mathematically, the theorem can be stated as follows: for every discrete memoryless channel, the channel capacity is

$$C = maxI(X;Y).$$

For any $\varepsilon > 0$ and $R < C$, for large enough N, there exists a code of length N and $rate \geq R$ and a decoding algorithm, such that the maximal probability of block error is $\leq \varepsilon$.

Assuming that we have a BSC and the probability of bit error P_b is acceptable, then rates up to $R(P_b)$ are achievable, where

$$R(P_b) = \frac{C}{1 - H(P_b)},$$

where $H(P_b)$ is the binary entropy function

$$H(P_b) = -P_b \log_2(P_b) - (1 - P_b) \log_2(1 - P_b).$$

For any P_b, rates greater than $R(P_b)$ are not achievable.

2.11.1 Proof of the theorem

As with several other major results in information theory, the proof of the noisy channel coding theorem includes an achievability result and a matching converse result. These two components serve to bound, in this case, the set of possible rates at which one can communicate over a noisy channel, and matching serves to show that these bounds are tight bounds.

The following outlines are only one set of many different styles available for study in information theory texts.

Proof of achievability

By an AEP-related argument, given a channel, length n strings of source symbols X_1^n, and length n strings of channel outputs Y_1^n, we can define a jointly typical set by the following:

$$\begin{aligned} A_\varepsilon^{(n)} = \Big\{ & (x^n, y^n) \in \mathcal{X} \times \mathcal{Y} \\ & 2^{-n(H(X)+\varepsilon} \leq p(X_1^n) \leq 2^{-n(H(X)-\varepsilon} \\ & 2^{-n(H(Y)+\varepsilon} \leq p(Y_1^n) \leq 2^{-n(H(X)-\varepsilon} \\ & 2^{-n(H(X,Y)+\varepsilon} \leq p(X_1^n, Y_1^n) \leq 2^{-n(H(X,Y)-\varepsilon} \Big\}. \end{aligned}$$

We say that two sequences X_1^n and Y_1^n are jointly typical if they lie in the jointly typical set defined above.

First, in the style of the random coding argument, we randomly generate 2^{nR} codewords of length n from a probability distribution Q.

This code is revealed to the sender and receiver. It is also assumed that one knows the transition matrix $p(y|x)$ for the channel being used.

A message W is chose according to t he uniform distribution on the set of codewords. That is

$$P(W = w) = 2^{-nR},$$

for any $w = 1, 2, ..., 2^{nR}$. After this, the message W is sent across the channel.

The receiver receives a sequence according to

$$P\big(y^n|x^n(w)\big) = \prod_{i=1}^{n} p\big(y_i|x_i(w)\big).$$

Sending these codewords across the channel, we receive Y_1^n, and decode to some source sequence if there exists exactly 1 codeword that is jointly typical with Y. If there are o jointly typical codewords, or if there are more than one, an error is declared. An error also occurs if a decoded codeword does not match the original codeword. This is called typical set decoding.

The probability of error of this scheme is divided into two parts:

1. First, error can occur if no jointly typical X sequences are found for a received Y sequence
2. Second, error can occur if an incorrect X sequence is jointly typical with a received Y sequence

By the randomness of the code construction, we can assume that the average probability of error averaged over all codes does not depend on the index sent. Thus, without loss of generality, we can assume $W = 1$.

From the joint AEP, we know that the probability that no jointly typical X exists goes to 0 as n grows large. We can bound this error probability by ε.

Also from the joint AEP, we know the probability that a particular $X_1^n(i)$ and the Y_1^n resulting from $W = 1$ are jointly typical is$\leq 2^{-n(I(X;Y)-3\varepsilon)}$.

Define $\forall i = 1, 2, ..., 2^{nR}$ the following E_i sets

$$E_i = \left\{ \big(X_1^n(i), Y_1^n\big) \in A_\varepsilon^{(n)} \right\},$$

as the event that a message i is jointly typical with the sequence received when message 1 is sent. Hence the probability of error can be given as follows:

$$\begin{aligned} P(error) &= P(error|W = 1) \leq \\ &\leq P(E_1^c) + \sum_{i=2}^{2^{nR}} P(E_i) \leq \\ &\leq P(E_1^c) + (2^{nR} - 1)2^{-n(I(X;Y)-3\varepsilon)} \leq \\ &\leq \varepsilon + 2^{-n(I(X;Y)-R-3\varepsilon)}. \end{aligned}$$

We can observe that as $n \to \infty$, if $R < I(X;Y)$ for the channel, the probability of error will go to 0.

Finally, given that the average codebook is shown to be "good," we know that there exists a codebook whose performance is better than the average, and so satisfies our need for arbitrarily low error probability communicating across the noisy channel.

Proof of weak converse

Suppose a code of 2^{nR} codewords. Let W be drawn uniformly over this set as an index. Let X^n and Y^n be the codewords and received codewords, respectively. Then

1. $nR = H(W) = H(W|Y^n) + I(W, Y^n)$ using idetities involving entropy and mutual information

2. $nR \leq H(W|Y^n) + I(X^n(W); Y^n)$ since X is a function of W

3. $nR < leq 1 + P_e^{(n)} nR + I\big(X^n(W); Y^n\big)$ by the use of Fano's inequality

4. $nR < leq 1 + P_e^{(n)} nR + nC$ by the fact that capacity is maximized mutual information.

The result of these steps is that

$$P_e^{(n)} \geq 1 - \frac{1}{nR} - \frac{C}{R}.$$

As the block length n goes to infinity, we obtain $P_e^{(n)}$ is bounded away from 0 if R is greater than C – we can get arbitrarily low rates of error only if R is less than C.

Proof of strong converse A strong converse theorem, proven by Wolfowitz in 1957, states that,

$$P_e \geq 1 - \frac{4A}{n(R-C)^2} - e^{-\frac{n(R-C)}{2}}$$

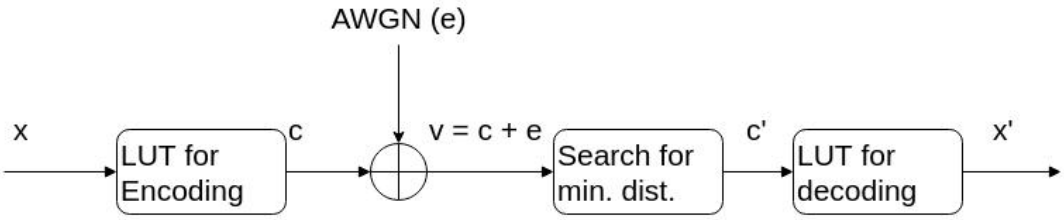
for some finite positive constant A. While the weak converse state that the error probability is bounded away from zero as n tends to infinity, the strong converse states that the error goes to 1. Thus, C is a sharp threshold between perfectly reliable and completely unreliable communication.

2.12 12. topic

Description: Define and explain the relationship between the following properties and parameters of error correcting codes: minimum code distance, code-length and message-length versus performance (Singleton and Hamming bounds), general algorithmic complexity of coding with tables

---

The generic coding scheme is the following:



For error correcting codes, we can introduce some properties and parameters, such as minimum code distance, code-length and message-length versus performance (Singleton and Hamming bounds), and we can give the measure of the algoritmic complexity of the general coding with tables.

Assuming that we have a $\mathcal{C}$ set of codes, and it length is 2^k, where k is the length of the message. Then we can define the minimum code distance by the following equation:

$$d_{min} = min_{c,c' \in \mathcal{C}, c \neq c'} d(c, c') = w_{min},$$

where $d(c, c')$ is the Hamming-distance between c and c', so we count the number of differences between the two vectors element-wise. w_{min} is the minimum weight (number of 1s) of every $c \in \mathcal{C}$.

The optimal code has the maximum of this d_{min}.

Assuming that the length of the code we are transmitting is n, the length of the message is k, and the number of errors we can identify and correct is denoted by t, then we have two bounds which indicates the performance of a given code:

1. Singleton bound, which states that
$$d_{min} \leq n - k + 1,$$
where $n - k$ is the parity of the code.

2. Hamming bound, which states that
$$\sum_{i=0}^{t} \binom{n}{i} \leq 2^{n-k}.$$

If for a given code, $d_{min} = n - k + 1$, then this code is called Max Distance Separable, or MDS.
If for a given code, $\sum_{i=0}^{t} \binom{n}{i} = 2^{n-k}$, then it is called Perfect code.
The relationship between t and d_{min} is the following:

$$t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \tag{2}$$

It is important to state that, the t is the number of correctable errors, while we can define another number l, which indicates the errors which can be identified, and

$$l = d_{min} - 1.$$

In practice, our ultimate goal is to find MDS codes in order the achieve the best possible performance.

With the table-version of the generic coding scheme this can be done, however the algorithmic on-line complexity of the algorithm is three times $O(2^k)$, which is terribly bad.

2.13 13. topic

Description: Introduce the concept of linear block coding and explain the meaning of: systematic codes, generator matrix and parity check matrix (and their relationship), algorithmic complexity of linear block coding (including detection)

The general coding scheme with look-up tables is not so efficient. In order to reduce the complexity of the encoding-decoding algorithms, we can introduce the concept of linear block coding.

Let us assume that we have a set of vectors, $\mathcal{G} := g_1, g_2, ..., g_k$, and $\forall i \in [1, ..., k] : dim(g_i) = n$.

The G set spans the code space, C, so if we are assuming that we have a u message vector, with length k, then we can get the corresponding c code vector with a simple linear transformation, given as follows:

$$c = \sum_{i=1}^{k} u_i g_i.$$

From the $\mathcal{G}$ set we can build a matrix, $G_{k\times n}$, and its rows are the g_i vectors. This is the so called generator matrix. Hence the summation above can be written as matrix-vector multiplication:

$$c = \sum_{i=1}^{k} u_i g_i = uG. \tag{3}$$

This is important, because we can reduce the exponential execution time to polinomial, because it needs $O(kn)$ time.

The next addition of the concept is the systematic codes. Systematic codes are unique in the sense that $\forall c \in \mathcal{C}$, the first k element of c is the coded message vector itself. It can be achieved via the proper construction of G, because if we choose $G_{k\times n} = (I_{k\times k}, B_{k\times(n-k)})$, then $\forall u in \mathcal{U}$ message vector, the corresponding c vector contains u in its first k element.

The decoding of systematic codes are very easy, because we do not have to use LUT for that, instead we can just truncate the last $n - k$ elements, and we get the message.

So far so good, we eliminate two of the three LUT. Unfortunately, we still have to search for the closest c' in the generic coding scheme, which still has $O(2^k)$ time complexity.

In order to eliminate the last part with exponential complexity, we have to introduce the idea of the parity check matrix, $H_{(n-k)\times n}$, which has the following property:

$$Hc^T = 0^T,$$

$\forall c \in \mathcal{C}$. Its role is to signal the errors which might occur during the transmission.

We can further take a look at the property of the paritiy check matrix, and its relationship with the G generator matrix:

$$H(uG)^T = HG^T u^T = 0^T.$$

Because this should be true for every $u \in \mathcal{U}$, we can conclude that

$$HG^T = 0.$$

If the code is systematic, then $H_{(n-k)\times n} = (A_{(n-k)\times k}, I_{(n-k)\times(n-k)}$.

From this, and the definition of G we can derive that

$$A_{(n-k)\times k} + B^T_{k\times(n-k)} = 0 \rightarrow A = B. \tag{4}$$

Equality occurs because these matrices are taking value from the $0, 1$ set, so subtraction is equivalent to addition.

Using the parity check matrix, we can identify the e error vector, and then we can add it to the received v vector, and we should get the original c code vector, due to the binary vector addition's property.

Let us take a look at the "work" of H parity check matrix. First, we multiply it with v^T from right

$$Hv^T = s^T.$$

In this equation, everything is known or and s, the syndrome vector be calculated. This is often called key equation as well. Further on, we can substitute v with $c + e$, so

$$H(c + e)^T = Hc^T + He^T = He^T,$$

due to the property of H. So we derived that

$$He^T = s^T,$$

where only the e vector is unknown, but it is a linear system of equations, which can be solved, although it is under-determined. It can be seen as an indirect observation of e via H and s.

The problem with this is the under-determined nature. Due to that, we have 2^k possible solutions for a specific s vector. Introducing the $E_s = e : He^T = s^T$, $|E_s| = 2^k$ error group, we choose the $e \in E_s$ error vector as the most likely error which might occurred during the transmission.

The most likely error vector is with the following property:

$$e_s = \min_{e \in E_s} w(e),$$

where $w(e)$ is the weight of e in the sense of the number of 1s in e. It is easy to proof that this will be the most likely error vector, because in the case of an e error vector, with length n,

$$P(\#oferrors = i) = P_b^i(1 - P_b)^{n-i} = \left(\frac{P_b}{1 - P_b}\right)^i (1 - P_b)^n,$$

and knowing that $P_b \leq \frac{1}{2}$, we can see that P_b decreases $O\big(exp(i)\big)$.

Now, we have identified the most probably error vector, so after this, we have to add it to the received v vector, and we will get back the c code vector.

The algorithmic complexity of the whole process:

1. Multiplication with G : $O(k * n)$
2. Identification with H : $O(2^{n-k})$
3. Truncation : $O(1)$

The Identification part is done via a LUT, and the LUT can be calculated off-line, so the calculation time of it is not discussed here.

2.14 14. topic

Description: Give the construction of binary Hamming codes (define the corresponding matrices and the error correcting capability)

In the field of communication theory, we do not want to have communication without errors, because it is not achievable, but we want codes, which are capable of correcting a predefined t number of errors instead.

Having said that, we can construct a code, which is capable of correcting every single error. It is often called Hamming code as well.

The technological relevance of Hamming codes is in wired communication.

Let us denote the e vector, which has 1 in the i-th poisition with e_i. Note that i is yet to be determined.

Start the construction from the key equation, so

$$Hv^T = H(c+e)^T = Hc^T + He^T = He^T = s.$$

Substituting e with e_i, we get the i-th row of H matrix as s. Then with a wired matching algorithm, we can define where is the error in e, because if for every row of H, we take the XOR of the row and s element-wise, and then we feed the result to an and AND gate, then we can easily get the value of i.

The conditions of the algorithm are the followings:

1. $\forall i, j \in [1, ..., n], i \neq j : h_i \neq h_j$, where h_i, and h_j are rows of H, so none of the rows of H can be equal
2. $\forall i : h_i \neq 0$, so none of the rows of H can be the whole zero vector

It implies that there is $n = 2^{n-k} - 1$ possible columns for H. The number of columns can be rewritten

$$n + 1 = 2^{n-k} = \sum_{i=0}^{t=1} \binom{n}{i},$$

so the Hamming codes are perfect.

Because the Hamming codes are systematic codes, we only have to construct the H matrix, after that, we just get the G generator matrix transposing the A part of H, due to the property of systematic codes.

The code design for a given P_b and a QoS (Quality of Service), denoted by γ is done via recursion. First we pick n and k, then

1. We check the $P_b' = \Psi(P_b) \leq 10^{-\gamma}$, where $(1 - P_b')^k = (1 - P_b)^n + n(1 - P_b)^{n-k} P_b$.
2. If the inequality is satisfied, then we construct the H and then the G matrices, recpect to n and k
3. If the inequality is not satisfied, then we pick another n, k pair and go back to check the inequality

If we pick 3 for n, and 1 for k, then we get an MDS code, because

$$d_{min} = n - k + 1.$$

Hence we have only two codewords, $(0, 0, 0)$ and $(1, 1, 1)$, and their Hamming-distance is 3.

2.15 15. topic

Description: Describe the Reed Solomon codes (generator matrix, parity check matrix, performance)

In order to achieve a code, which is capable of correcting t number of errors, we have to introduce some new concepts:

1. Extension to q-ary domain, which requieres the concept of finite fields over q, namely Galois-Field, denoted by $GF(q)$
2. Polinoms over $GF(q)$
3. Linear Feed-Back Shift Registers (LFBSR), Linear Feed-Forward Shift Registers (LFFSR)
4. Linear cyclic codes
5. Error Trapping Algorithm

2.15.1 Extension to q-ary domain, $GF(q)$

Unfortunately, if we want to correct 2 or more errors, then we have to extend the components from binary to q-ary. Although it involves a significant danger, because due to the q-ary domain, they probability of error is increased a huge amount!

A finite field can be characterized by a set as follows:

$$GF(q) = 0, 1, ..., q-1.$$

This field is closed to the basic arithmetic operations, such as "$+$" and "$*$". Let us discuss the properties of these operations:

1. Addition ("$+$"):
   (a) closed: $\forall \alpha, \beta \in GF(q) : \alpha + \beta \in GF(q)$
   (b) commutativity: $\forall \alpha, \beta \in GF(q) : \alpha + \beta = \beta + \alpha$
   (c) associativity: $\forall \alpha, \beta, \gamma \in GF(q) : (\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$
   (d) unit: $\forall \alpha \in GF(q), \exists 0 : \alpha + 0 = \alpha$
   (e) inverse: $\forall \alpha \in GF(q), \exists \beta \in GF(q) : \alpha + \beta = 0$
2. Multiplication ("$*$"):
   (a) closed: $\forall \alpha, \beta \in GF(q) : \alpha * \beta \in GF(q)$
   (b) commutativity: $\forall \alpha, \beta \in GF(q) : \alpha * \beta = \beta * \alpha$
   (c) associativity: $\forall \alpha, \beta, \gamma \in GF(q) : (\alpha * \beta) * \gamma = \alpha * (\beta * \gamma)$
   (d) unit: $\forall \alpha \in GF(q), \exists 0 : \alpha * 0 = \alpha$
   (e) inverse: $\forall \alpha \in GF(q) \ \{0\}, \exists \beta \in GF(q) : \alpha * \beta = 0$
3. The two operation above is linked via the distributive rule, so $\forall \alpha, \beta, \gamma \in GF(q) : \alpha(\beta+\gamma) = \alpha\beta+\alpha\gamma$

Hence we have proven that the $GF(q)$ is really a field, because it satisfy every property of a field. The $GF(q)$ has another two property which is necessary:

1. If we raise every element to the $q-1$. power, then we get 1
2. If the powers of $\alpha \in GF(q)$ produce every element of the $GF(q)$, then α is called primitive element.

2.15.2 Polinoms over $GF(q)$

We can define polinoms over $GF(q)$ in the sense that the coefficients of the polinom are from a pre-defined $GF(q)$

$$a(x) = \sum_{i=0}^{n} = a - ix^i, a_i \in GF(q),$$

and the possible solution of $a(x) = 0$ equation, in other words, the possible values of x are the elements of the $GF(q)$. The degree of $a(x)$ is n, denoted by $deg\big(a(x)\big) = n$.

Assuming that we have another polinom over $GF(q)$, $b(x)$, and its degree is m, $m \leq n$. Then if we can calculate the sum, $c(x)$ of $a(x)$ and $b(x)$ with the following expression:

$$c(x) = a(x) + b(x) = \sum_{i=m+1}^{n} a_i x^i + \sum_{i=0}^{m} (a_i + b_i)x^i. \tag{5}$$

Similarly we can produce the product of $a(x)$ and $b(x)$ as follows:

$$c(x) = a(x)b(x) = \sum_{i=1}^{nm} \sum_{j=1}^{\min i,n} a_j b_{i-j} x^i.$$

A given c vector can be transformed into the polinomial space via the following transformation:

$$\mathbb{X}(c) = c(x) = \sum_{i=0}^{n} c_i x^i.$$

The summation of two polynomial is just the summation of two vectors, but the ordinary product of two polynomial is the discrete convolution of two vectors.

The consequence of the fundamental theory of algebra

$\forall a(x), d(x) : deg\big(a(x)\big) = n > deg\big(d(x)\big) = k, \exists q(x), r(x) : a(x) = q(x)d(x)+r(x)$, where $deg\big(r(x)\big) < deg\big(d(x)\big)$.

Another consequence of the theorem is the following: if we substitute u to x in $a(x)$ for which $a(x) = 0$, then $\exists b(x)$ for which

$$a(x) = b(x)(x - u), \tag{6}$$

where $deg\big(b(x)\big) < deg\big(a(x)\big)$.

The proof of if is quite simple, because

$$a(x)\Big|_{x=u} = b(x)(x - u)\Big|_{x=u} + r(x) = 0, \tag{7}$$

and $b(x)(x - u)$ is 0, if $x = 0$, then $r(x)$ should be 0 as well.

The fundamental theory of algebra is as follows:

$$a(x) = c * \prod_{i=1}^{deg(a(x))} (x - u_i),$$

where c is some constant. Another consequence is that, the number of roots of a polynomial cannot exceed the degree of the polynomial.

The fundamental theory of algebra states that every non-zero, single-variable, degree n polynomial with complex coefficients has, counted with multiplicity, exactly n roots.

2.15.3 Shift registers

The (linear) shift registers are well-known parts of the information and communication systems, because they can perform operations real-time. There are three-kind of shift registers that have to be introduced in order to build Reed-Solomon codes:

1. Linear Feed Forward Shift Registers, which can perform multiplication of two polynomials
2. Linear Feed Back Shift Registers, which can perform division of two polynomials, and it has two subtype:
   (a) Division without remainder
   (b) Division with remainder

2.15.4 Linear cyclic codes

Linear cyclic codes means that if we have a $c \in \mathcal{C}$ code vector, and we apply the $\mathcal{S}$ shift operator on it, then we get another c' which is in the set $\mathcal{C}$ as well.

In case of polynomials as codes, we can define the shifting with the following expression:

$$c'(x) = xc(x) \mod (x^n - 1),$$

where $xc(x) = \sum_{i=0}^{n-1} c_i x^{i+1}$. Carrying out the division, we get

$$c'(x) = c_{n-1} + \sum_{i=0}^{n-1} c_i x^{i+1}. \tag{8}$$

Further on, we can rewrite this as follows:

$$xc(x) = c_{n-1}(x^n - 1) + c'(x).$$

So, linear cyclic codes have the following two properties for every $c(x), c'(x) \in \mathcal{C}$:

1. $xc(x) \mod x^n - 1 \in \mathcal{C}$
2. $\alpha c(x) + \beta c'(x) \in \mathcal{C}$

For every linear cyclic code there exists a $g(x)$ generator polynomial, for which:

1. $deg\big(g(x)\big) = n - k$
2. $g_{n-k} = 1$ (main polynom)
3. $\forall c \in \mathcal{C} : c(x) = u(x)g(x)$
4. $g(x) \big| x^n - 1$

The third statement indicates that the encoding is just a simply multiplication of the message and the generator polynomial, carried out on a LFFSR.

In order to proof it let $a(x) \in \mathcal{C}$, $deg\big(a(x)\big) = m < deg\big(c(x)\big), c(x) \in \mathcal{C}$.

Construct the $g(x)$ generator polinom as follows:

$$g(x) = a_m^{-1} a(x),$$

where a_m^{-1} is the multiplicative inverse of a_m in $GF(q)$. This $g(x)$ is unique, because let us say, that there exists a $g'(x)$ for which

$$g(x) - g'(x) \in \mathcal{C}, \tag{9}$$

with $deg\big(g(x) - g'(x)\big) < m$, which is contradiction.

Using the linear cyclic property of the code, we know that $\sum_{i=0}^{n-m-1} u_i x^i g(x) \in \mathcal{C}$. This can be rewritte as follows:

$$\sum_{i=0}^{n-m-1} u_i x^i g(x) = g(x) \sum_{i=0}^{n-m-1} u_i x^i = u(x)g(x).$$

Furthermore, $\nexists c(x) : c(x) = u(x)g(x) + r(x)$, where $r(x) \neq 0$.

For this given $g(x)$ generator polynomial we can define the corresponding $h(x)$ parity check polynomial as well, hence we know that $h(x)c(x) = 0 \mod (x^n - 1)$. Substituting the previous result, we get $h(x)g(x)u(x) = 0 \mod (x^n - 1)$.

The last equality should hold for every $u(x)$ message polynomial, so $h(x)g(x) = x^n - 1$. Using this we can conclude that, there exists $h(x)$, which can be expressed as follows:

$$h(x) = \frac{x^n - 1}{g(x)}.$$

Further on, if we want to get back a $u(x)$ message polynomial from a received – and corrected – $c(x)$ code vector, we just have to divide it by the $g(x)$ generator polynom, because

$$c(x) = u(x)g(x) \rightarrow u(x) = \frac{c(x)}{g(x)}.$$

So, we have replaced the matrix multiplication of the encoding part with an LFFSR, which is real-time, and we replaced the truncation with another shift register, more precisely, an LFBSR, which carries out a division. To both shift register we can load the $g(x)$ generator polynomial, and during the communication it runs blazingly fast.

2.15.5 Error Trapping Algorithm

If we are using linear, cyclic codes, then we have managed to replace two part of the generic coding scheme with shift registers, which is very efficient due to the nature of these devices.

Unfortunately, we have not reduce the on-line complexity of the error groups LUT, so for a given n and k, the complexity of it is still $O(2^{n-k})$, plus we have to calculate the Hv matrix multiplication as well, which further increase the complexity, but with smaller amount.

The main idea of the ETA is the following:

$$v(x) = c(x) + e(x) = u(x)g(x) + e(x),$$

so if we divide the received $v(x)$ by $g(x)$, then we get the $e(x)$ as the remainder of the division. Hence, we know that $deg\big(e(x)\big) \leq n - k - 1$.

For this algorithm, we have to assume that every error which occurred in a given transmission is located in an $n - k$ length segment, where n is the length of the code word, while k is the length of the message word, then we can give the main idea of

It might be possible that the errors are not in the first $n - k - 1$ elements of the error vector, instead it starts from a give i_0, but then, the upper bound of the place of a given error is $i_0 + n - k - 1$.

Hence, generally we can write our $e(x)$ polynomial as follows:

$$e(x) = \sum_{j=0}^{n-k-1} e_{i_{0+j}} x^{i_{0+j}}.$$

It can bee seen that the degree of the $e(x)$ above is greater than $n - k - 1$, but fortunately we can resolve this issue.

Let us take the received $v(x)$ vector

$$v(x) = c(x) + e(x) = u(x)g(x) + e(x) = a(x)g(x) + r(x).$$

We want to show that, $r(x) \neq e(x)$, and $deg\big(r(x)\big) \leq n - k - 1$.

Let $e(x) = b(x)g(x) + s(x)$, and substitute this to the definition of $v(x)$:

$$v(x) = u(x)g(x) + b(x)g(x) + s(x).$$

Now grouping the equation, we get

$$v(x) = \big(u(x) + b(x)\big)g(x) + s(x) = a(x)g(x) + r(x).$$

If we divide $v(x)$ by $g(x)$ then we can observe the same remainder. Let us multiply the two sides of the equation above with x^{-i_0}:

$$x^{-i_0}v(x) = x^{-i_0}u(x)g(x) + x^{-i_0}e(x) = a'(x)g(x) + r(x),$$

where $x^{-i_0}e(x) = r(x)$, so $e(x) = x^{i_0}r(x)$. Hence we have to do i_0 left shift on $v(x)$ with the help of a shift register. After that we divide this $v'(x)$ by $g(x)$, and we get $r(x)$ vector which has to be shifted i_0 times right in order to get $e(x)$. After achieving $e(x)$, we can simply add it to the original $v(x)$, and we will get the correct $c(x)$.

The only problem with the algorith abova is that the exact value of i_0 is yet to be determined. Let us take a closer look at the following equation:

$$x^{-i_0}v(x) = a'(x)g(x) + r(x),$$

where $r(x) = x^{-i_0}e(x)$.

Hence, for an arbitrary i_0 we can express $x^{-i_0}e(x)$ as follows:

$$x^{-i_0}e(x) = b'(x)g(x) + r(x),$$

because $r(x) = x^{-i_0}e(x)$ if i_0 is the place of the first non-zero element of the e error vector.

If we want to correct t number of errors, then

$$d_{min} = w_{min} = 2t + 1.$$

We can write the equation above in the following form as well:

$$x^{-i_0}e(x) - r(x) = b'(x)g(x),$$

and if i_0 is the place of the first non-zero element of e, then $b'(x)g(x) = 0$. The weights of the equation's components are the followings:

$$\begin{aligned} w\big(x^{-i_0}e(x)\big) &< t \\ w\big(r(x)\big) &\geq t+1 \\ w\big(b'(x)g(x)\big) &\geq 2t+1. \end{aligned}$$

Our goal is to reduce the weight of $r(x)$ to be lower than $t+1$. Hence we can obtain this if we starts i from 0, and increment it in every step if $w\big(r(x)\big) \geq t+1$. If we get below the limit, we stop, and we stores i, because it is i_0, so we have to use this when shifting.

2.15.6 Reed-Solomon codes

Now we have everything at hand to describe the Reed-Solomon codes. We will give two version of it. One is with matrices over $GF(Q)$, while the other one is implemented on shift registers.

Reed-Solomon are very famous codes because they are MDS codes, so

$$d_{min} = n - k + 1,$$

and they are able to correct every t number of errors, where t is pre-defined.

Matrix form

Let say we have a set $\mathcal{A} = \alpha_0, \alpha_1, ..., \alpha_{n-1}$, where $\forall i, j = 1, 2, ..., n-1, i \neq j : \alpha_i \neq \alpha_j$. $|\mathcal{A}| = n = q-1$.

Then, if we have a $u = (u_0 u_1 ... u_{k-1})$ message vector, then we can construct the corresponding $u(x) = \sum_{i=0}^{k-1} u_i x^i$ polynomial, and

$$\forall j : c_j = u(x)\Big|_{x=\alpha_j} = \sum_{i=0}^{k-1} u_i \alpha^i,$$

where c_j is the j-th element of the corresponding code vector. Hence, to obtain the full c, we have to multiply u with G generator matrix. The arbitrary i, j element of the matrix can be given as follows:

$$G_{i,j} = \alpha_i^j,$$

where $i = 0, 1, ..., n-1$ and $j = 0, 1, ..., k-1$. It can be proven easily that these are MDS codes:

$$d_{min} = w_{min} = n - (k-1) = n - k + 1,$$

because the number of maximum zero components is $k-1$. In order to reduce the complexity of the G generator matrix, we can pick the primitive element, α from the $GF(q)$, because its powers generate every components of the $GF(q)$. Hence an arbitrary element of G can be rewritten as follows:

$$G_{i,j} = \alpha^{ij},$$

where $i = 0, 1, ..., n-1$ and $j = 0, 1, ..., k-1$ as well. The arbitrary element of H parity check matrix can be given nearly identically:

$$H_{i,j} = \alpha^{ij},$$

where $i = 0, 1, ..., n-1$ and $j = 1, 2, ..., n-k$. We can prove that this follows the property of H, such as

$$Hc^T = H(uG)^T = HG^T u^T = 0^T.$$

Since $\exists u \in \mathcal{U} : u \neq 0$, HG^T should be equal to zero.

Proving the statement is easy. Let us take the l-th element of Hc^T:

$$(Hc^T)_l = \sum_{i=0}^{n-1} \alpha^{li} c_i = \sum_{i=0}^{n-1} \alpha^{li} \sum_{j=0}^{k-1} \alpha ij u_j = \sum_{j=0}^{k-1} \left(\sum_{i=0}^{n-1} \alpha^{i(l+j)} \right) u_j.$$

The last result is a finite geometric series, so

$$\begin{aligned}
\sum_{i=0}^{n-1} \alpha^{i(l+j)} &= \frac{\alpha^{(l+j)n} - 1}{\alpha^{l+j} - 1} = \\
&= \frac{(\alpha^n)^{l+j} - 1}{\alpha^{l+j} - 1} = \\
&= \frac{(\alpha^{q-1})^{l+j} - 1}{\alpha^{l+j} - 1} = \\
&= \frac{1^{l+j} - 1}{\alpha^{l+j} - 1} = \\
&= 0.
\end{aligned}$$

We used the property of the $GF(q)$, namely $\forall i : \alpha_i^{q-1} = 1$. Now we can see that in the summation above the inner summation takes the value 1 for every l, so $Hc^T = 0$.

The decoding from c is done via solving the following equation for u:

$$uG = c,$$

where G and c are given, and we have to calculate u. It is an over-determined linear system of equations, because c has length n, while u has length k. It is always solvable, because we can truncate the unnecessary part of c, and we can solve the well-defined linear system of equations by taking the inverse of the $G_{k \times k}$ submatrix of G. Note that truncation is not enough itself, because R-S codes are not systematic.

Constructing an R-S code we have to define n and k. If we define a t, such as the numbers of errors to be corrected, then starting from the definition (using the MDS property of R-S)

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor, \tag{10}$$

and $n = q - 1$, we can calculate the difference between n and k, because it is two times larger than the number of errors that we want to correct.

The tricky part of the generic coding scheme in case of Reed-Solomon codes is the LUT for the group leaders of every syndrome vector, because the q-ary domain it raised from $O(2^{n-k})$ to $O(q^{n-k})$ which is incredibly high.

In order to reduce this overhead, we can construct R-S codes over shift registers.

Reed-Solomon codes with linear shift registers Fortunately Reed-Solomon codes are linear cyclic codes, however they are not systematic ones.

It means the R-S codes can be implemented on shift registers which is amazing, because it makes it real-time besides the MDS property.

We know that for R-S codes $\forall i, \alpha^i : c(\alpha^i) = 0$. Using this we can express $c(x)$ as follows:

$$c(x) = \prod_{i=1}^{n-k}(x - \alpha^i)u(x),$$

where $\prod_{i=1}^{n-k}(x - \alpha^i) = g(x)$.

Furthermore for every $i = 1, 2, , ..., n$ if we substitute α^i into $x^n - 1$, we get zero, because $(\alpha^i)^n = (\alpha^n)^i = (\alpha^{q-1})^i = 1^i = 1$. Hence

$$x^n - 1 = \prod_{i=1}^{n}(x - \alpha^i) = \prod_{i=1}^{n-k}(x - \alpha^i) \prod_{i=n-k+1}^{n}(x - \alpha^i),$$

where

$$\prod_{i=1}^{n-k}(x - \alpha^i) = g(x),$$

so

$$h(x) = \prod_{i=n-k+1}^{n}(x - \alpha^i).$$

Now we arrived to a very special place. We have an optimal (MDS) code with the optimal (real-time) realization. However, we cannot be exactly happy, because the complexity of the shift registers are so high cause of the q-ary domain, it cannot be implemented on the current families of VLSI chips.

In order to make it implementable, we should introduce the fundamentals of minimal polynoms over $GF(q)$, and the Bose-Chaudhuri-Hocquenghem codes, which are not MDS, but close to optimal and actually easily implementable, because every multiplication inside of a shift register is implemented with either a wire or nothing, because the polynomials in this code has 1 or 0 as coefficients.

2.16 16. topic

Description: Describe the steps of the PGZ algorithm for detection

---

The PGZ error detecton algorithm is used by e.g. the BCH code.

Peterson's algorithm is the second step of the generalized BCH decoding procedure. Peterson's algorithm is used to calculate the error locator polynomial coefficients $\lambda_1, \lambda_2, ..., \lambda_v$ of a polynomial

$$\Lambda(x) = 1 + \sum_{i=1}^{k} \lambda_i x^i.$$

Expect that we have at least $2t$ syndromes $s_c, ..., s_{c+2t-1}$. Let $k = t$.

The steps of the procedure are the followings:

1. Start by generating $S_{k \times k}$ matrix, with elements that are syndrome vectors. The general element of S can be given as follows: $S_{i,j} = s_{c+i+j}$, $i = 0, 1, ..., k-1$, $j = 0, 1, ..., k-1$
2. Generate a C vector, for which $C_i = s_{c+k+i}$, $i = 0, 1, ..., v-1$
3. Let Λ denote the unknown polynomial coefficients, which are given by $\Lambda_i = \lambda_{k-i}$, $i = 0, 1, ..., v-1$
4. Form the matrix equation $S\Lambda = -C$.
5. If the determinant of matrix S is nonzero, then we can actually find an inverse of this matrix and solve for the values of unknown Λ values.
6. If $det(S) = 0$, then decrement the value of k, and continue the algorithm with the first step. The algorithm goes until $k > 0$. If we reach $k = 0$, then declare an empty error locator polynomial.

After the algorithm, we should have the polynomial coefficients in Λ, even if it is empty.

2.17 17. topic

Description: Summarize the different description of convolution encoders (architecture, state graph and transfer function) and compare the performance with linear block coding

---

????

2.18 18. topic

Description: Briefly summarize the Viterbi algorithm and its complexity for decoding convolutional codes

---

????

2.19 19. topic

Description: Describe the CDMA/FH system

CDMA is a basic form of digital modulation used in spread spectrum signal transmission. It abbreviates the code division multiple access expression.

One of its versions is the frequency hopping CDMA, or shortly, FH-CDMA or CDMA/FH. It is the repetitive switching of frequency at the time of radio transmission.

This helps to reduce the strength of electronic warfare, that is, the unauthorized jamming or interception of telecommunications. Spread spectrum allows a signal to be carried over a frequency band, which is considerably wider than the minimum bandwidth needed by the information signal. The energy that is originally centered in the narrowband is spread by the transmitter over many different frequency band channels on a broader electromagnetic spectrum.

Some of the advantages are enhanced privacy – because it looks like a random sequence to an attacker –, reduced narrowband interference, and improved signal capacity.

In the FH-CDMA technique, a transmitter hops between all available frequencies based on a specific algorithm, which is either preplanned or random. The transmitter functions in synchronization with a receiver, which stays tuned to the exact same center frequency as the transmitter's.

A short data burst is carried on a narrowband. Afterwards, the transmitter tunes to a different frequency and transmits again. Therefore, the receiver has the ability to hop its frequency across a specified bandwidth many times per second, transmitting using one particular frequency for a specific time frame, then hopping to yet another frequency and transmitting again.

The devices that make use of FH-CDMA technology consume less power and are usually cost effective. The greatest benefit of FH-CDMA is based on the coexistence of various access points in the same area, which is not possible when using direct sequence (DS).

There are some rules that control the way the frequency-hopping devices are utilized. For example, in North America, the industrial, scientific, and medical wave band (ISM wave band) is split into 75 hopping channels. The power transmission of these hopping channels does not exceed 1 watt on any channel. This restriction makes sure that an individual device does not consume an excessive amount of bandwidth or remain excessively on a single frequency.

First we divide the $time - frequency$ plane into $k \times k$ blocks. For every time value to every row we can assign a frequency value, coded in binary numbers.

Every user has his/her own unique c_i code, which is a $k \times k$ block as well. These codes are binary superimposed codes. It means that the disjunctions (bitwise OR) of any pair of distinct at most m tuples of codewords have to be different. It means that we can define $O(N^2) = M$ number of different code, showed by Einarsson.

The encoding algorithm is simple, we just add the code to the message column-wise, so we add the first columns frequency values, then the second columns, etc. Every message means a symbol, and for every symbol we assign the corresponding numbered row from the block matrix.

Because many users can use the channel simultaneously, in the received code it might happen, that we have other users coded messages as well. Unfortunately it might possible that two users' coded message will interfere, in the sense that they have data in the same i, j block, which implies an erasure.

In case of decoding, we subtract every users code block from the received one, and after that we count the number of entries in every row for every decoded messages. After that we choose the row with the most entries as the originally sent message.

2.20 20. topic

Description: Describe the CDMA/DS system and the Walsh-Hadamard codes

---

CDMA-DS is different from the frequency hopping spread spectrum method. DS stands for direct sequences.

This digital modulation method is analogous to those used in simple radio transceivers. In the analog case, a low frequency data signal is time multiplied with a high frequency pure sine wave carrier, and transmitted. This is effectively a frequency convolution (Wiener–Khinchin theorem) of the two signals, resulting in a carrier with narrow sidebands.

In the digital case, the sinusoidal carrier is replaced by Walsh functions. These are binary square waves that form a complete orthonormal set. The data signal is also binary and the time multiplication is achieved with a simple XOR function. This is usually a Gilbert cell mixer in the circuitry.

Synchronous CDMA exploits mathematical properties of orthogonality between vectors representing the data strings.

Each user in synchronous CDMA uses a code orthogonal to the others' codes to modulate their signal. Orthogonal codes have a cross-correlation equal to zero; in other words, they do not interfere with each other.

Start with a set of vectors that are mutually orthogonal. (Although mutual orthogonality is the only condition, these vectors are usually constructed for ease of decoding, for example columns or rows from Walsh matrices.)

In mathematics, a Walsh matrix is a specific square matrix with dimensions of some power of 2, entries of ± 1, and the property that the dot product of any two distinct rows (or columns) is zero.

The natural ordered Hadamard matrix is defined by the recursive formula below, and the sequency ordered Hadamard matrix is formed by rearranging the rows so that the number of sign-changes in a row is in increasing order. Confusingly, different sources refer to either matrix as the Walsh matrix.

$$H(2^1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$H(2^2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

and in general

$$H(2^k) = \begin{bmatrix} H(2^{k-1}) & H(2^{k-1}) \\ H(2^{k-1}) & -H(2^{k-1}) \end{bmatrix} = H(2) \otimes H(2^{k-1}),$$

for $2 \leq k \in \mathbb{N}$, where $\otimes$ denotes the Kronecker product.

Each element of the set of orthogonal vectors will be assigned to individual users and are called code, chip code or chipping code.

Let v be the assigned vector to a users. A 1 bit is represented by transmitting a positive code, v, and a 0 bit is represented by a negative code, $-v$.

Now, due to physical properties of interference, if two signals at a point are in phase, they add to give twice the amplitude of each signal, but if they are out of phase, they subtract and give a signal that is the difference of the amplitudes. Digitally, this behaviour can be modelled by the addition of the transmitted vectors, component by component.

The encode is being in 2 steps:

1. First, we transform the $u \in 0, 1^k$ message vector to $u' \in -1, 1^k$.
2. Second, we encode u' with a given v vector, using the procedure above, so we produce the Kronecker (outer) product of u' and v, and we rewrite it to a vector, row-wise.

After the encoding, we transmit it to the receiver. Because it is transmitted in the air, it might be possible that others transmit their code to in the same time, so they add to produce the raw signal.

This raw signal is called an interference pattern. The receiver then extracts an intelligible signal for any known sender by combining the sender's code with the interference pattern.

It is done via the following algorithm:

1. Separate the received vector into k number of m length blocks, where m is the length of the code words
2. Produce the dot product of every separated blocks by the code word
3. The original message can be obtained, if we assign 1 to positive values and 0 to negative values.

if every values of decoded vector is zero, than it means that the sender did not transmit any data. It is because of the orthogonal nature of the code words.

The length of the $c_i, i = 1, 2, ...M$ code words can be calculated as follows:

$$dim(c_i) = N = \frac{T_s}{T_c},$$

where T_s is the period time of the service, while T_c is coming from technology. The first one is defined in the channel itself, while the latter one is the minimum period time which is capable of achieving on a given device.

2.20.1 Mathematically derivation of CDMA/DS

Let M, the number of users using the same communication channel. Then due to the element-wise addition of the message, we can treat the communication as bit-wise, because it can be extended further on.

Let us assume that every i user wants to send a $y_i \in -1, 1$ message over the channel. First, we have to multiply it with an $s_i(t)$ continuous function, and sum them up, due to the nature of the multiplexing. After that, we have to add some $\nu(t) \sim R(\tau) = N_0\delta(\tau)$ noise due to the noisy behaviour of the channel.

Now we can express the $x(t)$ received message as follows:

$$x(t) = \nu(t) + \sum_{i=1}^{M} y_i s_i(t).$$

Please note that the real model for this is the following:

$$x(t) = \nu(t) + \sum_{i=1}^{M} \alpha_i y_i s_i(t - \tau_i) = \nu(t) + \sum_{i=1}^{M} \alpha_i y_i * s_i(t),$$

where $*$ stands for convolution.

The procedure of the decoding for every i user is the following:

$$x_i(t) = \frac{1}{T_s} \int_0^{T_s} x(t) s_i(t) dt.$$

In order to get the $\hat{y}_i$ calculated – hopefully original – message we have to use the signum function for x_i, which return -1 if $x_i < 0$ and 1 if $x_i \geq 0$.

Hence we have to construct $\mathcal{C}_opt$ to satisfy the following:

$$\mathcal{C}_{opt} = \min_{\mathcal{C}} P(\hat{y} \neq y),$$

where $\hat{y}, y$ are the decoded and the sent message vector respectively, where the i-th component of them is the message of the i-th user.

Further on, let x be the decoded vector before applying the signum function then

$$x_l = \frac{1}{T_s}\int_0^{T_s} x(t)s_l(t)dt = \frac{1}{T_s}\int_0^{T_s}\left(\nu(t) + \sum_{i=1}^{M} y_i s_i(t)\right)s_l(t)dt =$$
$$= \sum_{i=1}^{M} y_i \frac{1}{T_s}\int_0^{T_s} s_i(t)s_l(t)dt + \frac{1}{T_s}\int_0^{T_s}\nu(t)s_l(t)dt.$$

Using our criteria for $dim(c_i)$, namely

$$N = dim(c) = \frac{T_s}{T_c},$$

we can derive how many users can be communicating over a channel at a given time.

Unfortunately, we have to make the system discrete. In order to do this, we have to construct the R matrix. The general element of R can be given as follows:

$$R_{l,i} = \frac{1}{T_s}\int_0^{T_s} s_l(t)s_i(t)dt = \frac{1}{N}\sum_{k=1}^{M} c_k^{(i)}c_k^{(l)} = \frac{1}{N}c^{(l)}c^{(i)}.,$$

where $c_k^{(j)}$ is the k-th element in the j-th code vector.

Hence we can get the arbitrary element x_l, $l = 1, 2, ..., M$ of x by the next summation:

$$x_l = \nu_l + \sum_{i=1}^{M} R_{l,i}y_i,$$

where

$$\nu_l = \frac{1}{T_s}\int_0^{T_s}\nu(t)s_l(t)dt.$$

The expression of x_l can be rewritten in matrix-vector multiplicative form:

$$x = Ry + \nu.$$

Hence we can make the coding scheme very easy, because every y multi-user message can be multiplied by R, then transmit over the channel, adding ν to it. However, the decoding part is yet to be determined.

Let us take a closer look at the x_l component:

$$x_l = \nu_l + R_{l,l}y_l + \sum_{i=1,i\neq l}^{M} R_{l,i}y_i,$$

where $\sum_{i=1,i\neq l}^{M} R_{l,i}y_i$ is the multi-user interference.

Furthermore, $c^{(i)}$ and $c^{(l)}$ are pair-wise orthogonal, so

$$R_{l,i} = \begin{cases} 1; & \text{if } l = i \\ 0; & \text{otherwise} \end{cases}$$

It is because

$$R_{l,l} = \frac{1}{N}\parallel c^{(l)} \parallel^2 = \frac{1}{N}\sum_{i=1}^{N}(c_i^{(l)})^2 = \frac{1}{N}N = 1.$$

It means that

$$x_l = y_l + \nu_l.$$

Unfortunately, high speed services are not economical in the case of using orthogonal codes.

The optimal multi user codeset can be defined as follows:

$$\mathcal{C}_{opt}\min MUI,$$

where MUI stands for multi user interference. Hence we have to satisfy the following equation for a predefined $\varepsilon > 0$:

$$P(\hat{y} \neq y) \leq \varepsilon.$$

We have not give the decoding algorithm yet, so let us derive that as well.

The expected value of $\nu_l, l = 1, 2, ..., M$ is as follows:

$$\mathbb{E}(\nu_l) = 0 = \mathbb{E}\left(\frac{1}{T_s}\int_0^{T_s} \nu(t)s_l(t)dt\right) = \frac{1}{T_s}\int_0^{T_s} \mathbb{E}\big(v(t)\big)s_l(t)dt,$$

because $\nu(t)$ is an additive, Gaussian white noise.

The Wiener–Khinchin theorem sates that $R_\nu(\tau)$ is the following:

$$R_\nu(\tau) = \int_{-\infty}^{\infty} s_\nu(f)e^{j2\pi f\tau}df, \tag{11}$$

so $R_\nu(\tau)$ is the Fourier-transform of $s_\nu(f)$ where R_ν is the auto-correlation function of ν, while $s_\nu(f)$ is the power spectral density function of $\nu(t)$.

Let us construct a matrix, from $K_{l,i}$ general elements, which can be expressed

$$\begin{aligned}
K_{l,i} &= \mathbb{E}(\nu_l\nu_i) = \\
&= \mathbb{E}\left(\frac{1}{T_s^2}\int_0^{T_s} \nu(t)s_l(t)dt\int_0^{T_s} \nu(\tau)s_i(\tau)d\tau\right) = \\
&= \frac{1}{T_s^2}\mathbb{E}\left(\iint_0^{T_s} \nu(t)\nu(\tau)s_l(t)s_i(\tau)dtd\tau\right) = \\
&= \frac{1}{T_s^2}\iint_0^{T_s} \mathbb{E}\big(\nu(t)\nu(\tau)\big)s_l(t)s_i(\tau)dtd\tau = \\
&= \frac{1}{T_s}\iint_0^{T_s} N_0\delta(t-\tau)s_l(t)s_i(\tau)dtd\tau = \\
&= \frac{N_0}{T_s^2}\int_0^{T_s} s_l(t)s_i(t)dt = \\
&= \frac{N_0}{T_s}\frac{1}{N}\sum_{k=1} c_k^{(l)}c_k^{(i)} = \\
&= R_{l-i}\frac{N_0}{T_s},
\end{aligned}$$

where we used that

$$\mathbb{E}\big(\nu(t)\nu(\tau)\big) = R_\nu(t-\tau),$$

and we know that the spectral density function of $\nu(t)$ is constant, with value N_0, so

$$R_\nu(t-\tau) = N_0\delta(t-\tau).$$

Now, let $\nu \sim \mathcal{N}(0, K)$, where K is the covariance matrix of ν, and

$$K = \frac{N_0}{T_s}R.$$

Now our goal is to minimize the probability of error, so we are about to construct a code, for which

$$\min P(\hat{y} \neq y).$$

On the decoding side we choose $\hat{y}$ for which

$$\begin{aligned}\max_{y\in -1,1^m} P(y|x) &\sim \max_{y\in -1,1^m} \frac{P(x|y)P(y)}{P(x)} \sim \\ &\sim \max_{y\in -1,1^m} P(x|y) \sim \\ &\sim \max_{y\in -1,1^m} \frac{1}{\sqrt{(2\pi)^M \det K}} e^{-\frac{1}{2}(x-Ry)^T K^{-1}(x-Ry)} \sim \\ &\sim \min_{y\in -1,1^m} (x-Ry)^T \frac{T_s}{N_0} R^{-1}(x-Ry) \sim \\ &\sim \min_{y\in -1,1^m} y^T Ry - 2x^T y.\end{aligned}$$

This minimization algorithm needs $O(2^M)$ step which can be further reduced with Hopfield Neural Network to $O(M^2)$, because if we consider the k-th step of the HNN algorithm, we can express the l-th element of $y(k+1)$ as follows:

$$y_l(k+1) = -sgn\Big\{\sum_{i=1}^{N} R'_{l,i} y_i(k) - x_l\Big\},$$

where

$$R' = R - diag(R),$$

for $l = 1, 2, ..., M$.

With this, we have everything at hand to perform CDMA/DS with arbitrary codes. We can use Walsh-Hadamard codes as well, which was described above.

2.21 21. topic

Description: Describe the CDMA/DS system with random codes

We can use CDMA/DS with not just truly orthogonal codes, but with pseudo-random ones as well. Pseudo-random number codes (pseudo-noise or PN code) can be generated very easily. These codes will sum to zero over a period of time. Although the sequence is deterministic because of the limited length of the linear shift register used to generate the sequence, they provide a PN code that can be used within a CDMA system to provide the spreading code required. They are used within many systems as there is a very large number that can be used.

A feature of PN codes is that if the same versions of the PN code are time shifted, then they become almost orthogonal, and can be used as virtually orthogonal codes within a CDMA system. It is often called quasi-orthogonal codes as well.

The code are generated by a pseudo-random noise generator, which means, that the elements have Bernoulli distribution, with $p = 0.5$.

A PN sequence has many features such as having an almost equal number of zeros and ones, very low correlation between shifted versions of the sequence, and very low cross-correlation with other signals such as interference and noise. However, it is able to correlate well with itself and its inverse. Another important aspect is the autocorrelation of the sequence as it determines the ability to synchronize and lock the spreading code for the received signal. This fight effectively effects the multiple interference and improves the SNR. M-sequences, Gold codes, and Kasami sequences are the examples of this class of sequences.

3 Infocommunication systems

3.1 1. topic

Description: Main applications of twisted pair cables (telephone, data, ADSL)

Wireline transmission media

- Symmetrical twisted pair copper cable
- Coaxial cable
- Optical fibre cable
- Constuction issues, connecting, error detection, error localization

Media and cable characteristics

- Transmission parameters (attenuation, delay, reflection, crosstalk, noises, interferences)
- Laying, connecting technologies
- Faults, fault localization
- Matching, accessories, termination

Twisted pair cable: Twisted pair cabling is a type of wiring in which two conductors of a single circuit are twisted together for the purposes of canceling out electromagnetic interference (EMI) from external sources.
Attennuation: Reduction in the strength of a signal. Unit: [dB] Characteristic impedance: Z0. The characteristic impedance of a transmission line is the ratio of the voltage and current of a wave travelling along the line.
NeXT: Near-end Crosstalk. Interference between two pairs in a cable is measured at the same end of the cable as the interfering transmitter.
FeXT: Fear-end Crosstalk. Interference between two pairs of a cable measured at the other end of the cable with respect to the interfering transmitter.
Main applications of twisted pair cable: Ethernet cabling (UTP), telephone and ADSL.

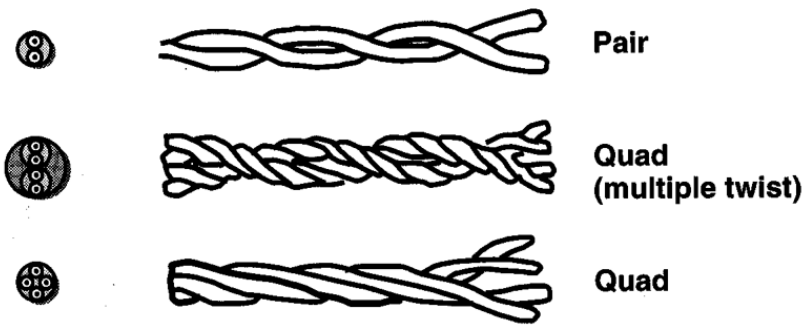


Figure 1: Cable twist type

The main idea of twisting is to reduce the external inference and noise.

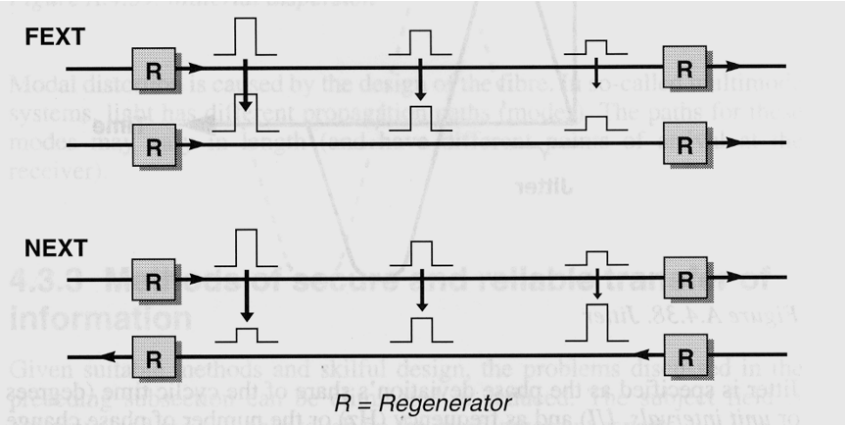


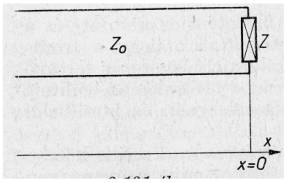
Figure 2: FeXT - NeXT

$$\alpha = \pm \sqrt{\frac{1}{2}(RG - \omega^2 LC) + \frac{1}{2}\sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)}}.$$

$$\beta = \pm \sqrt{\frac{1}{2}(\omega^2 LC - RG) + \frac{1}{2}\sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)}}.$$

$$Z_0 = \sqrt{\frac{R + \mathrm{j}\,\omega L}{G + \mathrm{j}\,\omega C}}$$

Figure 3: Characteristic impedance



$$r = U_0^- / U_0^+$$

$$r = \frac{Z - Z_0}{Z + Z_0}.$$

Figure 4: Phenomenon at the end of terminated wire

3.1.1 Telephone cabling via twisted pair cables

A telephone line or telephone circuit (or just line or circuit within the industry) is a single-user circuit on a telephone communication system. This is the physical wire or other signaling medium connecting the user's telephone apparatus to the telecommunications network, and usually also implies a single telephone number for billing purposes reserved for that user. Telephone lines are used to deliver landline telephone service and Digital subscriber line (DSL) phone cable service to the premises. Telephone overhead lines are connected to the public switched telephone network.

At first, they did not use twisted cable to transmit data from one point to another, and the lines were running above the grounds between poles.

Nowadays, modern lines may run underground, and may carry analog or digital signals to the exchange, or may have a device that converts the analog signal to digital for transmission on a carrier system.

In most cases, two copper wires (tip and ring) for each telephone line run from a home or other small building to a local telephone exchange. There is a central junction box for the building where the wires that go to telephone jacks throughout the building and wires that go to the exchange meet and can be connected in different configurations depending upon the subscribed telephone service. The wires between the junction box and the exchange are known as the local loop, and the network of wires going to an exchange, the access network.

3.1.2 Transmitting data via twisted pair cables

UTP cable is also the most common cable used in computer networking. Modern Ethernet, the most common data networking standard, can use UTP cables. Twisted pair cabling is often used in data networks for short and medium length connections because of its relatively lower costs compared to optical fiber and coaxial cable.

UTP is also finding increasing use in video applications, primarily in security cameras. Many cameras include a UTP output with screw terminals; UTP cable bandwidth has improved to match the baseband of television signals. As UTP is a balanced transmission line, a balun is needed to connect to unbalanced equipment, for example any using BNC connectors and designed for coaxial cable.

3.1.3 ADSL network with twisted pair cables

ADSL principles

- Asymmetric Digital Subscriber line
- A modem technology
- Convert existing twisted-pair telephone lines into access paths for multimedia and high speed data communication
- Can transmit to 30 Mbps downstream (VDSL 100 Mbps)
- Can transmit up to 20 Mbps upstream
- Transform the existing PSTN network to a powerful system capable of bringing multimedia, full motion video to the subscriber's home

Technology

- No ultimate technology!
- Frequency division multiplexing, time division multiplexing, modulation, error control, flow control, scrambling, signal processing, adaptation, STM-ATM, trellis coding, in-service performance monitoring and surveillance, initialisation, handshaking, channel analysis, are all mixed in ADSL
- More room for further development...

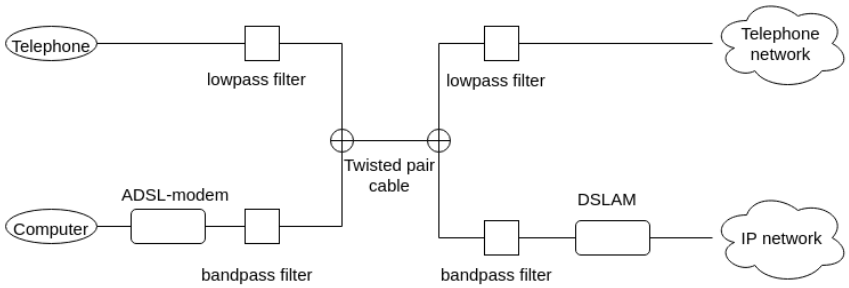


Figure 5: ADSL system components

VDSL Very-high-bit-rate digital subscriber line (VDSL or VHDSL) is a digital subscriber line (DSL) technology providing data transmission faster than asymmetric digital subscriber line (ADSL) over a single flat untwisted or twisted pair of copper wires (up to 52 Mbit/s downstream and 16 Mbit/s upstream), and on coaxial cable (up to 85 Mbit/s down- and upstream) using the frequency band from 25 kHz to 12 MHz. These rates mean that VDSL is capable of supporting applications such as high-definition television, as well as telephone services (voice over IP) and general Internet access, over a single connection. VDSL is deployed over existing wiring used for analog telephone service and lower-speed DSL connections. This standard was approved by ITU in November 2001.

3.2 2. topic

Description: Main characteristics of optical fiber cables. Main applications of fiber cables

---

An optical fiber cable is a cable containing one or more optical fibers that are used to carry light. The optical fiber elements are typically individually coated with plastic layers and contained in a protective tube suitable for the environment where the cable will be deployed. Different types of cable are used for different applications, for example long distance telecommunication, or providing a high-speed data connection between different parts of a building.

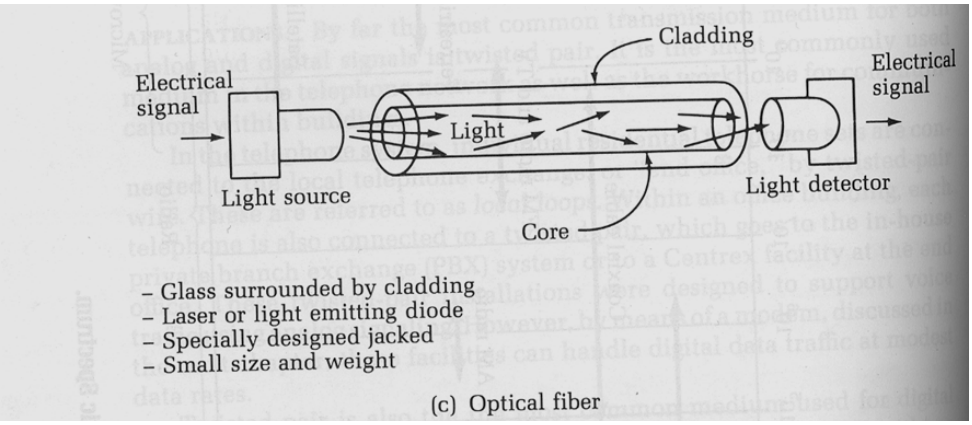


Figure 6: Principle of optical fibre

Optical fiber consists of a core and a cladding layer, selected for total internal reflection due to the difference in the refractive index between the two. In practical fibers, the cladding is usually coated with a layer of acrylate polymer or polyimide. This coating protects the fiber from damage but does not contribute to its optical waveguide properties. Individual coated fibers (or fibers formed into ribbons or bundles) then have a tough resin buffer layer and/or core tube(s) extruded around them to form the cable core. Several layers of protective sheathing, depending on the application, are added to form the cable. Rigid fiber assemblies sometimes put light-absorbing ("dark") glass between the fibers, to prevent light that leaks out of one fiber from entering another. This reduces cross-talk between the fibers, or reduces flare in fiber bundle imaging applications.

Optical cables transfer data at the speed of light in glass. This is the speed of light in vacuum divided by the refractive index of the glass used, typically around 180,000 to 200,000 km/s, resulting in 5.0 to 5.5 microseconds of latency per km. Thus the round-trip delay time for 1000 km is around 11 milliseconds.

The intermodal dispersion can be characterized by its coefficient

$$D_{lm} = \frac{\Delta\tau}{L}[ns/km] \tag{12}$$

where $\Delta\tau$ is the group delay difference between the slowest and fastest mode and L is the length of the cable.

The modal dispersion means that the signal is spread in time because the propagation velocity of the optical signal is not the same for all modes. Actually it further limits the bandwidth of multimode fibers.

Typical modern multimode graded-index fibers have 3 dB/km of attenuation loss (50 % loss per km) at 850 nm and 1 dB/km at 1300 nm. Singlemode 9/125 loses 0.4 dB/km at 1310 nm and 0.25 dB/km at 1550 nm. Very high quality singlemode fiber intended for long distance applications is specified at a loss of 0.19 dB/km at 1550 nm. POF (plastic optical fiber) loses much more: 1 dB/m at 650 nm. Plastic optical fiber is large core (about 1mm) fiber suitable only for short, low speed networks such as within cars.

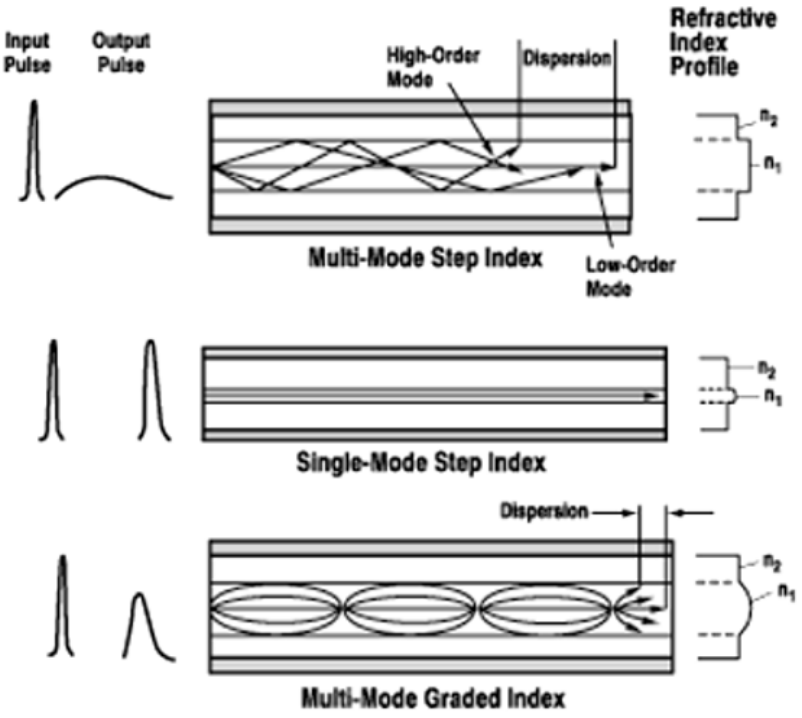


Figure 7: Types of mode propagation in fiber optic cable

Each connection made adds about 0.6 dB of average loss, and each joint (splice) adds about 0.1 dB. Depending on the transmitter power and the sensitivity of the receiver, if the total loss is too large the link will not function reliably.

Invisible IR light is used in commercial glass fiber communications because it has lower attenuation in such materials than visible light. However, the glass fibers will transmit visible light somewhat, which is convenient for simple testing of the fibers without requiring expensive equipment. Splices can be inspected visually, and adjusted for minimal light leakage at the joint, which maximizes light transmission between the ends of the fibers being joined.

Optical fibers are very strong, but the strength is drastically reduced by unavoidable microscopic surface flaws inherent in the manufacturing process. The initial fiber strength, as well as its change with time, must be considered relative to the stress imposed on the fiber during handling, cabling, and installation for a given set of environmental conditions. There are three basic scenarios that can lead to strength degradation and failure by inducing flaw growth: dynamic fatigue, static fatigues, and zero-stress aging.

3.3 3. topic

Description: Main radio wave propagation modes, transmission characteristics of radio connections

---

Radio transmission media

- Frequency bands and wave propagation modes
- Terrestrial radio connection
- Satellite communication
- In door radio connection

Media characteristics

- Transmission parameters (path loss, delay, fading, radio interferences)
- Reliability and availability - equipment and propagation parameters (lightning, snow, rain, fog, smoke)
- Openness – interferences - privacy

Fading: In wireless communications, fading is deviation of the attenuation affecting a signal over certain propagation media. The fading may vary with time, geographical position or radio frequency, and is often modeled as a random process. A fading channel is a communication channel comparising fading. In wireless systems, fading may either be due to multipath propagation, referred to as multipath induced fading, or due to shadowing from obstacles affecting the wave propagation, sometimes referred to as shadow fading.

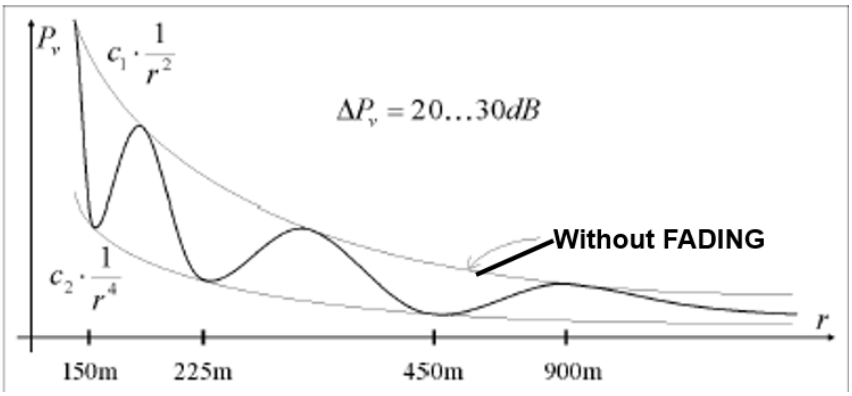
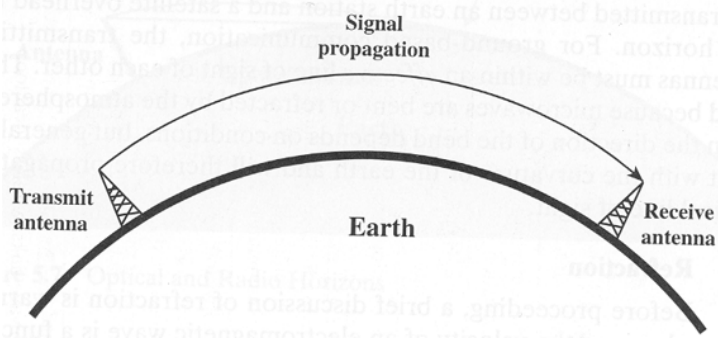


Figure 8: Fading

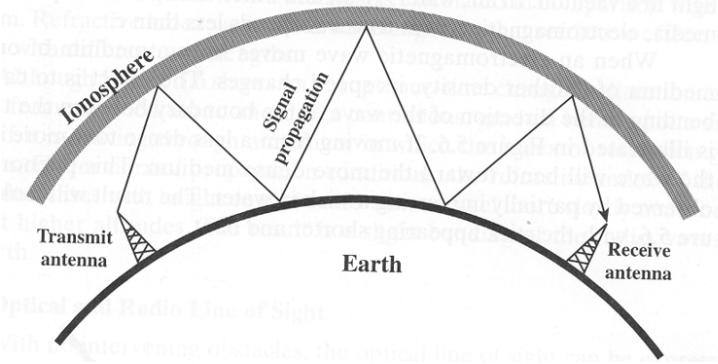
Frequency bands and wave propagation modes

- LF (30-300 kHz)
- MF (300-3000 kHz)
- HF (3-30 MHz)
- VHF (30-300 MHz)
- UHF (300-3000 MHz)
- SHF (centimetric waves, 3-30 GHz)
- EHF (millimetric waves, 30-300 GHz)



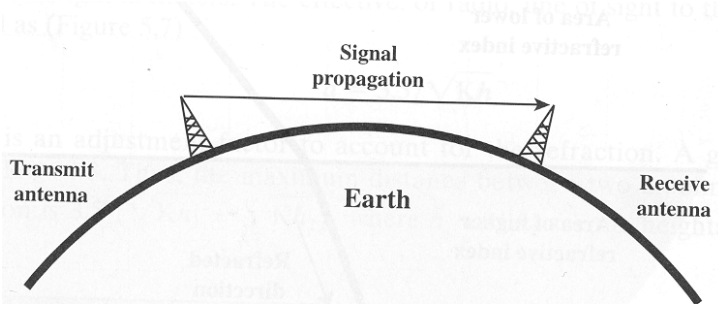
LF
MF

Figure 9: Ground wave propagation (below 2 MHz)



HF

Figure 10: Sky wave propagation (2 to 30 MHz)



UHF
VHF
SHF
EHF

Figure 11: Line-of-sight (LOS) propagation

3.4 4. topic

Description: Main functions of multiplexing and switching nodes in the networks, the main features of circuit switching, packet switching and cell switching

---

Main functions of multiplexing nad switching nodes

- To reduce transmission costs
- To utilize higher bandwidth
- "Framing" and "packing" of information
- TDM - Time Division Multiplexing
- FDM - Frequency Division Multiplexing
- CDMA - Code Division Multiple Access
- WDM - Wavelength Division Multiplexing

The Time Division Multiplexing concept

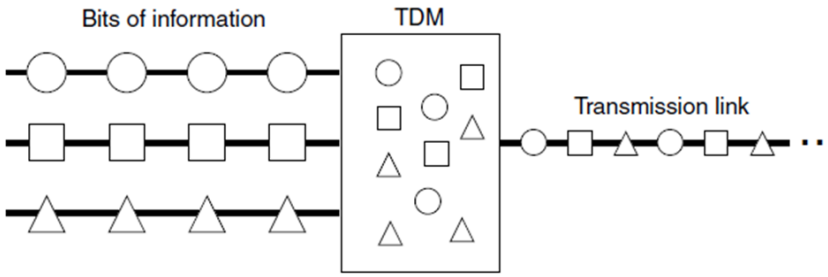


Figure 12: TDM

Time-division multiplexing (TDM) is a method of transmitting and receiving independent signals over a common signal path by means of synchronized switches at each end of the transmission line so that each signal appears on the line only a fraction of time in an alternating pattern. It is used when the data rate of the transmission medium exceeds that of signal to be transmitted. This form of signal multiplexing was developed in telecommunications for telegraphy systems in the late 19th century, but found its most common application in digital telephony in the second half of the 20th century.

In telecommunications, frequency-division multiplexing (FDM) is a technique by which the total bandwidth available in a communication medium is divided into a series of non-overlapping frequency sub-bands, each of which is used to carry a separate signal. This allows a single transmission medium such as the radio spectrum, a cable or optical fiber to be shared by multiple independent signals. Another use is to carry separate serial bits or segments of a higher rate signal in parallel.

The most natural example of frequency-division multiplexing is radio and television broadcasting, in which multiple radio signals at different frequencies pass through the air at the same time. Another example is cable television, in which many television channels are carried simultaneously on a single cable. FDM is also used by telephone systems to transmit multiple telephone calls through high capacity trunklines, communications satellites to transmit multiple channels of data on uplink and downlink radio beams, and broadband DSL modems to transmit large amounts of computer data through twisted pair telephone lines, among many other uses.

An analogous technique called wavelength division multiplexing is used in fiber-optic communication, in which multiple channels of data are transmitted over a single optical fiber using different wavelengths (frequencies) of light.

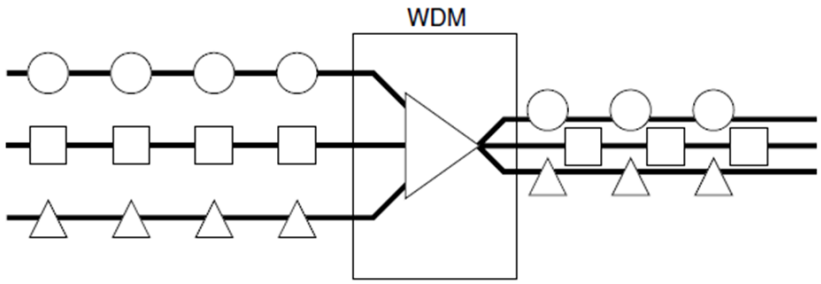


Figure 13: WDM

Wavelength Division Multiplexing Frequency Division Multiple Access principles

In fiber-optic communications, wavelength-division multiplexing (WDM) is a technology which multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths (i.e., colors) of laser light. This technique enables bidirectional communications over one strand of fiber, as well as multiplication of capacity.

The term wavelength-division multiplexing is commonly applied to an optical carrier (which is typically described by its wavelength), whereas frequency-division multiplexing typically applies to a radio carrier (which is more often described by frequency). Since wavelength and frequency are tied together through a simple directly inverse relationship, in which the product of frequency and wavelength equals c (the propagation speed of light), the two terms actually describe the same concept.

Switching techniques in public networks

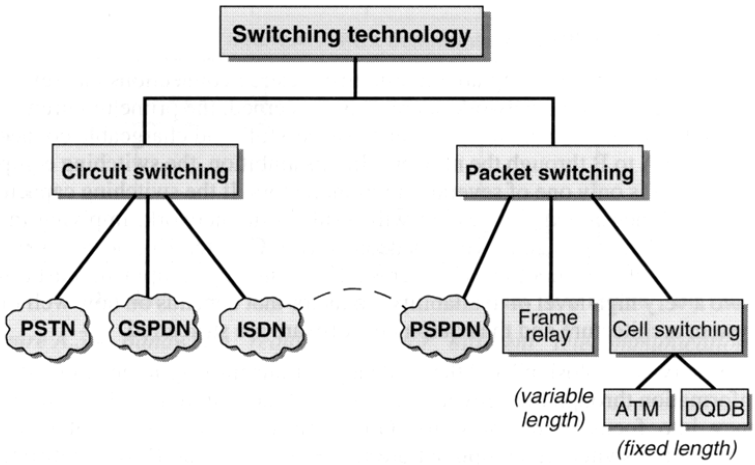


Figure 14: Switching technology

Main features of circuit, packet and cell switching

Circuit switching: Circuit switching is a methodology of implementing a telecommunications network in which two network nodes establish a dedicated communications channel (circuit) through the network before the nodes may communicate. The circuit guarantees the full bandwidth of the channel and remains connected for the duration of the communication session. The circuit functions as if the nodes were physically connected as with an electrical circuit.

The defining example of a circuit-switched network is the early analog telephone network. When a call is made from one telephone to another, switches within the telephone exchanges create a continuous wire circuit between the two telephones, for as long as the call lasts.

Circuit switching contrasts with packet switching which divides the data to be transmitted into packets transmitted through the network independently. In packet switching, instead of being dedicated to one communication session at a time, network links are shared by packets from multiple competing communication sessions, resulting in the loss of the quality of service guarantees that are provided by circuit switching.

In circuit switching, the bit delay is constant during a connection, as opposed to packet switching, where packet queues may cause varying and potentially indefinitely long packet transfer delays. No circuit can be degraded by competing users because it is protected from use by other callers until the circuit is released and a new connection is set up. Even if no actual communication is taking place, the channel remains reserved and protected from competing users.

Virtual circuit switching is a packet switching technology that emulates circuit switching, in the sense that the connection is established before any packets are transferred, and packets are delivered in order.

While circuit switching is commonly used for connecting voice circuits, the concept of a dedicated path persisting between two communicating parties or nodes can be extended to signal content other than voice. Its advantage is that it provides for continuous transfer without the overhead associated with packets making maximal use of available bandwidth for that communication. Its disadvantage is that it can be relatively inefficient because unused capacity guaranteed to a connection cannot be used by other connections on the same network.

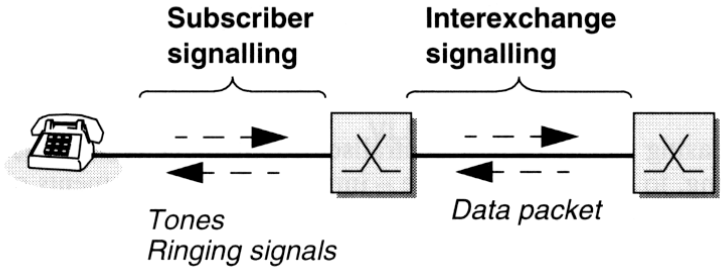


Figure 15: Signalling principles in circuit switching

Packet switching: Packet switching is a digital networking communications method that groups all transmitted data into suitably sized blocks, called packets, which are transmitted via a medium that may be shared by multiple simultaneous communication sessions. Packet switching increases network efficiency, robustness and enables technological convergence of many applications operating on the same network.

Packets are composed of a header and payload. Information in the header is used by networking hardware to direct the packet to its destination where the payload is extracted and used by application software.

Packet switching features delivery of variable bit rate data streams, realized as sequences of packets, over a computer network which allocates transmission resources as needed using statistical multiplexing or dynamic bandwidth allocation techniques. As they traverse network nodes, such as switches and routers, packets are received, buffered, queued, and transmitted (stored and forwarded), resulting in variable latency and throughput depending on the link capacity and the traffic load on the network.

Packet switching contrasts with another principal networking paradigm, circuit switching, a method which pre-allocates dedicated network bandwidth specifically for each communication session, each having a constant bit rate and latency between nodes. In cases of billable services, such as cellular communication services, circuit switching is characterized by a fee per unit of connection time, even when no data is transferred, while packet switching may be characterized by a fee per unit of information transmitted, such as characters, packets, or messages.

Packet mode communication may be implemented with or without intermediate forwarding nodes (packet switches or routers). Packets are normally forwarded by intermediate network nodes asynchronously using first-in, first-out buffering, but may be forwarded according to some scheduling discipline for fair queuing, traffic shaping, or for differentiated or guaranteed quality of service, such as weighted fair queuing or leaky bucket. In case of a shared physical medium (such as radio or 10BASE5), the packets may be delivered according to a multiple access scheme.

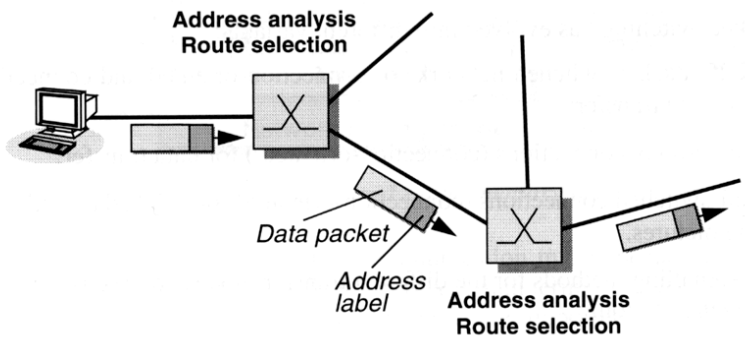


Figure 16: Signalling in packet switched networks

Cell switching: It refers to a method of statistically multiplexing small fixed-length packets, called "cells", to transport data between computers or kinds of network equipment. It is an unreliable, connection-oriented packet switched data communications protocol. It is often called cell relay.

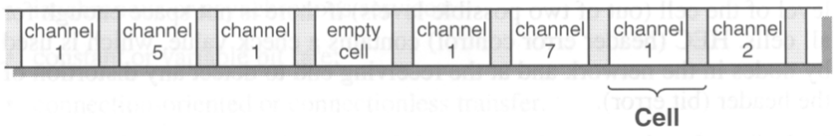


Figure 17: Cell switching

Cell relay transmission rates usually are between 56 kbit/s and several gigabits per second. ATM, a particularly popular form of cell relay, is most commonly used for home DSL connections, which often runs between 128 kbit/s and 1.544 Mbit/s (DS1), and for high-speed backbone connections (OC-3 and faster).

Cell relay protocols have neither flow control nor error correction capability, are information-content independent, and correspond only to layers one and two of the OSI Reference Model.

Cell relay can be used for delay- and jitter-sensitive traffic such as voice and video.

Cell relay systems break variable-length user packets into groups of fixed-length cells, that add addressing and verification information. Frame length is fixed in networking hardware, based on time delay and user packet-length considerations. One user data message may be segmented over many cells.

Cell relay systems may also carry bitstream-based data such as PDH traffic, by breaking it into streams of cells, with a lightweight synchronization and clock recovery shim. Thus cell relay systems may potentially carry any combination of stream-based and packet-based data. This is a form of statistical time division multiplexing.

Cell relay is an implementation of fast packet-switching technology that is used in connection-oriented broadband integrated services digital networks (B-ISDN, and its better-known supporting technology ATM) and connectionless IEEE 802.6 switched multi-megabit data service (SMDS).

At any time there is information to be transmitted; the switch basically sends the data units. Connections don't have to be negotiated like circuit switching. Channels don't have to be allocated because channels do not exist in ATM, and on condition that there is an adequate amount of bandwidth to maintain it, there can be indefinite transmissions over the same facility.

Cell relay utilizes data cells of a persistent size. Frames are comparable to data packets; however they contrast from cells in that they may fluctuate in size based on circumstances. This type of technology is not secure for the reason that its procedures do not support error handling or data recovery. Per se, all delicate and significant transmissions may perhaps be transported faster via fixed-sized cells, which are simpler to transmit compared to variable-sized frames or packets.

Cell relay is extremely reliable for transporting vital data. Switching devices give the precise method to cells as each endpoint address embedded in a cell. An example of cell relay is ATM, a prevalent form utilized to transfer a cell with a fixed size of 53 bytes.

3.5 5. topic

Description: Main elements and characteristics of PDH, SDH and ATM systems

---

Main elements and characteristics of PDH systems

The plesiochronous digital hierarchy (PDH) is a technology used in telecommunications networks to transport large quantities of data over digital transport equipment such as fibre optic and microwave radio systems.

The term plesiochronous is derived from Greek *plēsios*, meaning near, and chronos, time, and refers to the fact that PDH networks run in a state where different parts of the network are nearly, but not quite perfectly, synchronized.

The data rate is controlled by a clock in the equipment generating the data. The rate is allowed to vary by ±50 ppm of 2.048 kbit/s (according to ITU-T recommendation). This means that different data streams can (and probably do) run at slightly different rates from one another.

In order to move multiple data streams from one place to another, they are multiplexed in groups of four. This is done by taking 1 bit from stream #1, followed by 1 bit from stream #2, then #3, then #4. The transmitting multiplexer also adds additional bits in order to allow the far end receiving multiplexer to decode which bits belong to which data stream, and so correctly reconstitute the original data streams. These additional bits are called "justification" or "stuffing" bits.

Because each of the four data streams is not necessarily running at the same rate, some compensation has to be introduced. The transmitting multiplexer combines the four data streams assuming that they are running at their maximum allowed rate. This means that occasionally, (unless the 2 Mbit/s really is running at the maximum rate) the multiplexer will look for the next bit but it will not have arrived. In this case, the multiplexer signals to the receiving multiplexer that a bit is "missing". This allows the receiving multiplexer to correctly reconstruct the original data for each of the four 2 Mbit/s data streams, and at the correct, different, plesiochronous rates.

The resulting data stream from the above process runs at 8.448 Mbit/s (about 8 Mbit/s). Similar techniques are used to combine four 8 Mbit/s together, plus bit stuffing, giving 34 Mbit/s. Four 34 Mbit/s, gives 140. Four 140 gives 565.

PDH hierarchy:

- different in Japan, United States, Europe
- levels:
  - E1: 2 Mbit/s
  - E2: 8 Mbit/s
  - E3: 34 Mbit/s
  - E4: 140 Mbit/s
  - E5: 565 Mbit/s

Európai hierarchia:

hierarchia szint	0	E1	E2	E3	E4	E5
névleges sebesség [Mb/s]	0,064 (PCM)	2	8	34 (34>8x4!!!)	140	565
beszédcsatornák száma	1	30	4×30 = 120	4×120=480	4×480=1920	4×1920 = 7680
átviteli közeg	szimmetrikus kábel csavart érpár					
		koaxiális kábel				
	földfelszíni és műholdas rádió					
				fénykábel		

Figure 18: PDH Europe hierarchy

PCM (Pulse Code Modulation): 125μs frames

- European PCM Frame: 32 time slots * 8 bits * 8000 = 2048 kbit/s
- American PCM Frame: (24 time slots * 8 bits + 1 bit) * 8000 = 1544 kbit/s

Main elements and characteristics of SDH systems

- SDH – Synchronous Digital Hierarchy
- VC – Virtual Container (multiplexing level)
- STM-N Synchronous Transport Modules (line signal level)
- POH – path overhead (control and supervisory information)
- POH+Payload=VC
- A number of VCs can packaged into a larger VC

This is a standardized protocol that transfer multiple digital bit streams synchronously over optical fiber using lasers or highly coherent light from light-emitting diodes (LEDs). At low transmission rates data can also be transferred via an electrical interface. The method was developed to replace the plesiochronous digital hierarchy (PDH) system for transporting large amounts of telephone calls and data traffic over the same fiber without synchronization problems.

SDH differs from Plesiochronous Digital Hierarchy (PDH) in that the exact rates that are used to transport the data on SDH are tightly synchronized across the entire network, using atomic clocks. This synchronization system allows entire inter-country networks to operate synchronously, greatly reducing the amount of buffering required between elements in the network. SDH can be used to encapsulate earlier digital transmission standards, such as the PDH standard, or they can be used to directly support either Asynchronous Transfer Mode (ATM) or so-called packet over SDH (POS) networking. Therefore, it is inaccurate to think of SDH as communications protocols in and of themselves; they are generic, all-purpose transport containers for moving both voice and data. The basic format of a SDH signal allows it to carry many different services in its virtual container (VC), because it is bandwidth-flexible.

The protocol is a heavily multiplexed structure, with the header interleaved between the data in a complex way. This permits the encapsulated data to have its own frame rate and be able to "float around" relative to the SDH frame structure and rate. This interleaving permits a very low latency for the encapsulated data. Data passing through equipment can be delayed by at most 32 microseconds (μs), compared to a frame rate of 125 μs; many competing protocols buffer the data during such transits for at least one frame or packet before sending it on. Extra padding is allowed for the multiplexed data to move within the overall framing, as the data is clocked at a different rate than the frame rate. The protocol is made more complex by the decision to permit this padding at most levels of the multiplexing structure, but it improves all-around performance.

The basic unit of framing in SDH is a STM-1 (Synchronous Transport Module, level 1), which operates at 155.520 megabits per second (Mbit/s).

In packet-oriented data transmission, such as Ethernet, a packet frame usually consists of a header and a payload. The header is transmitted first, followed by the payload (and possibly a trailer, such as a CRC). In synchronous optical networking, this is modified slightly. The header is termed the overhead, and instead of being transmitted before the payload, is interleaved with it during transmission. Part of the overhead is transmitted, then part of the payload, then the next part of the overhead, then the next part of the payload, until the entire frame has been transmitted.

In the case of an STM-1, nine octets of overhead are transmitted, followed by 261 octets of payload. This is also repeated nine times until 2,430 octets have been transmitted, taking 125 μs. For SDH, this is often represented by displaying the frame graphically: as a block of 270 columns and nine rows for STM1. This representation aligns all the overhead columns, so the overhead appears as a contiguous block, as does the payload.

The Synchronous Transport Module, level 1 (STM-1) frame is the basic transmission format for SDH—the first level of the synchronous digital hierarchy. The STM-1 frame is transmitted in exactly

125 μs, therefore, there are 8,000 frames per second on a 155.52 Mbit/s OC-3 fiber-optic circuit. The STM-1 frame consists of overhead and pointers plus information payload. The first nine columns of each frame make up the section overhead and administrative unit pointers, and the last 261 columns make up the information payload. The pointers (H1, H2, H3 bytes) identify administrative units (AU) within the information payload. Thus, an OC-3 circuit can carry 150.336 Mbit/s of payload, after accounting for the overhead.

Carried within the information payload, which has its own frame structure of nine rows and 261 columns, are administrative units identified by pointers. Also within the administrative unit are one or more virtual containers (VCs). VCs contain path overhead and VC payload. The first column is for path overhead; it is followed by the payload container, which can itself carry other containers. Administrative units can have any phase alignment within the STM frame, and this alignment is indicated by the pointer in row four.

The section overhead (SOH) of a STM-1 signal is divided into two parts: the regenerator section overhead (RSOH) and the multiplex section overhead (MSOH). The overheads contain information from the transmission system itself, which is used for a wide range of management functions, such as monitoring transmission quality, detecting failures, managing alarms, data communication channels, service channels, etc.

The STM frame is continuous and is transmitted in a serial fashion: byte-by-byte, row-by-row.

Transport overhead The transport overhead is used for signaling and measuring transmission error rates, and is composed as follows:

Section overhead Called regenerator section overhead (RSOH) in SDH terminology: 27 octets containing information about the frame structure required by the terminal equipment.

Line overhead Called multiplex section overhead (MSOH) in SDH: 45 octets containing information about error correction and Automatic Protection Switching messages (e.g., alarms and maintenance messages) as may be required within the network. The error correction is included for STM-16 and above.

Administrative unit (AU) pointer Points to the location of the J1 byte in the payload (the first byte in the virtual container).

Path virtual envelope Data transmitted from end to end is referred to as path data. It is composed of two components:

Payload overhead (POH) 9 octets used for end-to-end signaling and error measurement.

Payload User data (774 bytes for STM-0, or 2,340 octets for STM-1).

The highest rate commonly deployed is the STM-256 circuit, which operates at rate of just under 38.5 Gbit/s.

SONET szintek	**STS-1**	**STS-3**	**STS-12**	**STS-48**	**STS-192**
SDH szintek		**STM-1**	**STM-4**	**STM-16**	**STM-64**
névleges átviteli sebesség	52 Mb/s	155 Mb/s	622 Mb/s	2,5 Gb/s	10 Gb/s
beszédcsatornák száma	672	USA: 3×672 = 2016 EU: 1920	EU: 4×1920 = 7680	EU: 4×7680 = 30720	EU: 4×30720 = 122880
átviteli közeg	földfelszíni és műholdas rádió				
	optikai kábel				

Figure 19: SDH Europe hierarchy

Transport modules

- RSOH - Regenerate Section Overhead
- MSOH - Multiplexer Section
- AU Pointer – Administrative Unit Pointer (specifies where the payload starts)
- Duration of STM-1 module is 125 μs

Main elements and characteristics of ATM systems

ATM: Asynchronous Transfer Mode (ATM) is, according to the ATM Forum, "a telecommunications concept defined by ANSI and ITU (formerly CCITT) standards for carriage of a complete range of user traffic, including voice, data, and video signals".

It was designed for a network that must handle both traditional high-throughput data traffic (e.g., file transfers), and real-time, low-latency content such as voice and video. The reference model for ATM approximately maps to the three lowest layers of the ISO-OSI reference model: network layer, data link layer, and physical layer.

ATM is a core protocol used over the SDH backbone of the public switched telephone network (PSTN) and Integrated Services Digital Network (ISDN), but its use is declining in favour of all IP.

ATM provides functionality that is similar to both circuit switching and packet switching networks: ATM uses asynchronous time-division multiplexing, and encodes data into small, fixed-sized packets (ISO-OSI frames) called cells. This differs from approaches such as the Internet Protocol or Ethernet that use variable sized packets and frames. ATM uses a connection-oriented model in which a virtual circuit must be established between two endpoints before the actual data exchange begins. These virtual circuits may be "permanent", i.e. dedicated connections that are usually pre-configured by the service provider, or "switched", i.e. set up on a per-call basis using signalling and disconnected when the call is terminated.

ATM eventually became dominated by Internet Protocol (IP) only technology (and Wireless or Mobile ATM never got any foothold).

In the ISO-OSI reference model data link layer (layer 2), the basic transfer units are generically called frames. In ATM these frames are of a fixed (53 octets or bytes) length and specifically called "cells".

Cell size

If a speech signal is reduced to packets, and it is forced to share a link with bursty data traffic (traffic with some large data packets) then no matter how small the speech packets could be made, they would always encounter full-size data packets. Under normal queuing conditions the cells might experience maximum queuing delays. To avoid this issue, all ATM packets, or "cells," are the same small size. In addition, the fixed cell structure means that ATM can be readily switched by hardware without the inherent delays introduced by software switched and routed frames.

Thus, the designers of ATM utilized small data cells to reduce jitter (delay variance, in this case) in the multiplexing of data streams. Reduction of jitter (and also end-to-end round-trip delays) is particularly important when carrying voice traffic, because the conversion of digitized voice into an analogue audio signal is an inherently real-time process, and to do a good job, the decoder (codec) that does this needs an evenly spaced (in time) stream of data items. If the next data item is not available when it is needed, the codec has no choice but to produce silence or guess — and if the data is late, it is useless, because the time period when it should have been converted to a signal has already passed.

At the time of the design of ATM, 155 Mbit/s Synchronous Digital Hierarchy (SDH) with 135 Mbit/s payload was considered a fast optical network link, and many plesiochronous digital hierarchy (PDH) links in the digital network were considerably slower, ranging from 1.544 to 45 Mbit/s in the USA, and 2 to 34 Mbit/s in Europe.

At this rate, a typical full-length 1500 byte (12000-bit) data packet would take 77.42 μs to transmit. In a lower-speed link, such as a 1.544 Mbit/s T1 line, a 1500 byte packet would take up to 7.8 milliseconds.

A queuing delay induced by several such data packets might exceed the figure of 7.8 ms several times over, in addition to any packet generation delay in the shorter speech packet. This was clearly unacceptable for speech traffic, which needs to have low jitter in the data stream being fed into the codec if it is to produce good-quality sound. A packet voice system can produce this low jitter in a number of ways:

Have a playback buffer between the network and the codec, one large enough to tide the codec over almost all the jitter in the data. This allows smoothing out the jitter, but the delay introduced by passage through the buffer would require echo cancellers even in local networks; this was considered too expensive at the time. Also, it would have increased the delay across the channel, and conversation is difficult over high-delay channels. Build a system that can inherently provide low jitter (and minimal overall delay) to traffic that needs it. Operate on a 1:1 user basis (i.e., a dedicated pipe). The design of ATM aimed for a low-jitter network interface. However, "cells" were introduced into the design to provide short queuing delays while continuing to support datagram traffic. ATM broke up all packets,

data, and voice streams into 48-byte chunks, adding a 5-byte routing header to each one so that they could be reassembled later. The choice of 48 bytes was political rather than technical. When the CCITT (now ITU-T) was standardizing ATM, parties from the United States wanted a 64-byte payload because this was felt to be a good compromise in larger payloads optimized for data transmission and shorter payloads optimized for real-time applications like voice; parties from Europe wanted 32-byte payloads because the small size (and therefore short transmission times) simplify voice applications with respect to echo cancellation. Most of the European parties eventually came around to the arguments made by the Americans, but France and a few others held out for a shorter cell length. With 32 bytes, France would have been able to implement an ATM-based voice network with calls from one end of France to the other requiring no echo cancellation. 48 bytes (plus 5 header bytes = 53) was chosen as a compromise between the two sides. 5-byte headers were chosen because it was thought that 10% of the payload was the maximum price to pay for routing information. ATM multiplexed these 53-byte cells instead of packets which reduced worst-case cell contention jitter by a factor of almost 30, reducing the need for echo cancellers.

An ATM cell consists of a 5-byte header and a 48-byte payload. The payload size of 48 bytes was chosen as described above.

ATM defines two different cell formats: UNI (User-Network Interface) and NNI (Network-Network Interface). Most ATM links use UNI cell format:

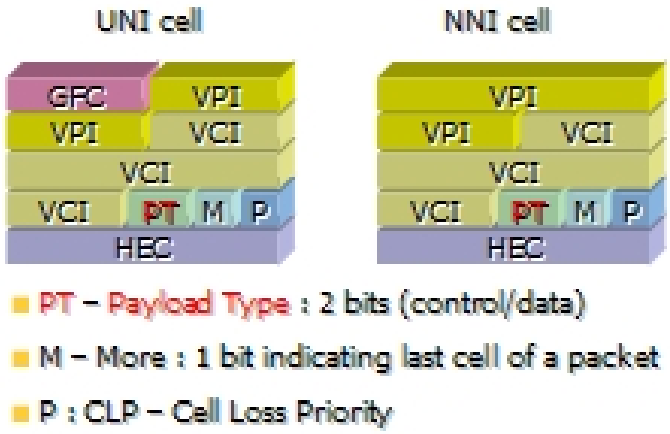


Figure 20: ATM UNI and NNI cell formats

1. GFC = Generic Flow Control (4 bits) (default: 4-zero bits)
2. VPI = Virtual Path Identifier (8 bits UNI, or 12 bits NNI)
3. VCI = Virtual Channel identifier (16 bits)
4. PT = Payload Type (3 bits)
   (a) PT bit 3 (msbit): Network management cell. If 0, user data cell and the following apply:
   (b) PT bit 2: Explicit forward congestion indication (EFCI); 1 = network congestion experienced
   (c) PT bit 1 (lsbit): ATM user-to-user (AAU) bit. Used by AAL5 to indicate packet boundaries.
5. CLP = Cell Loss Priority (1-bit)
6. HEC = Header Error Control (8-bit CRC, polynomial = $x^8 + x^2 + x + 1$)

ATM uses the PT field to designate various special kinds of cells for operations, administration and management (OAM) purposes, and to delineate packet boundaries in some ATM adaptation layers (AAL). If the most significant bit of the PT field is 0, this is a user data cell, and the other two bits are used to indicate network congestion and as a general purpose header bit available for ATM adaptation layers.

If the msbit of the PT bit is 1, this is a management cell, and the other two bits indicate the type. (Network management segment, network management end-to-end, resource management, and reserved for future use.)

Several ATM link protocols use the HEC field to drive a CRC-based framing algorithm, which allows locating the ATM cells with no overhead beyond what is otherwise needed for header protection. The 8-bit CRC is used to correct single-bit header errors and detect multi-bit header errors. When multi-bit header errors are detected, the current and subsequent cells are dropped until a cell with no header errors is found.

A UNI cell reserves the GFC field for a local flow control/submultiplexing system between users. This was intended to allow several terminals to share a single network connection, in the same way that two Integrated Services Digital Network (ISDN) phones can share a single basic rate ISDN connection. All four GFC bits must be zero by default.

The NNI cell format replicates the UNI format almost exactly, except that the 4-bit GFC field is re-allocated to the VPI field, extending the VPI to 12 bits. Thus, a single NNI ATM interconnection is capable of addressing almost 2^{12} VPs of up to almost 2^{16} VCs each (in practice some of the VP and VC numbers are reserved).

Cells in practice

ATM supports different types of services via AALs. Standardized AALs include AAL1, AAL2, and AAL5, and the rarely used AAL3 and AAL4. AAL1 is used for constant bit rate (CBR) services and circuit emulation. Synchronization is also maintained at AAL1. AAL2 through AAL4 are used for variable bitrate (VBR) services, and AAL5 for data. Which AAL is in use for a given cell is not encoded in the cell. Instead, it is negotiated by or configured at the endpoints on a per-virtual-connection basis.

Following the initial design of ATM, networks have become much faster. A 1500 byte (12000-bit) full-size Ethernet frame takes only 1.2 μs to transmit on a 10 Gbit/s network, reducing the need for small cells to reduce jitter due to contention. Some consider that this makes a case for replacing ATM with Ethernet in the network backbone. However, it should be noted that the increased link speeds by themselves do not alleviate jitter due to queuing. Additionally, the hardware for implementing the service adaptation for IP packets is expensive at very high speeds. Specifically, at speeds of OC-3 and above, the cost of segmentation and reassembly (SAR) hardware makes ATM less competitive for IP than Packet Over SONET (POS); because of its fixed 48-byte cell payload, ATM is not suitable as a data link layer directly underlying IP (without the need for SAR at the data link level) since the OSI layer on which IP operates must provide a maximum transmission unit (MTU) of at least 576 bytes. SAR performance limits mean that the fastest IP router ATM interfaces are STM16 - STM64 which actually compares, while as of 2004 POS can operate at OC-192 (STM64) with higher speeds expected in the future.

On slower or congested links (622 Mbit/s and below), ATM does make sense, and for this reason most asymmetric digital subscriber line (ADSL) systems use ATM as an intermediate layer between the physical link layer and a Layer 2 protocol like PPP or Ethernet.

At these lower speeds, ATM provides a useful ability to carry multiple logical circuits on a single physical or virtual medium, although other techniques exist, such as Multi-link PPP and Ethernet VLANs, which are optional in VDSL implementations. DSL can be used as an access method for an ATM network, allowing a DSL termination point in a telephone central office to connect to many internet service providers across a wide-area ATM network. In the United States, at least, this has allowed DSL providers to provide DSL access to the customers of many internet service providers. Since one DSL termination point can support multiple ISPs, the economic feasibility of DSL is substantially improved.

ATM operates as a channel-based transport layer, using virtual circuits (VCs). This is encompassed in the concept of the Virtual Paths (VP) and Virtual Channels. Every ATM cell has an 8- or 12-bit Virtual Path Identifier (VPI) and 16-bit Virtual Channel Identifier (VCI) pair defined in its header. The VCI, together with the VPI, is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. The length of the VPI varies according to whether the cell is sent on the user-network interface (on the edge of the network), or if it is sent on the network-network interface (inside the network).

As these cells traverse an ATM network, switching takes place by changing the VPI/VCI values (label swapping). Although the VPI/VCI values are not necessarily consistent from one end of the connection to the other, the concept of a circuit is consistent (unlike IP, where any given packet could get to its destination by a different route than the others). ATM switches use the VPI/VCI fields to identify the

Virtual Channel Link (VCL) of the next network that a cell needs to transit on its way to its final destination. The function of the VCI is similar to that of the data link connection identifier (DLCI) in frame relay and the Logical Channel Number & Logical Channel Group Number in X.25.

Another advantage of the use of virtual circuits comes with the ability to use them as a multiplexing layer, allowing different services (such as voice, Frame Relay, $n \times 64 channels$, IP). The VPI is useful for reducing the switching table of some virtual circuits which have common paths.

Using cells and virtual circuits for traffic engineering

Another key ATM concept involves the traffic contract. When an ATM circuit is set up each switch on the circuit is informed of the traffic class of the connection.

ATM traffic contracts form part of the mechanism by which "quality of service" (QoS) is ensured. There are four basic types (and several variants) which each have a set of parameters describing the connection.

1. CBR - Constant bit rate: a Peak Cell Rate (PCR) is specified, which is constant
2. VBR - Variable bit rate: an average or Sustainable Cell Rate (SCR) is specified, which can peak at a certain level, a PCR, for a maximum interval before being problematic
3. ABR - Available bit rate: a minimum guaranteed rate is specified
4. UBR - Unspecified bit rate: traffic is allocated to all remaining transmission capacity

VBR has real-time and non-real-time variants, and serves for "bursty" traffic. Non-real-time is sometimes abbreviated to vbr-nrt

Most traffic classes also introduce the concept of Cell Delay Variation Tolerance (CDVT), which defines the "clumping" of cells in time.

Traffic policing

To maintain network performance, networks may apply traffic policing to virtual circuits to limit them to their traffic contracts at the entry points to the network, i.e. the user–network interfaces (UNIs) and network-to-network interfaces (NNIs): Usage/Network Parameter Control (UPC and NPC).[9] The reference model given by the ITU-T and ATM Forum for UPC and NPC is the generic cell rate algorithm (GCRA),[10][11] which is a version of the leaky bucket algorithm. CBR traffic will normally be policed to a PCR and CDVt alone, whereas VBR traffic will normally be policed using a dual leaky bucket controller to a PCR and CDVt and an SCR and Maximum Burst Size (MBS). The MBS will normally be the packet (SAR-SDU) size for the VBR VC in cells.

If the traffic on a virtual circuit is exceeding its traffic contract, as determined by the GCRA, the network can either drop the cells or mark the Cell Loss Priority (CLP) bit (to identify a cell as potentially redundant). Basic policing works on a cell by cell basis, but this is sub-optimal for encapsulated packet traffic (as discarding a single cell will invalidate the whole packet). As a result, schemes such as Partial Packet Discard (PPD) and Early Packet Discard (EPD) have been created that will discard a whole series of cells until the next packet starts. This reduces the number of useless cells in the network, saving bandwidth for full packets. EPD and PPD work with AAL5 connections as they use the end of packet marker: the ATM User-to-ATM User (AUU) Indication bit in the Payload Type field of the header, which is set in the last cell of a SAR-SDU.

Traffic shaping

Traffic shaping usually takes place in the network interface card (NIC) in user equipment, and attempts to ensure that the cell flow on a VC will meet its traffic contract, i.e. cells will not be dropped or reduced in priority at the UNI. Since the reference model given for traffic policing in the network is the GCRA, this algorithm is normally used for shaping as well, and single and dual leaky bucket implementations may be used as appropriate.

Types of virtual circuits and paths

ATM can build virtual circuits and virtual paths either statically or dynamically. Static circuits (permanent virtual circuits or PVCs) or paths (permanent virtual paths or PVPs) require that the circuit is composed of a series of segments, one for each pair of interfaces through which it passes.

PVPs and PVCs, though conceptually simple, require significant effort in large networks. They also do not support the re-routing of service in the event of a failure. Dynamically built PVPs (soft PVPs or SPVPs) and PVCs (soft PVCs or SPVCs), in contrast, are built by specifying the characteristics of the circuit (the service "contract") and the two end points.

Finally, ATM networks create and remove switched virtual circuits (SVCs) on demand when requested by an end piece of equipment. One application for SVCs is to carry individual telephone calls when a network of telephone switches are inter-connected using ATM. SVCs were also used in attempts to replace local area networks with ATM.

Virtual circuit routing

Most ATM networks supporting SPVPs, SPVCs, and SVCs use the Private Network Node Interface or the Private Network-to-Network Interface (PNNI) protocol. PNNI uses the same shortest-path-first algorithm used by OSPF and IS-IS to route IP packets to share topology information between switches and select a route through a network. PNNI also includes a very powerful summarization mechanism to allow construction of very large networks, as well as a call admission control (CAC) algorithm which determines the availability of sufficient bandwidth on a proposed route through a network in order to satisfy the service requirements of a VC or VP.

Call admission and connection establishment

A network must establish a connection before two parties can send cells to each other. In ATM this is called a virtual circuit (VC). It can be a permanent virtual circuit (PVC), which is created administratively on the end points, or a switched virtual circuit (SVC), which is created as needed by the communicating parties. SVC creation is managed by signaling, in which the requesting party indicates the address of the receiving party, the type of service requested, and whatever traffic parameters may be applicable to the selected service. "Call admission" is then performed by the network to confirm that the requested resources are available and that a route exists for the connection.

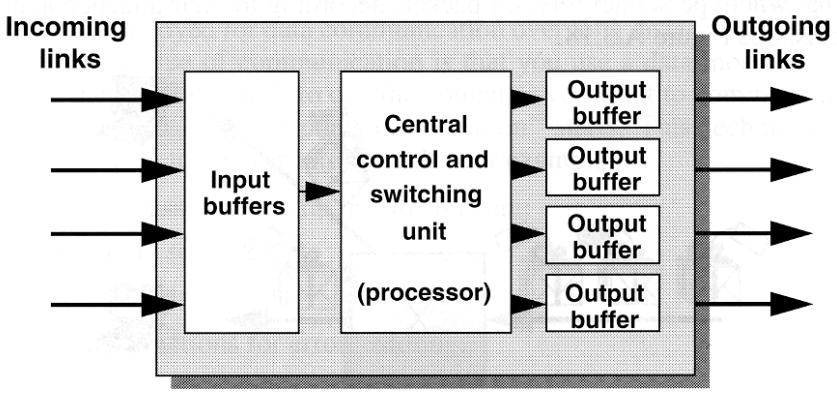


Figure 21: ATM packet node structure

- ATM cell switching principle
- Fixed cell (packet) length - 53 bytes
- 5 octets header, 48 octet payload

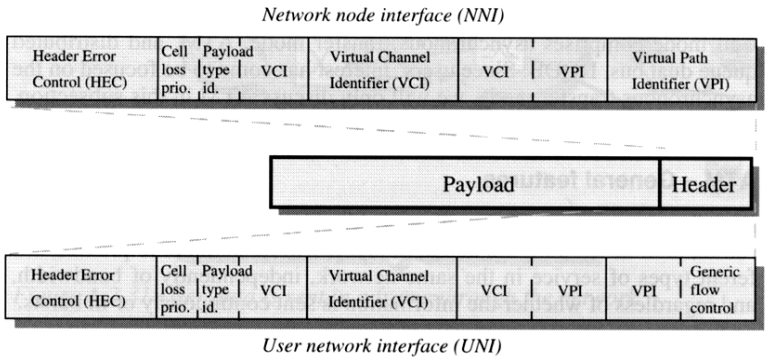


Figure 22: header

A **Virtual Channel (VC)** denotes the transport of ATM cells which have the same unique identifier, called the Virtual Channel Identifier (VCI). This identifier is encoded in the cell header.
A **Virtual Path (VP)** denotes the transport of ATM cells belonging to virtual channels which share a common identifier, called the Virtual Path Identifier (VPI), which is also encoded in the cell header. A virtual path, in other words, is a grouping of virtual channels which connect the same end-points. This two layer approach results in improved network performance. Once a virtual path is set up, the addition/removal of virtual channels is straightforward.

3.6 6. topic

Description: Main elements of a GSM network (MSC, BSC, BTS, HLR, VLR, LA, MS. . . ..)

The main components of a GSM network are the followings:

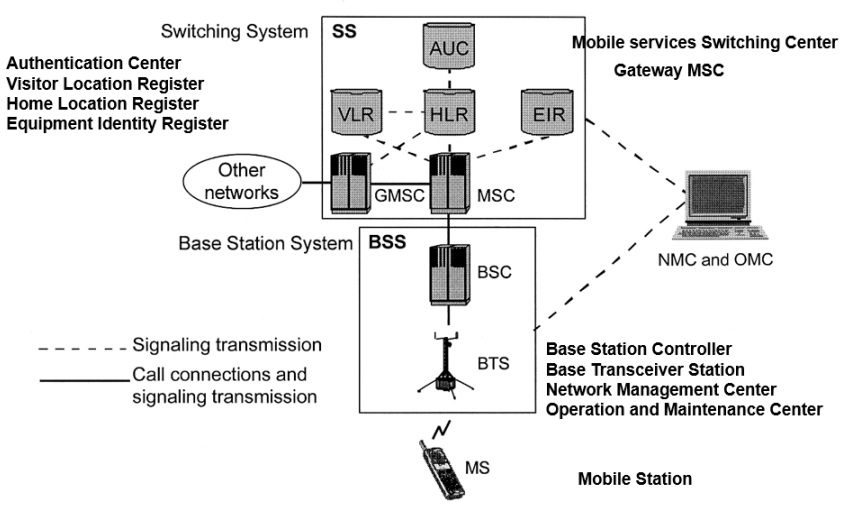


Figure 23: GSM network components

Mobil station

- Used by mobile subscriber to communicate with the network
- Consist of mobile terminal and Subscriber Identity Module (SIM)
- Subscription is separated from the mobile terminal
- Subscription information is stored in a ”smart card”
- Hand-held MS, Car-installed MS

The possible states of MS

- Idle: the MS is ON but a call is not in progress
- Active: The MS is ON and a call is in progress
- Detached: The MS is OFF

Idle key terms

- Registration: MS informs a network that it is attached
- Roaming: MS moves around the network in idle mode
- International Roaming: MS moves into a network which is not its home network
- Location Updating: MS inform the network when enters in new LA
- Locating: BSC function to suggest connection to another cell based on MS measurement reports
- Paging: The network tries to contact an MS by broadcasting message containing MS identity

Active key terms

- Handover: Process, where a call is switched from one physical channel to another, while MS moves around

MS registration

- MS power ON
- MS scans for control channel frequencies
- MS measures signal levels and records it
- MS tunes to the strongest frequency
- MS register to the network
- Network update the MS status to idle
- Network store location information

MS roaming

The idle MS moves thorough the network, scan the control channels, tune to the strongest channel, in new LA inform the network of its new location.

The actions of the network during roaming and handover

Mobile Station Roaming Number (MSRN)

- CC Country Code (36 for Hungary)
- NDC National Destination Code (20 for Telenor)
- SN service Node

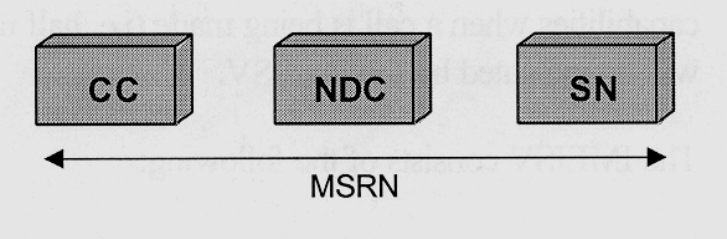


Figure 24: Mobile Station Roaming Number (MSRN)

Basic Handover

- BSC send handover-required message to the MSC
- The MSC ask the target MSC to assist. The Target MSC allocates a handover number that reroutes the call
- A handover request is sent down to the new BSC
- The BSC tells the new BTS to activate a TCH
- The MSC receives the information about the new Traffic CHannel

- The MSC passes info on new TCH from new BSC
- A speech path to the new MSC is set up
- A handover command goes to the MS with frequency and time slot data in the new cell
- The MS sends handover burst on the new TCH
- The target MSC is informed that the handover successful
- A new path in the Group Switch is set up

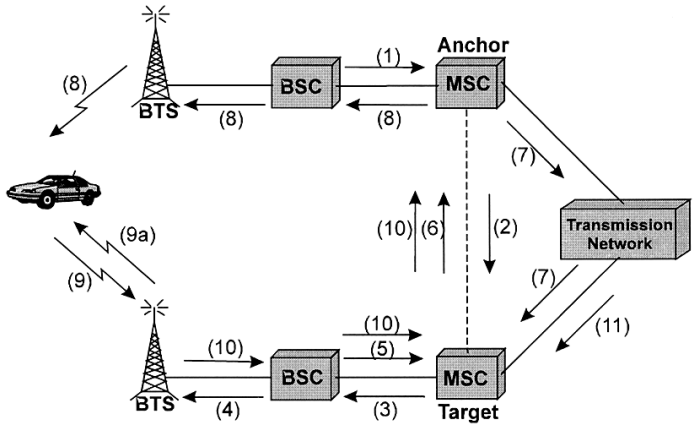


Figure 25: Basic Handover

The functions of HLR and VLR in the GSM network

Home Location Register (HLR):

Centralized network database for

- Subscriber identity
- Subscriber supplementary services
- Subscriber location information
- Subscriber authentication information

Visitor Location Register (VLR):

Information about subscribers located in an MSC service area (a copy of HLR information)

Functions of MSC and BSC in the GSM network

Station Controller (BSC):

- Manages all the radio related functions of the network
- MS handover
- Radio channel assignment
- Collection of cell configuration data
- Controlled by MSC

Mobile Switching Center:

- Billing
- Delivers SMSs from subscribers to SMSC
- Arranges handovers
- supplementary services
- Controls BSC

Steps to call MS

1. Call entering to GSM network is routed to the nearest GMSC
2. The GSM analyse the MSISDN to find the HLR (subscriber registered in) The MSC/VLR address is stored in HLR, the IMSI is stored in HLR
3. The HLR send request to an MSRN to the MSC/VLR included in the message the IMSI
4. The MSRN is returned via HLR to the GMSC
5. The GMSC routes the call to the MSC/VLR by MSRN
6. The MSC/VLR retrieve the Ms's IMSI
7. Using IMSI MSC identifies LA
8. The MS is paged in cells in the LA
9. MS responds, authentication, cipher mode setting, IMEI check are carried out
10. Traffic channel connected from MSC to BSC and the BTS

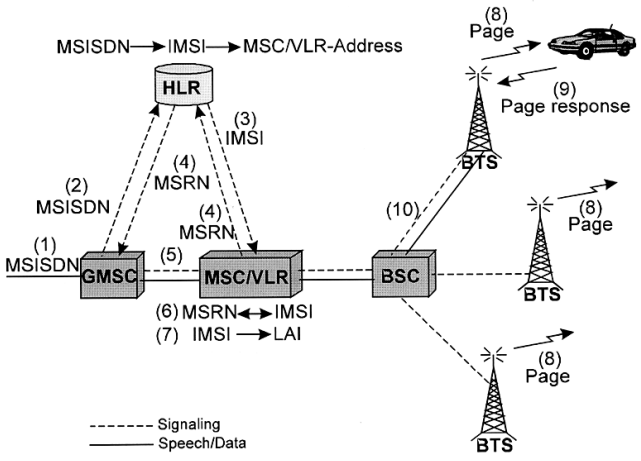


Figure 26: Call to an MS

Steps to receive call from MS

1. Call start with a signalling channel using RACH (Random Access Channel)
2. MS indicates request, IMSI is analyzed, MS marked busy in The VLR
3. Authentication is performed by MSC
4. Ciphering is initiated, IMEI validated
5. MSC receives a setup message from MS (including B number)
6. Link established between MSC and BSC to assign traffic channel
7. Call confirmation
8. Call accepted

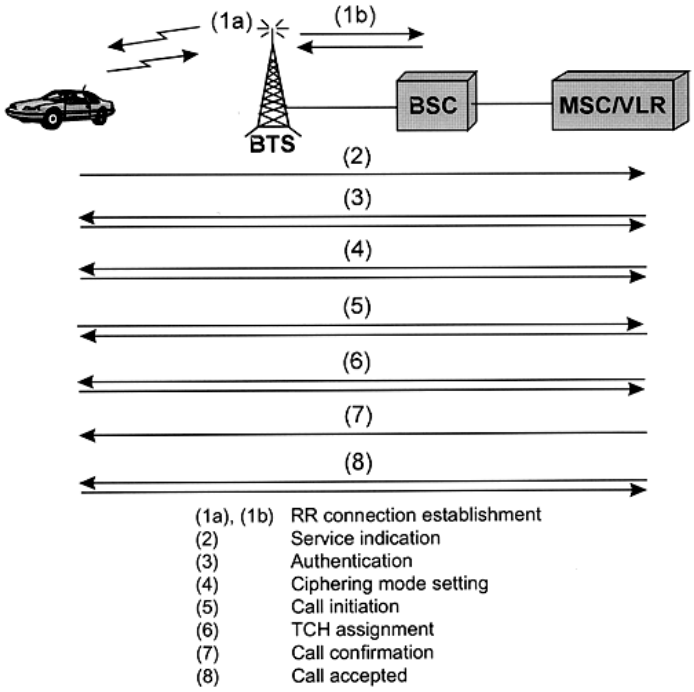


Figure 27: Call from MS

Supported in GSM system

GSM specification items

- Voice oriented services
- Separation of terminal and subscription
- Europe-wide international roaming
- Low bit-rate speech coding
- High bandwidth utilisation
- Low power consumption in inactive mode

- Standards for system concept and air interface
- No direct call number information on air interface
- Encrypted speech coding on air interface
- Authentication process
- Handover up to 200km/h (car-phone or hand-held in train)
- Outdoor and indoor coverage

3.7 7. topic

Description: Basic services, supplementary services, service quality requirements in different services

---

Basic and supplementary services

Teleservices from provider point of view:

- Basic services (mandatory service elements with minimal quality requirements e.g. real time, understandable...)
- Supplementary services (to make basic services even more usable, e.g. call transfer, conference call, automatic call back on busy, wake up services, least cost routing services, credit card based call...
- Value added services (e.g. bank transaction by phone, televoting, telephone based donation...)

The main service quality requirements in different services

Network and terminal requirements:

- Voice, music, video
  - sensitive on delay (max. 300 ms)
  - sensitive on jitter (max 30 ms)
  - sensitive on video/voice synchrony (lip-sync)
  - error tolerant, (bit error rate 10-3 acceptable!!!)
- Games
  - sensitive on delay (max. 10 ms)
  - sensitive on error
- Data, still picture
  - sensitive on error (BER min. 10-6 , error control)
  - delay and jitter tolerant (www=world wide waiting)

3.8 8. topic

Description: Digital modulation systems (BPSK, QPSK, QAM)

Amplitude Shift Keying (ASK)

Amplitude-shift keying (ASK) is a form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave. In an ASK system, the binary symbol 1 is represented by transmitting a fixed-amplitude carrier wave and fixed frequency for a bit duration of T seconds. If the signal value is 1 then the carrier signal will be transmitted; otherwise, a signal value of 0 will be transmitted.

Different symbols are represented with different voltages. If the maximum allowed value for the voltage is A, then all the possible values are in the range $[-A, A]$ and they are given by:

$$v_i = \frac{2A}{L-1} i - A,$$

for $i = 0, 1, ..., L-1$. Hence the difference between one voltage and the other is:

$$\Delta = \frac{2A}{L-1}.$$

The symbols v[n] are generated randomly by an IT source S, then an impulse generator creates impulses with an area of $v[n]$. These impulses are sent to the filter h_t to be sent through the channel. In other words, for each symbol a different carrier wave is sent with the relative amplitude.

Out of the transmitter, the signal $s(t)$ can be expressed in the form:

$$s(t) = \sum_{n=-\infty}^{\infty} v[n] h_t(t - nT_s).$$

In the receiver, after the filtering through $h_r(t)$ the signal is:

$$z(t) = n_r(t) + sum_{n=-\infty}^{\infty} v[n] g(t - nT_s),$$

where we use the notation:

$$n_r(t) = n(t) * h_r(f)$$
$$g(t) = h_t(t) * h_c(f) * h_r(t),$$

where $*$ indicates the convolution between two signals. After the A/D conversion the signal $z[k]$ can be expressed in the form:

$$z[k] = n_r[k] + v[k]g[0] + \sum_{n \neq k} v[n] g[k-n].$$

In this relationship, the second term represents the symbol to be extracted. The others are unwanted: the first one is the effect of noise, the third one is due to the inter-symbol interference.

If the filters are chosen so that $g(t)$ will satisfy the Nyquist ISI criterion, then there will be no inter-symbol interference and the value of the sum will be zero, so:

$$z[k] = n_r[k] + v[k]g[0],$$

so the transmission will be affected only by noise.

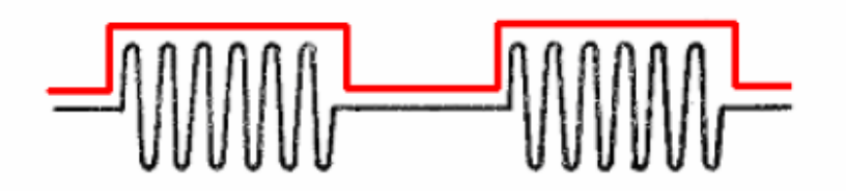


Figure 28: Amplitude Shift Keying (ASK)

Probability of error

The probability density function of having an error of a given size can be modelled by a Gaussian function; the mean value will be the relative sent value, and its variance will be given by:

$$\sigma_N^2 = \int_{-\infty}^{\infty} \Phi_N(f)|H_r(f)|^2 df,$$

where $Phi_N(f)$ is the spectral density of the noise within the band and $H_r(f)$ is the continuous Fourier transform of the impulse response of the filter $h_r(f)$.

The probability of making an error is given by the law of total probability:

$$P_e = \sum_{i=0}^{L-1} P(error|H_i) * P(H_i),$$

where $P(error|H_i)$ is the conditional probability of making an error given that a symbol v_i has been sent and $P(H_i)$ is the probability of sending symbol v_i.

If the latter follows uniform distribution, then

$$P(H_i) = \frac{1}{L}.$$

The probability of making an error after a single symbol has been sent is the area of the Gaussian function falling under the functions for the other symbols. The difference between these symbols mean is $\frac{2Ag(0)}{L-1}$.

If we call P^+ the area under one side of the Gaussian, the sum of all the areas will be: $2LP^+ - 2P^+$. The total probability of making an error can be expressed in the form:

$$P_e = 2\left(1 - \frac{1}{L}\right)P^+.$$

We have now to calculate the value of P+. In order to do that, we can move the origin of the reference wherever we want: the area below the function will not change.

It does not matter which Gaussian function we are considering, the area we want to calculate will be the same. The value we are looking for will be given by the following integral:

$$P^+ = \int_{\frac{Ag(0)}{L-1}}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_N} e^{-\frac{x^2}{2\sigma_N^2}} dx = \frac{1}{2} erfc\left(\frac{Ag(0)}{\sqrt{2}(L-1)\sigma_N}\right),$$

where $erfc(c)$ is the complementary error function. Putting all these results together, the probability to make an error is:

$$P_e = 2\left(1 - \frac{1}{L}\right)\frac{1}{2} erfc\left(\frac{Ag(0)}{\sqrt{2}(L-1)\sigma_N}\right).$$

From this formula we can easily understand that the probability to make an error decreases if the maximum amplitude of the transmitted signal or the amplification of the system becomes greater; on the other hand, it increases if the number of levels or the power of noise becomes greater.

This relationship is valid when there is no intersymbol interference, i.e. $g(t)$ is a Nyquist function.

Binary Phase Shift Keying (BPSK)

Phase-shift keying (PSK) is a digital modulation scheme that conveys data by changing (modulating) the phase of a reference signal (the carrier wave).

The modulation is impressed by varying the sine and cosine inputs at a precise time. It is widely used for wireless LANs, RFID and Bluetooth communication.

BPSK (also sometimes called PRK, phase reversal keying, or 2PSK) is the simplest form of phase shift keying (PSK). It uses two phases which are separated by 180°and so can also be termed 2-PSK. It does not particularly matter exactly where the constellation points are positioned, and in this figure they are shown on the real axis, at 0°and 180°. This modulation is the most robust of all the PSKs since it takes the highest level of noise or distortion to make the demodulator reach an incorrect decision. It is, however, only able to modulate at 1 bit/symbol (as seen in the figure) and so is unsuitable for high data-rate applications.

In the presence of an arbitrary phase-shift introduced by the communications channel, the demodulator is unable to tell which constellation point is which. As a result, the data is often differentially encoded prior to modulation.

BPSK is functionally equivalent to 2-QAM modulation.

The general form of BPSK follows the equation (for the signal):

$$s_n(t) = \sqrt{\frac{2E_b}{T_b}} cos\big(2\pi f_c t + \pi(1-n)\big),$$

for $n = 0, 1$. This yield two phases, 0 and π. In the specific form, binary data is often conveyed with the following signals:

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}} cos\big(2\pi f_c t + \pi\big),$$

for binary 0, and

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} cos\big(2\pi f_c t\big),$$

for binary 1. Note that f_c is the frequency of the carrier-wave.

Hence, the signal-space can be represented by the single basis function:

$$\varphi(t) = \sqrt{\frac{2}{T_b}} cos(2\pi f_c t),$$

where 1 is represented by $\sqrt{E_b}\varphi(t)$, and 0 is represented by $-\sqrt{E_b}\varphi(t)$. This assignment is, of course, arbitrary.

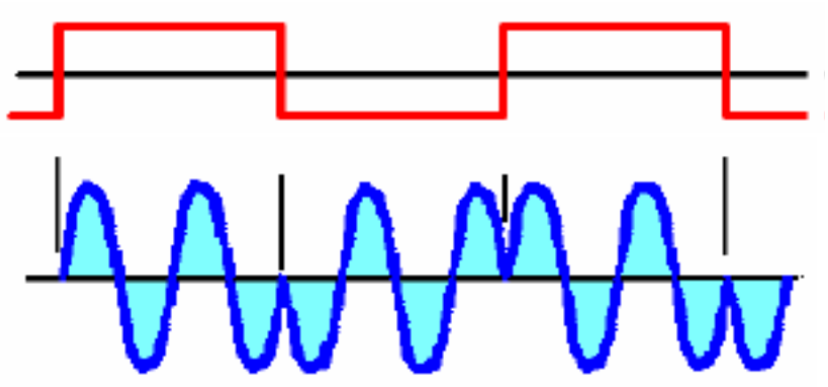


Figure 29: Binary Phase Shift Keying (BPSK)

Probability of error

The bit error rate (BER) of BPSK in AWGN channels can be calculated as:

$$P_b = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right),$$

where E_b is the energy per bit value, while N_0 is the power spectral density of the noise.

Since there is only one bit per symbol, this is also the symbol error rate as well.

Qadrature Phase Shift Keying (QPSK)

Sometimes this is known as quadriphase PSK, 4-PSK, or 4-QAM. (Although the root concepts of QPSK and 4-QAM are different, the resulting modulated radio waves are exactly the same.) QPSK uses four points on the constellation diagram, equispaced around a circle. With four phases, QPSK can encode two bits per symbol, shown in the diagram with Gray coding to minimize the bit error rate (BER) — sometimes misperceived as twice the BER of BPSK.

The mathematical analysis shows that QPSK can be used either to double the data rate compared with a BPSK system while maintaining the same bandwidth of the signal, or to maintain the data-rate of BPSK but halving the bandwidth needed. In this latter case, the BER of QPSK is exactly the same as the BER of BPSK - and deciding differently is a common confusion when considering or describing QPSK. The transmitted carrier can undergo numbers of phase changes.

Given that radio communication channels are allocated by agencies such as the Federal Communication Commission giving a prescribed (maximum) bandwidth, the advantage of QPSK over BPSK becomes evident: QPSK transmits twice the data rate in a given bandwidth compared to BPSK - at the same BER. The engineering penalty that is paid is that QPSK transmitters and receivers are more complicated than the ones for BPSK. However, with modern electronics technology, the penalty in cost is very moderate.

As with BPSK, there are phase ambiguity problems at the receiving end, and differentially encoded QPSK is often used in practice.

The implementation of QPSK is more general than that of BPSK and also indicates the implementation of higher-order PSK. Writing the symbols in the constellation diagram in terms of the sine and cosine waves used to transmit them:

$$s_n(t) = \sqrt{\frac{2E_s}{T_s}} cos\big(2\pi f_c t + (2n-1)\frac{\pi}{4}\big),$$

for $n = 1, 2, 3, 4$. This yields the four phases $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$ and $\frac{7\pi}{4}$ as needed.

This results in a two dimensional signal space with unit basis functions:

$$\varphi_1(t) = \sqrt{\frac{2}{T_2}} cos(2\pi f_c t)$$
$$\varphi_2(t) = \sqrt{\frac{2}{T_2}} sin(2\pi f_c t).$$

The first basis function is used as the in-phase component of the signal and the second as the quadrature component of the signal.

Hence, the signal constellation consists of the signal-space 4 points:

$$\left(\pm \sqrt{\frac{E_s}{2}}, \pm \sqrt{\frac{E_s}{2}} \right).$$

The factors of 1/2 indicate that the total power is split equally between the two carriers.

Comparing these basis functions with that for BPSK shows clearly how QPSK can be viewed as two independent BPSK signals. Note that the signal-space points for BPSK do not need to split the symbol (bit) energy over the two carriers in the scheme shown in the BPSK constellation diagram.

Summarize the above statements:

- Two carriers: sine wave (Q) and cosine wave (I)
- The modulated signal is the sum of the two components
- One symbol is two bits

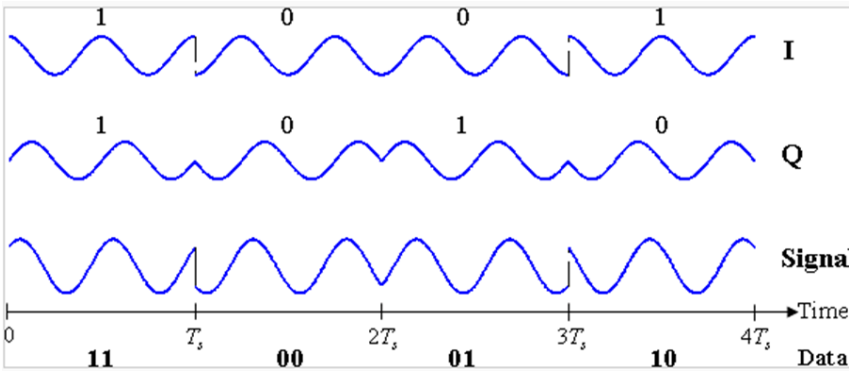


Figure 30: Qadrature Phase Shift Keying (QPSK)

Probability of error

Although QPSK can be viewed as a quaternary modulation, it is easier to see it as two independently modulated quadrature carriers. With this interpretation, the even (or odd) bits are used to modulate the in-phase component of the carrier, while the odd (or even) bits are used to modulate the quadrature-phase component of the carrier. BPSK is used on both carriers and they can be independently demodulated.

As a result, the probability of bit-error for QPSK is the same as for BPSK:

$$P_b = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right).$$

However, in order to achieve the same bit-error probability as BPSK, QPSK uses twice the power (since two bits are transmitted simultaneously).

The symbol error rate is given by:

$$P_s = 1 - (1 - P_b)^2 = 2Q\left(\sqrt{\frac{E_b}{N_0}}\right) - Q^2\left(\sqrt{\frac{E_b}{N_0}}\right).$$

If the SNR is high (as is necessary for practical QPSK systems) the probability of symbol error may be approximated:

$$P_s \approx 2Q\left(\sqrt{\frac{E_b}{N_0}}\right).$$

Qadrature Amplitude Modulation (QAM)

Quadrature amplitude modulation (QAM) is both an analog and a digital modulation scheme. It conveys two analog message signals, or two digital bit streams, by changing (modulating) the amplitudes of two carrier waves, using the amplitude-shift keying (ASK) digital modulation scheme or amplitude modulation (AM) analog modulation scheme.

The two carrier waves of the same frequency, usually sinusoids, are out of phase with each other by 90°and are thus called quadrature carriers or quadrature components – hence the name of the scheme.

The modulated waves are summed, and the final waveform is a combination of both phase-shift keying (PSK) and amplitude-shift keying (ASK), or, in the analog case, of phase modulation (PM) and amplitude modulation.

In the digital QAM case, a finite number of at least two phases and at least two amplitudes are used. PSK modulators are often designed using the QAM principle, but are not considered as QAM since the amplitude of the modulated carrier signal is constant. QAM is used extensively as a modulation scheme for digital telecommunication systems. Arbitrarily high spectral efficiencies can be achieved with QAM by setting a suitable constellation size, limited only by the noise level and linearity of the communications channel.

QAM is being used in optical fiber systems as bit rates increase; QAM16 and QAM64 can be optically emulated with a 3-path interferometer.

When transmitting two signals by modulating them with QAM, the transmitted signal will be of the form:

$$s(t) = \mathbb{R}\Big\{ \big(I(t) + iQ(t)\big)e^{i2\pi f_0 t} \Big\} = I(t)cos(2\pi f_0 t) - Q(t)sin(2\pi f_0 t).$$

where $I(t)$ and $Q(t)$ are the modulating signals, f_0 is the carrier frequency.

At the receiver, these two modulating signals can be demodulated using a coherent demodulator. Such a receiver multiplies the received signal separately with both a cosine and sine signal to produce the received estimates of $I(t)$ and $Q(t)$ respectively. Because of the orthogonality property of the carrier signals, it is possible to detect the modulating signals independently.

In the ideal case $I(t)$ is demodulated by multiplying the transmitted signal with a cosine signal:

$$r(t) = s(t)cos(2\pi f_0 t) = I(t)cos(2\pi f_0 t)cos(2\pi f_0 t) - Q(t)sin(2\pi f_0 t)cos(2\pi f_0 t).$$

Using standard trigonometric identities, we can write it as:

$$\begin{aligned} r(t) &= \frac{1}{2}I(t)\big(1 + cos(4\pi f_0 t)\big) - \frac{1}{2}Q(t)sin(4\pi f_0 t) = \\ &= \frac{1}{2}I(t) + \frac{1}{2}\big(I(t)cos(4\pi f_0 t) - Q(t)sin(4\pi f_0 t)\big). \end{aligned}$$

Low-pass filtering $r(t)$ removes the high frequency terms (containing $4\pi f_0 t$), leaving only the $I(t)$ term. This filtered signal is unaffected by $Q(t)$, showing that the in-phase component can be received independently of the quadrature component. Similarly, we may mulitply $s(t)$ by a sine wave, and then low-pass filter to extract $Q(t)$.

Analog QAM suffers from the same problem as Single-sideband modulation: the exact phase of the carrier is required for correct demodulation at the receiver. If the demodulating phase is even a little off, it results in crosstalk between the modulated signals. This issue of carrier synchronization at the receiver must be handled somehow in QAM systems. The coherent demodulator needs to be exactly in phase with the received signal, or otherwise the modulated signals cannot be independently received. This is achieved typically by transmitting a burst subcarrier or a Pilot signal.

- Two carriers: sine wave (Q) and cosine wave (I)
- The modulated signal is the sum of the two components
- Different amplitude and differnt phase values for one symbol
- 16QAM means: one symbol is four bits

Probability of error

The following definitions are needed in determining error rates:

1. M = Number of symbols in modulation constellation
2. E_b = Energy-per-bit
3. E_s = Energy-per-symbol = kE_b with k bits per symbol
4. N_0 = Noise power spectral density (W/Hz)
5. P_b = Probability of bit-error

6. P_{bc} = Probability of bit-error per carrier
7. P_s = Probability of symbol-error
8. P_{sc} = Probability of symbol-error per carrier
9. $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{1}{2}t^2} dt,\ x \geq 0$

$Q(x)$ is related to the complementary Gaussian error function by

$$Q(x) = \frac{1}{2} erfc\left(\frac{1}{\sqrt{2}}\right),$$

which is the probability that x will be under the tail of the Gaussian PDF towards positive infinity.

The error rates quoted here are thos in AWGN channels.

Where coordinates for constellation points are given in this section, note that they represent a non-normalised constellation. That is, if a particular mean average energy were required (e.g. unit average energy), the constellation would need to be linearly scaled.

Probability of error of rectangular QAM

Rectangular QAM constellations are, in general, sub-optimal in the sense that they do not maximally space the constellation points for a given energy. However, they have the considerable advantage that they may be easily transmitted as two pulse amplitude modulation (PAM) signals on quadrature carriers, and can be easily demodulated. The non-square constellations, dealt with below, achieve marginally better bit-error rate (BER) but are harder to modulate and demodulate.

The first rectangular QAM constellation usually encountered is 16-QAM, the constellation diagram for which is shown here. A Gray coded bit-assignment is also given. The reason that 16-QAM is usually the first is that a brief consideration reveals that 2-QAM and 4-QAM are in fact binary phase-shift keying (BPSK) and quadrature phase-shift keying (QPSK), respectively. Also, the error-rate performance of 8-QAM is close to that of 16-QAM (only about 0.5 dB better[citation needed]), but its data rate is only three-quarters that of 16-QAM.

Expressions for the symbol-error rate of rectangular QAM are not hard to derive but yield rather unpleasant expressions. For an even number of bits per symbol, k, exact expressions are available. They are most easily expressed in a per carrier sense:

$$P_{sc} = 2\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3E_s}{(M-1)N_0}}\right),$$

so

$$P_s = 1 - (1 - P_{sc})^2.$$

The bit-error rate depends on the bit to symbol mapping, but for $\frac{E_b}{N_0} \gg 1$ and a Gray-coded assignment – so that we can assume each symbol error causes only one bit error – the bit-error rate is approximately

$$P_{bc} \approx \frac{P_{sc}}{\frac{1}{2}k} = \frac{4}{k}\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3E_s}{(M-1)N_0}}\right).$$

Since the carriers are independent, the overall bit error rate is the same as the per-carrier error rate, just like BPSK and QPSK, so $P_b = P_{bc}$.

An exact and general closed-form expression of the Bit Error Rates (BER) for rectangular type of Quadrature Amplitude Modulation (QAM) over AWGN and slow, flat, Rician fading channels were derived analytically.

Consider a $(LM) - QAM$ system with $2\log_2(L)$ levels and $2\log_2(M)$ levels in the I-channel and Q-channel, respectively and a two-dimensional grey code mapping employed. It was shown that the generalized expression for the conditional BER on SNR ρ over AWGN channel is

$$P_b(error|\rho) = \frac{1}{\log_2(LM)}\left(\sum_{i=1}^{\log_2(L)} P_b(E_i^L|\rho) + \sum_{i=1}^{\log_2(M)} P_b(E_i^M|\rho)\right),$$

where

$$P_b(E_i^P|\rho) = \frac{2}{P}\sum_{j=0}^{(1-2^{-i}P-1} (-1)^{\left\lfloor \frac{j2^{i-1}}{P} \right\rfloor} \left(2^{i-1} - \left\lfloor \frac{j2^{i-1}}{P} + \frac{1}{2} \right\rfloor\right) Q\left((2j+1)\sqrt{\frac{6\rho}{L^2+M^2-2}}\right).$$

For odd k, such as 8-QAM ($k = 3$) it is harder to obtain symbol-error rates but a tight upper bound is

$$P_s \leq 4Q\left(\sqrt{\frac{3kE_b}{(M-1)N_0}}\right).$$

Beside the rectangular QAM, there exists other QAM algorithms, which are not rectangular, but this is out of the scope of this work.

3.9 9. topic

Description: Typical structures and technologies in PSTN networks

---

The public switched telephone network (PSTN) is the aggregate of the world's circuit-switched telephone networks that are operated by national, regional, or local telephony operators, providing infrastructure and services for public telecommunication. The PSTN consists of telephone lines, fiber optic cables, microwave transmission links, cellular networks, communications satellites, and undersea telephone cables, all interconnected by switching centers, thus allowing most telephones to communicate with each other. Originally a network of fixed-line analog telephone systems, the PSTN is now almost entirely digital in its core network and includes mobile and other networks, as well as fixed telephones.

We divide the followings into 6 parts, which are:

1. Brief history
2. Basics
3. Network structures
4. Network implementation
5. Network development trends
6. Missing topic

Brief history

1. 1876 A. G. Bell Patent of telephone (50.000 phone within 3 years)
2. 1877 T. A. Edison patent of carbon microphone (covering long distances)
3. 1878 Puskás Tivadar the first telephone exchange in Connecticut
4. 1881 Puskás Ferenc the first telephone exchange in Budapest
5. 1890 The first Wien-Budapest international telephone connection
6. 1892 The first automatic telephone exchange in Indiana
7. 1928 The first automatic telephone exchange in Budapest
8. 1997 The last manual exchange moved to museum in Hungary
9. The technical level of Hungarian PSTN networks are very high due to the fast development from 1992-2002

Basics

The following principles are the basic of telephony:

1. 2/4 wire for voice
2. feeding of circuit
3. access solutions
4. backbone
5. signalling basics for a telephone call
6. source of revenues

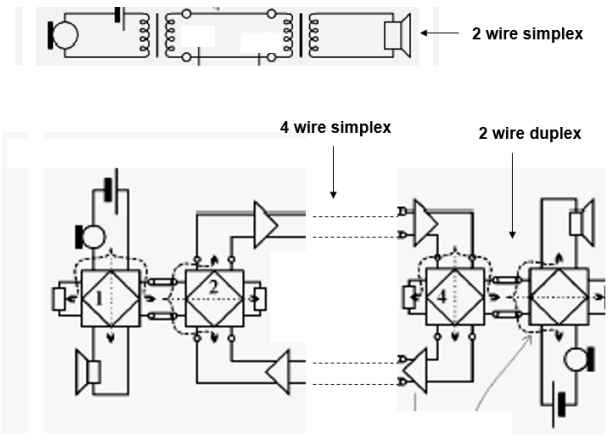


Figure 31: 2/4 wire for voice

7. ADSL principles

2/4 wire

Feeding of circuits
With circuit feeding, we can get a DC line current, which is from $20mA$ to $60mA$.

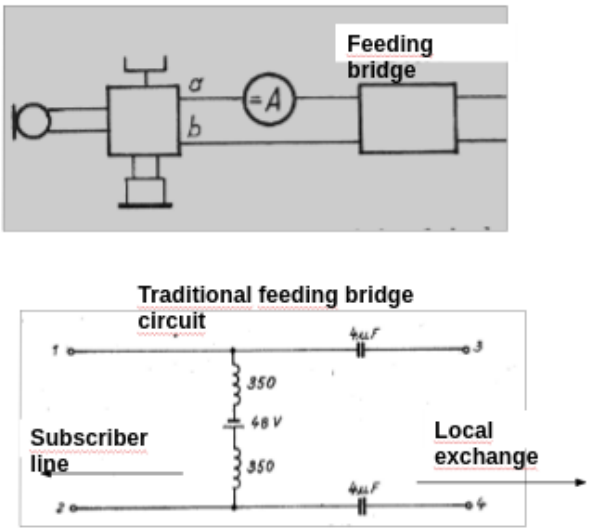


Figure 32: Circuit feeding

Access solutions

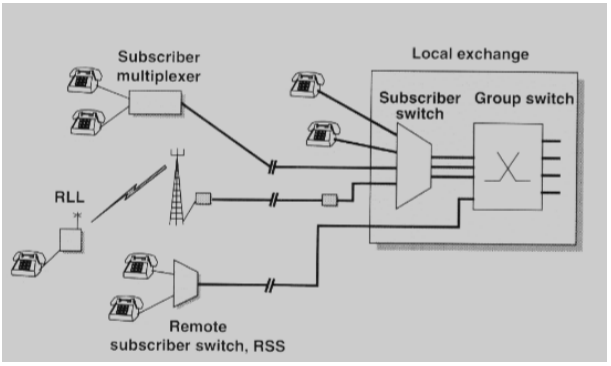


Figure 33: Access solutions

Backbone

A backbone is a part of computer network that interconnects various pieces of network, providing a path for the exchange of information between different LANs or sub-networks. A backbone can tie together diverse networks in the same building, in different buildings in a campus environment, or over wide areas. Normally, the backbone's capacity is greater than the networks connected to it.

In a small country with dense population the backbone has a some issues. The backbone is fully optical, only spare radio connections. The covered distances between nodes have an upper bound of 100 km.

In Hungary the source of telecom traffic (including telephone, TV program, internet) conectrated in Budapest (like the road traffic).

Concentration of switching capacities in higher level nodes.

Fault tolerant topology is required for reliable services.

Signalling basics for a telephone call

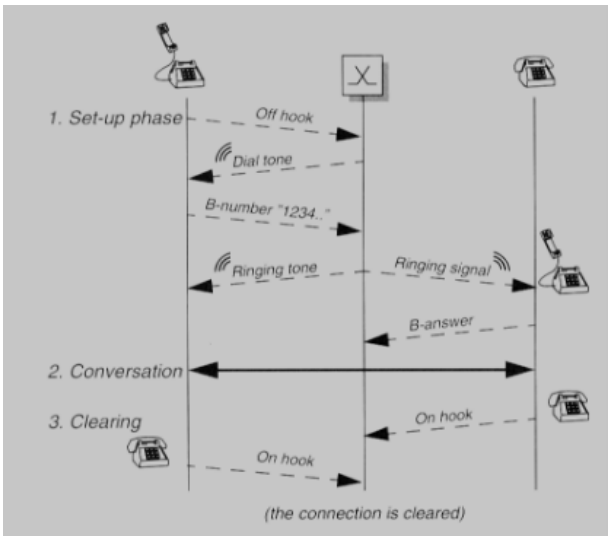


Figure 34: Signaling basic for a telephone call

Source of revenues
The revenues of the networks are the monthly fees and traffic based fees.

ADSL principles

- Asymmetric Digital Subscriber line
- A modem technology
- Convert existing twisted-pair telephone lines into access paths for multimedia and high speed data communication
- Can transmit to 30 Mbps downstream (VDSL 100 Mbps)
- Can transmit up to 20 Mbps upstream
- Transform the existing PSTN network to a powerful system capable of bringing multimedia, full motion video to the subscriber's home
- Max covered distance 3,6 km
- ITU_T G.992.x standards
- The frequency band separated in 3 parts: PSTN/ISDN, uplink, downlink
- Data links 4,3125 kHz channels
- Discrete Multi Tone (DMT) coding, 256 channels
- 1-5 channels for PSTN/ISDN
- 32 channels for uplink
- 218 channels for downlink

Network structures

There are 5 basic structures in PSTN/ISDN networks, and combining them we can make new (hybrid) structures as well. The basics are the followings:

1. Star topology
2. Multipolar topology
3. Meshed topology
4. Ring topology
5. Bus topology
6. Tree topology

There is no example for the tree topology in the picture above, but the MATAV network structure is a tree topology, see below.

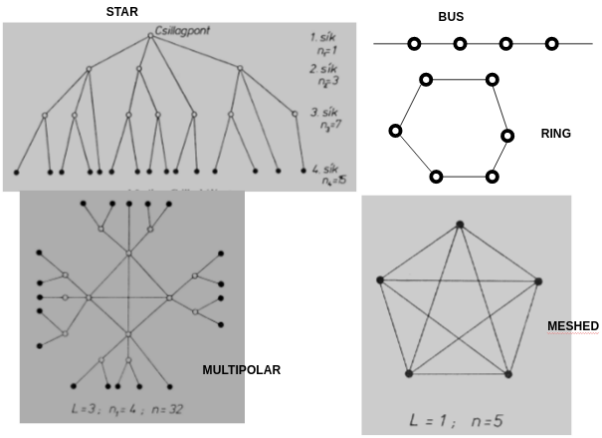


Figure 35: Topologies

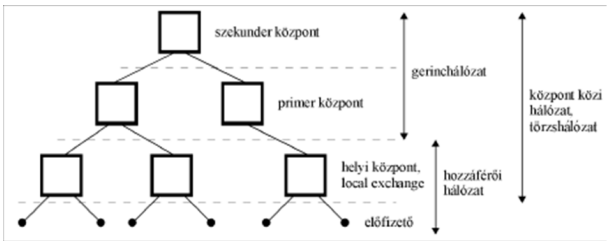


Figure 36: MATAV structure

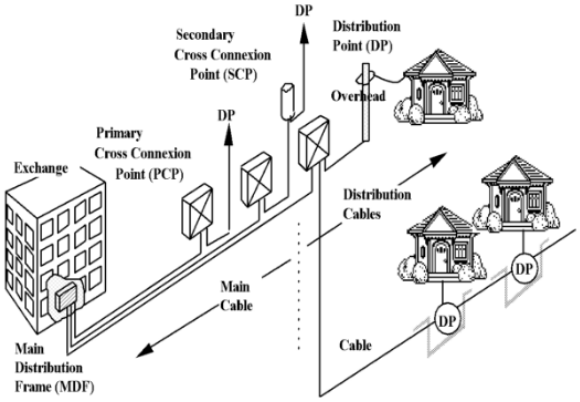


Figure 37: Access network implementation

Network implementation and development trends

There are several issues with networks:

1. Access issues: rural, city, downtown, areas, in-door cabling, underground copper cables, overhead cables, ADSL, Fibre to the Curb (FTC), distribution frames
2. Core issues: underground fibres, overhead fibres
3. positioning the switching and multiplexing nodes
4. traffic issue: dimensioning of switches and links

An example of the access network implementation:

The services might be pre-paid or post-paid. Nowadays the pre-paid networks are decreasing in number.

The cost of the networks are shown in the figure above:

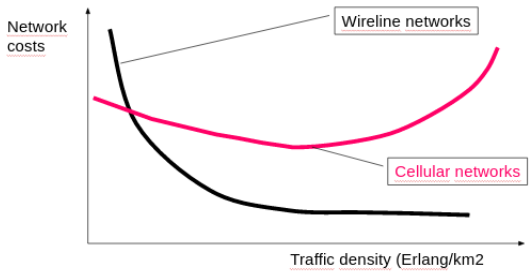


Figure 38: Cost of networks vs traffic density

3.10 10. topic

Description: CaTV, private (academic and university) networks

---

There are numerous private networks nowadays, such as

1. Closed User Group, Special Purpose network
2. Railway, transport, pipeline, fleet
3. Water management
4. Energy systems
5. Emergency services
6. Police networks
7. Military networks
8. Government networks
9. Company-wide networks (MOL, OTP)
10. Global Company Networks (Coca Cola)
11. Seat Reservation Networks (SITA)
12. Insurrance companies, Retail Chains (e.g. TESCO)

The common feature of private networks are the followings:

1. Internal numbering schemes, addressing system
2. Strictly regulated gateway function for interconnection to other (public) networks
3. The transmission part of networks might be leased line or own connection (radio)
4. The multiplexing, switching, management, authentication processes are private functions
5. Task oriented service quality parameters (reliability, usability, error rate, response time, redundancy, backup time ...)
6. Separated frequency management ("governmental" use)

Examples of private networks:

1. Hungarnet – for research and academic community in Hungary
2. Pázmány CU is one of the members
3. Governmental support (?)
4. Part of EU GEANT project
5. The transmission part is set of leased dark fibre connections
6. The switching and operation function in the hand of HUNGARNET (NIIF)

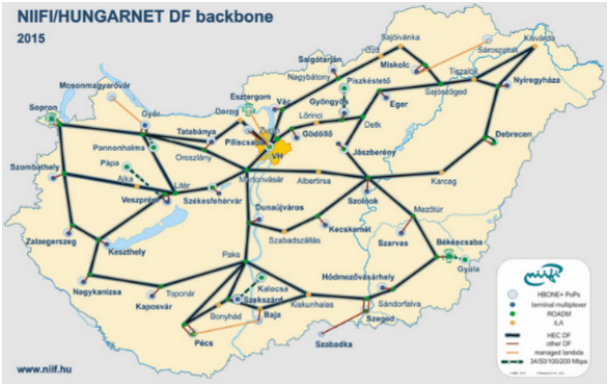


Figure 39: NIIF/HUNGARNET backbone

Here you can see the structure of the NIIF/HUNGARNET backbone:

HBONE is the backbone (computer) network of NIIF or Hungarian Academic Community. HBONE delivers services to Hungarian universities and colleges, primary and secondary schools, research and development institutions, libraries, public collections and also to several other non-profit public institutions. HBONE is a separate (telecommunication) network with a closed user-base. HBONE consists of the versatile core, and also the regional center (PoP) routers connected to core routers (directly or indirectly). In the PoPs NIIF operates versatile and robust communication equipments (DWDM, router, switch), servers and other devices required to serve other NIIF services in a climatised and UPS protected environment.

The topology of it is star, because in Hungary, everything is concentrated in Budapest.

The GEANTÍ network is one of the largest and most complex research and education networks in the world. Hence it needs to support a diverse range of users and services from standard IP transit services to ultra-high capacity data centre interconnects. GÉANT has built a dual layer network able to integrate these service demands across a single core structure. The two layers are: transmission and packet.

The transmission layer is built on the dark fibre core of the network, either on GÉANT points of presence (PoPs) or on leased wavelengths from commercial providers or national research and education networking (NREN) organisations. GÉANT uses industry-leading Infinera DTN-X equipment to light this fibre, a cutting-edge optical transmission equipment that integrates hundreds of optical components on a single chip to deliver 500G super-channels. GÉANT employs Infinera DTN-X boxes and optical amplifiers to drive the fibre backbone. The DTN-X also includes an integrated OTN switching layer with a GMPLS control plane which allows rapid provision of bandwidth and fast capacity restoration in the case of fibre cuts. The Infinera equipment is used to deliver GÉANT Lambda services and IP trunks between the GÉANT routers.

The packet layer is a converged layer that supports both Layer2 and Layer3 services. This means that we offer both Ethernet connections (GÉANT Plus) and IP services on this layer. GÉANT implements the packet layer with Juniper MX equipment. MX is a carrier grade router which supports labels switching (MPLS), router virtualisation and more. The GÉANT Plus service is delivered using MPLS technology in the MXs. IP services include: GÉANT IP, Layer 3 VPN, GÉANT World Service, peering service.

Basic issues in private network planning

1. Existing or new building
2. Single site or separated sites
3. Integrated or dedicated networks
4. Selecting of the transport technologies (optical, copper or radio)
5. Design of the network topology (star, meshed, ...)
6. Optimal placing and dimensioning of nodes

7. Duct system planning

Existing or new building

1. The lifetime of the building is minimum 100 years.
2. The lifetime of a network technology is about 10 years.
3. The capacity demand is permanently increasing.
4. The physical place of the duct system is defined by the building construction.
5. Critical places are: vertical ducts, backbone parts, distribution frames.
6. Further critical issues: powering, climatic system capacities, uninterrupted powering

Multiple site networks

1. Multi site systems need standardised interfaces (physical, protocol and signalling)
2. Interconnection links are usually leased lines. Managed leased links and spare capacities can provide the required reliability. Spared links might be switched connections.
3. Independent path or technology (radio or wired) can improve availability.

Integrated or dedicated networks

1. The terminals might be computers, TV-sets, mobile phones radio sets or universal devices like the smartphones.
2. Popular solution is a unified access like a structured network.
3. The structured network has vertical and horisontal links. The interconnection points are in distribution frames.
4. Radio based access fits well to the structured systems.

Technology selection

1. In one optical fibre in one wavelength window can be transported 10 Gbit/s
2. In a UTP cable up to 100m can be transported 10 Gbit/s
3. A WIFI access point can transport 300Mbit/s using 802.11.n standards and 6 Gbit/s 802.11.ac in the 5GHz band
4. Transport technology standards are in IEEE 802.3 series
5. 1 m UTP CAT6 takes about 0,5$, easy to install, one port takes about 5$
6. 1 m optical cable takes about 1$, installation require special tools and skills, one port takes about 200$.
7. UTP cables can be install easily in new ducts.
8. Optical cables can be installed (e. g. by compressed air into existing holes, ducts)
9. a 802.11.n WIFI access points takes about 40$
10. The prices are decreasing!

Topology selection

1. The basic form is the star topology in the horizontal part.
2. The physical place of star nodes is price sensitive.
3. The meshed horizontal is advised in the case of high reliability.
4. The multiple connection to outside (public) networks can improve the availability.

Optimal placing and dimensioning of nodes

1. A small (16-20 port switch) takes about 100$. – and this is equal to the price of 200m UTP cable!
2. A new node in a room is economical in the case of 20 terminals, if the nearest existing node distance is more then 10 m.
3. The usual port number of active devices are 5-8-16-20-40. Their price is decreasing.

Duct system planning

1. Careful laying and installation is required. The radius of turning is defined. The fixing of cables must be soft.
2. The ducts of structured cables must be well separated from the powering cabling. Careful grounding is required for safety reasons and to reduce interferences.
3. Spare capacity in the ducts must be minimum 5%!

CaTV

Cable television is a system of delivering television programming to paying subscribers via radio frequency (RF) signals transmitted through coaxial cables or, in the 2010s, light pulses through fiber-optic cables. This contrasts with broadcast television, in which the television signal is transmitted over the air by radio waves and received by a television antenna attached to the television. FM radio programming, high-speed Internet, telephone services, and similar non-television services may also be provided through these cables. Analog television was standard in the 20th century, but since the 2000s, cable systems have been upgraded to digital cable operation.

A "cable channel" (sometimes known as a "cable network") is a television network available via cable television. When available through satellite television, including direct broadcast satellite providers such as DirecTV, Dish Network and BSkyB, as well as via IPTV providers such as Verizon FIOS and AT&T U-verse is referred to as a "satellite channel". Alternative terms include "non-broadcast channel" or "programming service", the latter being mainly used in legal contexts. Examples of cable/satellite channels/cable networks available in many countries are HBO, MTV, Cartoon Network, E!, Eurosport and CNN International.

The abbreviation CaTV is often used for cable television. It originally stood for Community Access Television or Community Antenna Television, from cable television's origins in 1948. In areas where over-the-air TV reception was limited by distance from transmitters or mountainous terrain, large "community antennas" were constructed, and cable was run from them to individual homes. The origins of cable broadcasting for radio are even older as radio programming was distributed by cable in some European cities as far back as 1924.

Main characteristics of CaTV

- Traditional AM VSB TV sets
- Set top boxes for receiving DVB programs (including demodulator, MPEG decoder and some sort of descramblers)
- Internal frequency plan with 8 MHz raster (free assignment of programs to 8 MHz channels)

- Low split system: from 5 MHz up to 55 (50, 68) MHz for the uplink path, from 70 (87) MHz up to 630 MHz for the analogue downlink path and 630-862MHz for digital downlink path
- 8 TV and 8 radio channel in one 8 MHz channel (in the digital channels)
- The nominal impedance at all connection points of CATV system is 75 ohms

Topologies

- string
- tap-off
- star

3.11 11. topic

Description: Main functions and characteristics of terminals, interfaces, regulation of terminals

---

Terminals:

- Terminals are parts of the networks but individual elements
- No terminals = No electronic communications
- Terminals are commerced in normal shops and supermarkets and they are owned by users

Eg. Telephone:

- Basic technical functions and requirements
- Handset requirements
- Hands free terminal requirements
- Keyboard requirements
- Display requirements
- Intelligence in the terminal
- Special requirements of elderly or handicapped people

Main functions

- **B** (battery supply)
- **O** (overload protection)
- **R** (ringing)
- **S** (supervision, signalling)
- **C** (coding)
- **H** (hybrid, 2/4 wire transformation)
- **T** (testing)

There are some functional elements of telephone terminals, such as speech, dialer, ringing circuits, handset. Additionally there can be display, number memory, message recorder, B-number display, SMS receiver, etc.

Regulations

Teleservices from user point of view:

- Interactive services (telephone, videoconference, ...)
- Messaging services (voice mail, e-mail, ...)
- Retrieval services (account balance retrieval, time table, ...)
- Distribution services (cable TV, personalized news by fax, ...)

3.11.1 Historical stages of regulation, the reason of competition instead of monopoly in electronic communication

The brief history of the regulation of telecommuncation:

- Default (governmental) monopoly (established price, obligation to supply, terminals, uniform system – low number of regulation needed)
- Separation of the mailing services, telecommunication, broadcasting, official areas
- Private owners, concessive aggrement, but keeping the exclusiveness and the obligation to supply and develop
- Restricted cimpetition, helping the market's newcomers, liberalise the asset market with the obligation of the influential participants (many things to be regulated)
- Equal market conditions (few things to be regulated)

The scope, objective, and main principles of the electronic communications act is to establishment a reliable, transparent regulatory framework that facilitates the development of the electronic communications infrastructure, services and new technologies related to it, enhancing competition regardless of the technology applied.

The following licenses are necessary to provide electronic communication services, construct electronic communication facilities, distribute electronic communication equipment via numbering and addressing (identifiers):

- Identifiers may only be used subject to assignment licenses
- Except: Internet Protocol (IP) and electronic mail addresses as well as domain names

To promote competitions, there are some obligations in the electronic communication:

- Carrier selection
- Unbundling of local loop
- Number portability

The electronic communications service is a limited resource, and it is regulated in the electronic communcation act. This means a service normally provided for remuneration which consists wholly or mainly in the conveyance of signals on electronic communications networks, including telecommunications services and transmission services in networks used for broadcasting, but exclude services providing, or exercising editorial control over, content transmitted using electronic communications networks and services; it does not include information society services, as defined in Article 1 of Directive 98/34/EC, which do not consist wholly or mainly in the conveyance of signals on electronic communications networks;

- Area of policy: society of information
- Strategies (Europe2020), acting plans (Digital Agenda)
- Known issues:
  - Divided markets
  - Interoperability issues
  - Increasing number of cyber-crime, default distrust
  - Decreasing affinity to invest into the information technology
  - Unsatisfying quality of the research and innovation
  - Too many digital marksmen too few great specialists
  - Small window to solve problems like this in the EU

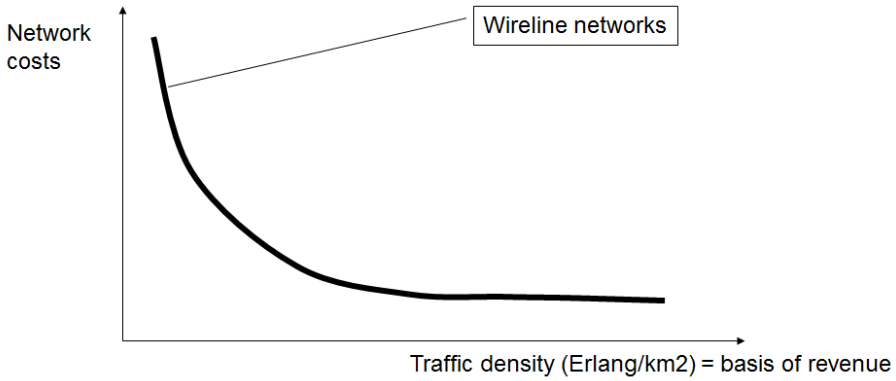


Figure 40: Market failure of electronic communication services = need for regulation

- The goal was to liberalise the electronic communcation sector and to raise the competition. The legal base of this was the accepted directive in 2002: Framework Access Autorisation Universal Service Privacy. This was included into the hungarian rules of law in 2003 with the C law.
- In 2009, the Committee reviewed the directives and created the BEREC (Body of European Regulators for Electronic Communications). The modificated directives was included into the hungarian rules of law in 2011 with the CVII-th rule.

3.12 12. topic

Description: Wireless LAN principles, IEEE802.11 standard

Wi-Fi or WiFi (Wireless Fidelity) is a technology that allows electronic devices to connect to a wireless LAN (WLAN), mainly using the 2.4 gigahertz (12 cm) UHF and 5 gigahertz (6 cm) SHF ISM radio bands. It follows the 802.11 standard.

Wi-Fi connections can be disrupted or the Internet speed lowered by having other devices in the same area. Many 2.4 GHz 802.11b and 802.11g access-points default to the same channel on initial startup, contributing to congestion on certain channels. Wi-Fi pollution, or an excessive number of access points in the area, especially on the neighboring channel, can prevent access and interfere with other devices' use of other access points, caused by overlapping channels in the 802.11g/b spectrum, as well as with decreased signal-to-noise ratio (SNR) between access points. This can become a problem in high-density areas, such as large apartment complexes or office buildings with many Wi-Fi access points. It is advised to only use channel 1-6-11.

Additionally, other devices use the 2.4 GHz band: microwave ovens, ISM band devices, security cameras, ZigBee devices, Bluetooth devices, video senders, cordless phones, baby monitors, and, in some countries, amateur radio, all of which can cause significant additional interference. It is also an issue when municipalities or other large entities (such as universities) seek to provide large area coverage.

In addition to running on different channels, multiple Wi-Fi networks can share channels.

A service set is the set of all the devices associated with a particular Wi-Fi network. The service set can be local, independent, extended or mesh.

Each service set has an associated identifier, the 32-byte Service Set Identifier (SSID), which identifies the particular network. The SSID is configured within the devices that are considered part of the network, and it is transmitted in the packets. Receivers ignore wireless packets from networks with a different SSID

RLAN (Radio LAN, also called WLAN)

- Radio version of LAN system
- Typical transmitting distance: 150 m or least

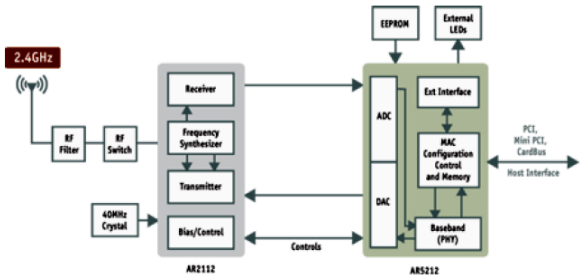


Figure 41:

The 802.11 Architecture

- User Stations (laptop PCs and PDAs)
- Access Points (APs)
- Backbone Network (Distribution System, DS)
- The User Stations competing for access over a shared medium is termed the Basic Service Set (BSS)
- Two or more of these BSSs are interconnected by a DS network
- The complete set of BSSs and the interconnecting network are termed an extended service set (ESS)

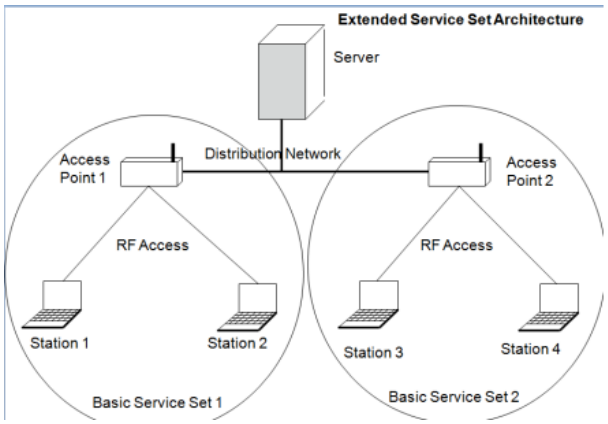


Figure 42:

Media Access Control (MAC)

- MAC is mandatory for all stations
- MAC is to assemble data into a frame including local address and error detection field
- MAC checks the frame address, perform error correction on the frame, disassemble the frame and passes it to the Logical Link Control
- The LLC identifies higher layer programs to handle the data and provides and interface to these higher-layer programs while perform flow and error control

3.13 13. topic

Description: IPTV, MPEG, TS, multimedia program distribution

IPTV

IPTV (Internet Protocol Television) is a system where a digital television service is delivered using Internet Protocol over a network infrastructure, which may include delivery by a broadband connection.

IPTV is typically supplied by a service provider using a closed network infrastructure. This closed network approach is in competition with the delivery of TV content over the public Internet, called Internet Television.

IPTV is defined as multimedia services such as television/video/audio/text/graphics/data delivered over IP-based networks managed to support the required level of QoS/QoE, security, interactivity and reliability.

MPEG-2 principles

- Intra-coding relies on two characteristics of typical images. First, not all spatial frequencies are simultaneously present, and second, the higher the spatial frequency, the lower the amplitude is likely to be. Intra-coding requires analysis of the spatial frequencies in an image.
- Inter-coding relies on finding similarities between successive pictures. The next picture can be created by sending only the picture differences. The shifting process is controlled by a pair of horizontal and vertical displacement values (collectively known as the motion vector) that is transmitted to the decoder. The motion vector transmission requires less data than sending the picture-difference data.

Structure

- Hierarchikus
- Szekvencia
- Képcsoport
- Kép
- Szelet
- Makroblokk
- Blokk

MPEG stream and transport stream

An elementary stream is an endless near real-time signal. Program streams have variable-length packets with headers. In a transport stream the PES packets are further subdivided into short fixed-size packets and multiple programs can be carried in the same stream.

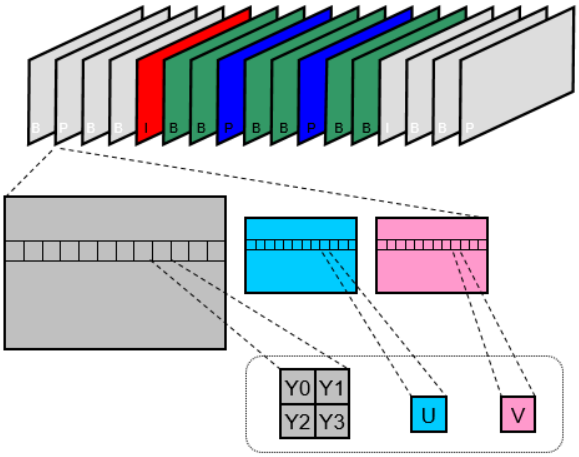


Figure 43: MPEG2 structure

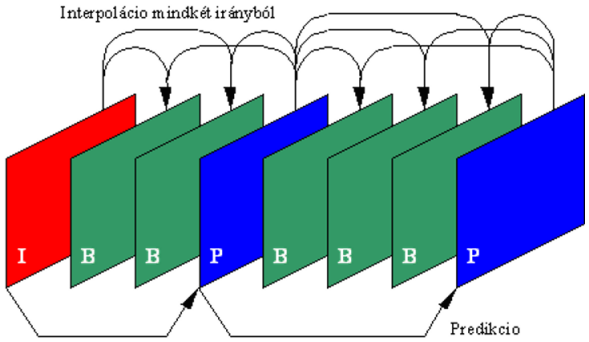


Figure 44: MPEG2 structure

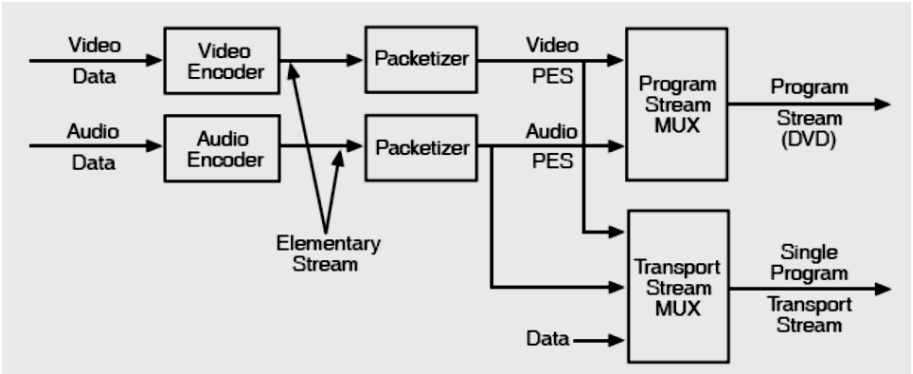


Figure 45: MPEG2 streams

3.14 14. topic

Description: VoIP, SÍP, ADSL

VoIP principles and versions

Voice over Internet Protocol (VoIP): voice traffic carried wholly or partly using IP over broadband networks competing with incumbent operators. VoIP is an acronym for Voice Over Internet Protocol, or in more common terms phone service over the Internet. If you have a reasonable quality Internet connection you can get phone service delivered through your Internet connection instead of from your local phone company. VoIP $\neq$ Skype!!

Key Issues:

- SIP - Session Initiation Protocol
- Voice CODEC
- Packet Loss Control

Versions:

- Cordless Hard Phones
- Dialup Hard Phones A dialup hard phone is a hard phone with a built-in modem instead of the Ethernet port
- WLAN or WiFi Phones A WLAN or WiFi phone is a hard phone with a built-in WiFi transceiver unit instead of an Ethernet port to connect to a WiFi base station and from there to a remote VoIP server
- Hard Phones (voice and video) Hard phones with video telephony support
- Soft Phones (voice only) A soft phone is an IP telephone in software. It can be installed on a personal computer and function as an IP phone. Soft phones require appropriate audio hardware to be present on the personal computer they run
- Soft Phones (voice and video)

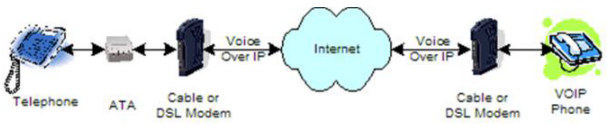


Figure 46:

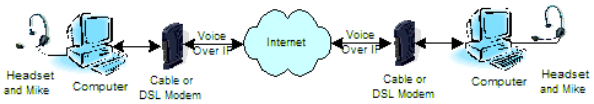


Figure 47:

VoIP codecs

Codecs are used to convert an analog voice signal to digitally encoded version. Codecs vary in the sound quality, the bandwidth required, the computational requirements, etc.

Each service, program, phone, gateway, etc typically supports several different codecs, and when talking to each other, negotiate which codec they will use.

GIPS, GSM, ITU,SILK...

- GIPS Family - 13.3 Kbps and up
- GSM - 13 Kbps (full rate), 20ms frame size
- iLBC - 15Kbps, 20ms frame size: 13.3 Kbps, 30ms frame size
- ITU G.711 - 64 Kbps, sample-based Also known as alaw/ulaw
- ITU G.722 - 48/56/64 Kbps ADPCM 7Khz audio bandwidth
- ITU G.722.1 - 24/32 Kbps 7Khz audio bandwidth (based on Polycom's SIREN codec)
- ITU G.722.1C - 32 Kbps, a Polycom extension, 14Khz audio bandwidth
- ITU G.722.2 - 6.6Kbps to 23.85Kbps. Also known as AMR-WB. CELP 7Khz audio bandwidth
- ITU G.723.1 - 5.3/6.3 Kbps, 30ms frame size
- ITU G.726 - 16/24/32/40 Kbps
- ITU G.728 - 16 Kbps
- ITU G.729 - 8 Kbps, 10ms frame size
- Speex - 2.15 to 44.2 Kbps
- LPC10 - 2.5 Kbps
- DoD CELP - 4.8 Kbps
- SILK

3.14.1 ADSL networks

ADSL principles

- Asymmetric Digital Subscriber line
- A modem technology
- Convert existing twisted-pair telephone lines into access paths for multimedia and high speed data communication
- Can transmit to 30 Mbps downstream (VDSL 100 Mbps)
- Can transmit up to 20 Mbps upstream
- Transform the existing PSTN network to a powerful system capable of bringing multimedia, full motion video to the subscriber's home

Technology

- No ultimate technology!
- Frequency division multiplexing, time division multiplexing, modulation, error control, flow control, scrambling, signal processing, adaptation, STM-ATM, trellis coding, in-service performance monitoring and surveillance, initialisation, handshaking, channel analysis, are all mixed in ADSL

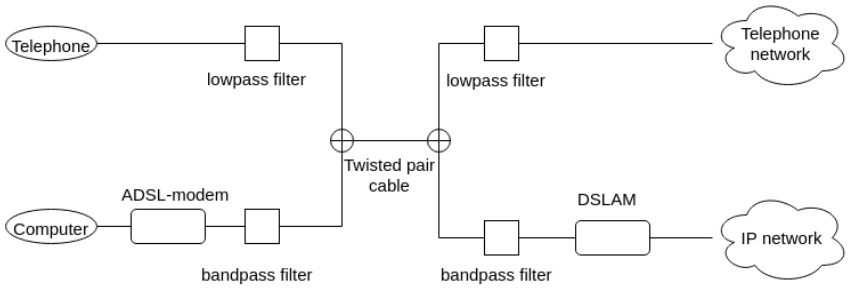


Figure 48: ADSL system components

- More room for further development...

VDSL Very-high-bit-rate digital subscriber line (VDSL or VHDSL) is a digital subscriber line (DSL) technology providing data transmission faster than asymmetric digital subscriber line (ADSL) over a single flat untwisted or twisted pair of copper wires (up to 52 Mbit/s downstream and 16 Mbit/s upstream), and on coaxial cable (up to 85 Mbit/s down- and upstream) using the frequency band from 25 kHz to 12 MHz. These rates mean that VDSL is capable of supporting applications such as high-definition television, as well as telephone services (voice over IP) and general Internet access, over a single connection. VDSL is deployed over existing wiring used for analog telephone service and lower-speed DSL connections. This standard was approved by ITU in November 2001.