

# Waveform Coding Techniques

Document ID: 8123

---

## **Introduction**

### **Prerequisites**

- Requirements

- Components Used

- Conventions

### **Pulse Code Modulation**

- Filtering

- Sampling

- Digitize Voice

- Quantization and Coding

- Companding

- A-law and  $\mu$ -law Companding

### **Differential Pulse Code Modulation**

### **Adaptive DPCM**

- Specific 32 KB/s Steps

### **Related Information**

---

## **Introduction**

Although humans are well equipped for analog communications, analog transmission is not particularly efficient. When analog signals become weak because of transmission loss, it is hard to separate the complex analog structure from the structure of random transmission noise. If you amplify analog signals, it also amplifies noise, and eventually analog connections become too noisy to use. Digital signals, having only "one-bit" and "zero-bit" states, are more easily separated from noise. They can be amplified without corruption. Digital coding is more immune to noise corruption on long-distance connections. Also, the world's communication systems have converted to a digital transmission format called pulse code modulation (PCM). PCM is a type of coding that is called "waveform" coding because it creates a coded form of the original voice waveform. This document describes at a high level the conversion process of analog voice signals to digital signals.

## **Prerequisites**

### **Requirements**

There are no specific requirements for this document.

### **Components Used**

This document is not restricted to specific software and hardware versions.

### **Conventions**

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

# Pulse Code Modulation

PCM is a waveform coding method defined in the ITU–T G.711 specification.

## Filtering

The first step to convert the signal from analog to digital is to filter out the higher frequency component of the signal. This makes things easier downstream to convert this signal. Most of the energy of spoken language is somewhere between 200 or 300 hertz and about 2700 or 2800 hertz. Roughly 3000–hertz bandwidth for standard speech and standard voice communication is established. Therefore, they do not have to have precise filters (it is very expensive). A bandwidth of 4000 hertz is made from an equipment point of view. This band-limiting filter is used to prevent aliasing (antialiasing). This happens when the input analog voice signal is undersampled, defined by the Nyquist criterion as  $F_s < 2(BW)$ . The sampling frequency is less than the highest frequency of the input analog signal. This creates an overlap between the frequency spectrum of the samples and the input analog signal. The low-pass output filter, used to reconstruct the original input signal, is not smart enough to detect this overlap. Therefore, it creates a new signal that does not originate from the source. This creation of a false signal when sampling is called aliasing.

## Sampling

The second step to convert an analog voice signal to a digital voice signal is to sample the Filtered input signal at a constant sampling frequency. It is accomplished by using a process called pulse amplitude modulation (PAM). This step uses the original analog signal to modulate the amplitude of a pulse train that has a constant amplitude and frequency. (See Figure 2.)

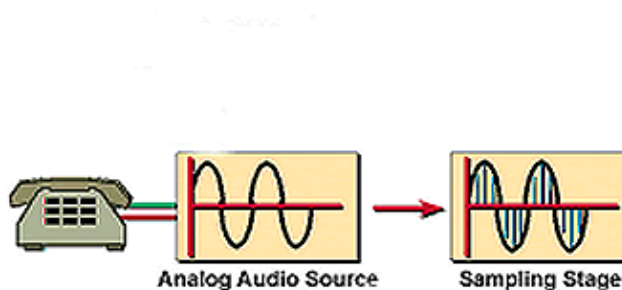
The pulse train moves at a constant frequency, called the sampling frequency. The analog voice signal can be sampled at a million times per second or at two to three times per second. How is the sampling frequency determined? A scientist by the name of Harry Nyquist discovered that the original analog signal can be reconstructed if enough samples are taken. He determined that if the sampling frequency is at least twice the highest frequency of the original input analog voice signal, this signal can be reconstructed by a low-pass filter at the destination. The Nyquist criterion is stated like this:

$$F_s > 2(BW)$$

$F_s$  = Sampling frequency

$BW$  = Bandwidth of original analog voice signal

**Figure 1: Analog Sampling**

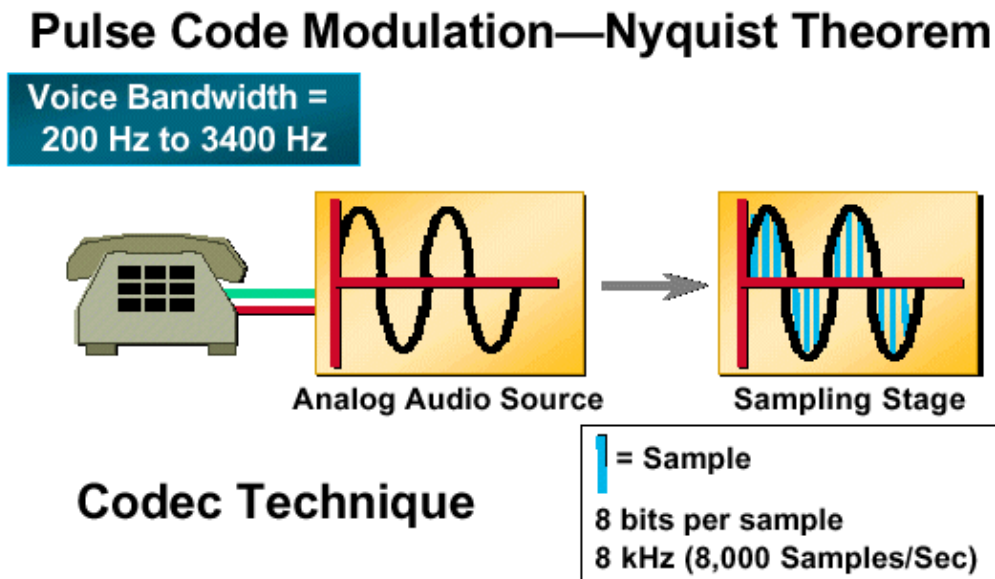


## Digitize Voice

After you filter and sample (using PAM) an input analog voice signal, the next step is to digitize these samples in preparation for transmission over a Telephony network. The process of digitizing analog voice signals is called PCM. The only difference between PAM and PCM is that PCM takes the process one step further. PCM decodes each analog sample using binary code words. PCM has an analog-to-digital converter on the source side and a digital-to-analog converter on the destination side. PCM uses a technique called quantization to encode these samples.

## Quantization and Coding

Figure 2: Pulse Code Modulation – Nyquist Theorem



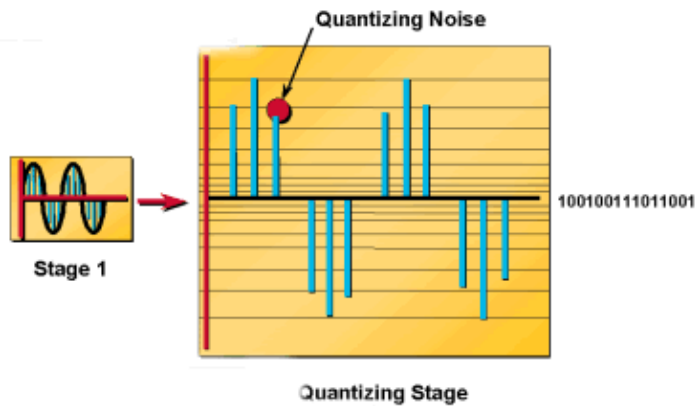
Quantization is the process of converting each analog sample value into a discrete value that can be assigned a unique digital code word.

As the input signal samples enter the quantization phase, they are assigned to a quantization interval. All quantization intervals are equally spaced (uniform quantization) throughout the dynamic range of the input analog signal. Each quantization interval is assigned a discrete value in the form of a binary code word. The standard word size used is eight bits. If an input analog signal is sampled 8000 times per second and each sample is given a code word that is eight bits long, then the maximum transmission bit rate for Telephony systems using PCM is 64,000 bits per second. Figure 2 illustrates how bit rate is derived for a PCM system.

Each input sample is assigned a quantization interval that is closest to its amplitude height. If an input sample is not assigned a quantization interval that matches its actual height, then an error is introduced into the PCM process. This error is called quantization noise. Quantization noise is equivalent to the random noise that impacts the signal-to-noise ratio (SNR) of a voice signal. SNR is a measure of signal strength relative to background noise. The ratio is usually measured in decibels (dB). If the incoming signal strength in microvolts is  $V_s$ , and the noise level, also in microvolts, is  $V_n$ , then the signal-to-noise ratio,  $S/N$ , in decibels is given by the formula  $S/N = 20 \log_{10}(V_s/V_n)$ . SNR is measured in decibels (dB). The higher the SNR, the better the voice quality. Quantization noise reduces the SNR of a signal. Therefore, an increase in quantization noise degrades the quality of a voice signal. Figure 3 shows how quantization noise is generated. For coding purpose, an  $N$  bit word yields  $2^N$  quantization labels.

Figure 3: Analog to Digital Conversion

## Pulse Code Modulation— Analog to Digital Conversion



One way to reduce quantization noise is to increase the amount of quantization intervals. The difference between the input signal amplitude height and the quantization interval decreases as the quantization intervals are increased (increases in the intervals decrease the quantization noise). However, the amount of code words also need to be increased in proportion to the increase in quantization intervals. This process introduces additional problems that deal with the capacity of a PCM system to handle more code words.

SNR (including quantization noise) is the single most important factor that affects voice quality in uniform quantization. Uniform quantization uses equal quantization levels throughout the entire dynamic range of an input analog signal. Therefore, low signals have a small SNR (low-signal-level voice quality) and high signals have a large SNR (high-signal-level voice quality). Since most voice signals generated are of the low kind, having better voice quality at higher signal levels is a very inefficient way of digitizing voice signals. To improve voice quality at lower signal levels, uniform quantization (uniform PCM) is replaced by a nonuniform quantization process called companding.

### Companding

Companding refers to the process of first compressing an analog signal at the source, and then expanding this signal back to its original size when it reaches its destination. The term companding is created by combining the two terms, compressing and expanding, into one word. At the time of the companding process, input analog signal samples are compressed into logarithmic segments. Each segment is then quantized and coded using uniform quantization. The compression process is logarithmic. The compression increases as the sample signals increase. In other words, the larger sample signals are compressed more than the smaller sample signals. This causes the quantization noise to increase as the sample signal increases. A logarithmic increase in quantization noise throughout the dynamic range of an input sample signal keeps the SNR constant throughout this dynamic range. The ITU-T standards for companding are called A-law and u-law.

### A-law and u-law Companding

A-law and u-law are audio compression schemes (codecs) defined by Consultative Committee for International Telephony And Telegraphy (CCITT) G.711 which compress 16-bit linear PCM data down to eight bits of logarithmic data.

#### A-law Compressor

Limiting the linear sample values to twelve magnitude bits, the A-law compression is defined by this equation, where  $A$  is the compression parameter ( $A=87.7$  in Europe), and  $x$  is the normalized integer to be

compressed.

$$F(x) = \begin{cases} \frac{A * |x|}{1 + \ln(A)} & 0 \leq |x| < \frac{1}{A} \\ \frac{\text{sgn}(x) * (1 + \ln(A|x|))}{1 + \ln(A)} & \frac{1}{A} \leq |x| \leq 1 \end{cases}$$

### u-law Compander

Limiting the linear sample values to thirteen magnitude bits, the u-law (u-law and Mu-law are used interchangeably in this document) compression is defined by this equation, where m is the compression parameter (m=255 in the U.S. and Japan) and x is the normalized integer to be compressed.

$$F(x) = \frac{\text{sgn}(x) * \ln(1 + \mu|x|)}{\ln(1 + \mu)} \quad 0 \leq |x| \leq 1$$

A-law standard is primarily used by Europe and the rest of the world. u-law is used by North America and Japan.

### Similarities Between A-law and u-law

- Both are linear approximations of logarithmic input/output relationship.
- Both are implemented using eight-bit code words (256 levels, one for each quantization interval). Eight-bit code words allow for a bit rate of 64 kilobits per second (kbps). This is calculated by multiplying the sampling rate (twice the input frequency) by the size of the code word (2 x 4 kHz x 8 bits = 64 kbps).
- Both break a dynamic range into a total of 16 segments:
  - ◆ Eight positive and eight negative segments.
  - ◆ Each segment is twice the length of the preceding one.
  - ◆ Uniform quantization is used within each segment.
- Both use a similar approach to coding the eight-bit word:
  - ◆ First (MSB) identifies polarity.
  - ◆ Bits two, three, and four identify segment.
  - ◆ Final four bits quantize the segment are the lower signal levels than A-law.

### Differences Between A-law and u-law

- Different linear approximations lead to different lengths and slopes.
- The numerical assignment of the bit positions in the eight-bit code word to segments and the quantization levels within segments are different.
- A-law provides a greater dynamic range than u-law.
- u-law provides better signal/distortion performance for low level signals than A-law.
- A-law requires 13-bits for a uniform PCM equivalent. u-law requires 14-bits for a uniform PCM equivalent.
- An international connection needs to use A-law, u to A conversion is the responsibility of the u-law country.

# Differential Pulse Code Modulation

At the time of the PCM process, the differences between input sample signals are minimal. Differential PCM (DPCM) is designed to calculate this difference and then transmit this small difference signal instead of the entire input sample signal. Since the difference between input samples is less than an entire input sample, the number of bits required for transmission is reduced. This allows for a reduction in the throughput required to transmit voice signals. Using DPCM can reduce the bit rate of voice transmission down to 48 kbps.

How does DPCM calculate the difference between the current sample signal and a previous sample? The first part of DPCM works exactly like PCM (that is why it is called differential PCM). The input signal is sampled at a constant sampling frequency (twice the input frequency). Then these samples are modulated using the PAM process. At this point, the DPCM process takes over. The sampled input signal is stored in what is called a predictor. The predictor takes the stored sample signal and sends it through a differentiator. The differentiator compares the previous sample signal with the current sample signal and sends this difference to the quantizing and coding phase of PCM (this phase can be uniform quantizing or companding with A-law or  $\mu$ -law). After quantizing and coding, the difference signal is transmitted to its final destination. At the receiving end of the network, everything is reversed. First the difference signal is dequantized. Then this difference signal is added to a sample signal stored in a predictor and sent to a low-pass filter that reconstructs the original input signal.

DPCM is a good way to reduce the bit rate for voice transmission. However, it causes some other problems that deal with voice quality. DPCM quantizes and encodes the difference between a previous sample input signal and a current sample input signal. DPCM quantizes the difference signal using uniform quantization. Uniform quantization generates an SNR that is small for small input sample signals and large for large input sample signals. Therefore, the voice quality is better at higher signals. This scenario is very inefficient, since most of the signals generated by the human voice are small. Voice quality needs to focus on small signals. To solve this problem, adaptive DPCM is developed.

## Adaptive DPCM

Adaptive DPCM (ADPCM) is a waveform coding method defined in the ITU-T G.726 specification.

ADPCM adapts the quantization levels of the difference signal that generated at the time of the DPCM process. How does ADPCM adapt these quantization levels? If the difference signal is low, ADPCM increases the size of the quantization levels. If the difference signal is high, ADPCM decreases the size of the quantization levels. So, ADPCM adapts the quantization level to the size of the input difference signal. This generates an SNR that is uniform throughout the dynamic range of the difference signal. Using ADPCM reduces the bit rate of voice transmission down to 32 kbps, half the bit rate of A-law or  $\mu$ -law PCM. ADPCM produces "toll quality" voice just like A-law or  $\mu$ -law PCM. Coder must have feedback loop, using encoder output bits to recalibrate the quantizer.

## Specific 32 KB/s Steps

Applicable as ITU Standards G.726.

- Turn A-law or  $\mu$ -law PCM samples into a linear PCM sample.
  - Calculate the predicted value of the next sample.
  - Measure the difference between actual sample and predicted value.
  - Code difference as four bits, send those bits.
  - Feed back four bits to predictor.
  - Feed back four bits to quantizer.
-

## Related Information

- **Voice Technology Support**
  - **Voice and Unified Communications Product Support**
  - **Recommended Reading: Troubleshooting Cisco IP Telephony**
  - **Technical Support – Cisco Systems**
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Feb 02, 2006

Document ID: 8123

---