

# **56Kbps Data Transmission across the PSTN**

**Rockwell Semiconductor Systems  
Network Access Division  
October 14, 1998**

**P. Michael Henderson**  
(949) 221-4104  
(949) 221-7314 (fax)  
[michael.henderson@rss.rockwell.com](mailto:michael.henderson@rss.rockwell.com)

On September 10, 1996, Rockwell Semiconductor Systems surprised the communications industry with its announcement of a revolutionary new dial up modem technology for communicating across the Public Switched Telephone Network (PSTN) at rates up to 56 Kbps.

Prior to this announcement, the modem communications industry had convinced itself that communications across the PSTN were limited by Shannon's Theorem to rates below 35Kbps. Although Rockwell published a white paper describing its 56Kbps technology shortly after the public announcement, and submitted its technology for standardization to both the ITU and the ANSI TR30 committees, there have been continual requests for an in-depth discussion of how the technology works, with particular emphasis on how this technology gets around the so-called Shannon Limit. This paper attempts to address these issues.

## Modem Fundamentals

Before describing how Rockwell's 56Kbps technology works, let me first discuss how a traditional analog voice band modem works.

The voice band telephone channel is a bandpass channel, traditionally thought of as operating from about 300 Hz to 3,000 Hz. Modem modulations, therefore, had to operate within this band. Early modems used tones (e.g., FSK) which fell within this frequency band for communicating data but the information density was not very high (the number of bits per hertz was significantly less than one). Quadrature amplitude modulation (QAM) was a significant improvement, offering information densities of multiple bits per hertz.

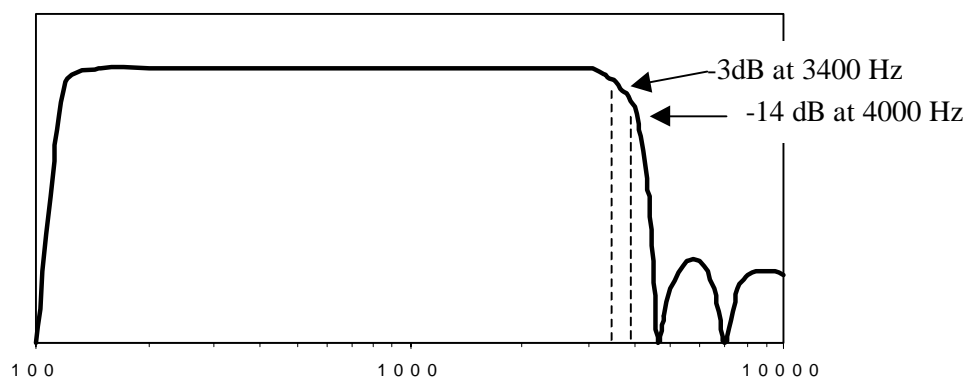


Figure 1: Approximate frequency response of the filters associated with the codec in the line card. An ordinary telephone call will go through at least two of these filters, causing significant high frequency rolloff, usually 28 dB or more at 4 KHz.

QAM operates by modulating a carrier sine wave signal in both amplitude and phase. Each unique combination of amplitude and phase is known as a “symbol”. In the general case, a symbol is defined as an information carrying token which is sent from the transmitter to the receiver.

In the early days of modems, these tokens were called “baud” in honor of the French inventor Emile Baudot who, in 1875, invented a 5 bit code for representing the alphabet. Each 5 bits were a token communicating a letter of the alphabet or a control code. Recently, however, the term baud has become corrupted in common usage, with people using baud to mean bits per second. This is a throwback to the early days of modems when a baud carried only one bit (so the baud rate and the bit rate were the same). Thus, 300 bit per second modems became 300 baud modems in techie-talk. The problem is that after a baud started carrying multiple bits, people were still using baud to mean bits per second. For modem designers, hearing someone describe a modem as a “9600 baud” or a “28.8 baud” modem is like fingernails on a blackboard. To avoid this, the modem cognoscenti began using the term “symbol”.

Now, when you modulate a sine wave, the resulting signal is no longer a single frequency sine wave. The resulting signal will be a range of frequencies, related to the signal which is modulated onto the carrier. Assuming random bits, the bandwidth of a modulated QAM signal is equal to the symbol rate. That is, if you send 2400 symbols per second, the bandwidth of the modem signal will be 2400 Hz. The V.32 modulation, for example, uses a carrier of 1800 Hz and a symbol rate of 2400 symbols per second. The bandwidth of the signal, therefore, is from 600 Hz to 3000 Hz.

Over time, modem designers began to realize that the telephone network was getting better and that more bandwidth was available. Newer modem modulations began to take advantage of these higher bandwidth channels. The highest V.34 rate, for example, uses a carrier of 1959 Hz, and a symbol rate of 3429 symbols per second, giving a bandwidth from about 244 Hz to 3674 Hz.

QAM, you’ll recall, is a modulation of a signal in both amplitude and phase. And when you talk of amplitude and phase, you immediately think of vectors. It turns out that modem designers use the concept of vectors to visualize the symbols being transmitted. For example, the figure below shows one representative symbol, a combination of amplitude and phase.

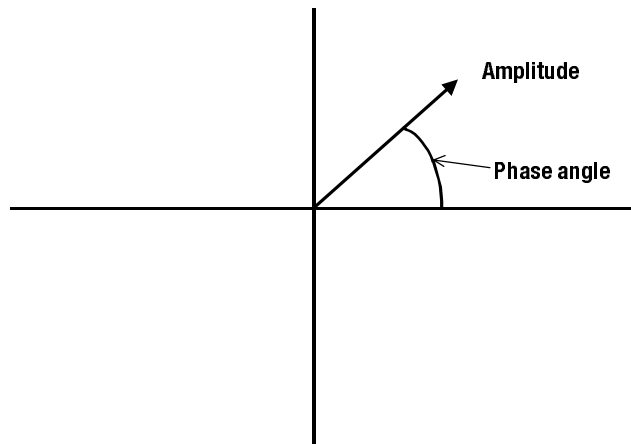


Figure 2: Graph of a single symbol, showing the amplitude and phase

As you try to represent a large number of symbols, however, the graph can get fairly busy. Additionally, some symbols will have the same phase angle but different values of amplitude. These two vectors will lay atop one another and it will be hard to see the one with the lesser amplitude. Because of this, modem designers graph only the end points of the vectors, using a dot to represent the end point location of the vector. When a large number of symbols are drawn on a graph, the resulting figure is known as a “constellation” because it begins to look like a star map. A simple constellation of only four symbols is shown in figure 3 below.

Figure 3 also illustrates the decision region, another aspect of the transmission of a symbol from the transmitter to the receiver. Although the transmitter sends the symbol accurately, the transmission channel can modify the symbol in ways that cause the symbol received by the receiver to be different than the one transmitted. For example, suppose a noise transient occurs during the transmission of the symbol. This may cause the amplitude to be larger or smaller than what was transmitted, which makes the constellation point move compared to what was transmitted. This difference is called the “error vector”. However, as long as the received constellation point is within the decision region, it will be interpreted as the correct symbol and the proper bits will be communicated.

As the modem uses more and more symbol points, the decision region shrinks, leading to a higher error rate in the presence of noise. Thus, a modem designer can’t just keep increasing the number of symbols to gain a higher data rate – at some point you can no longer communicate data at an acceptable error rate.

To a large degree, this is why your V.34 modem doesn’t connect at 33.6 Kbps very often. To communicate at 33.6 Kbps with V.34, the modem has to use a very large number of symbol points (1,664 symbols, 10.7 bits per symbol). Since the transmit power is limited, the overall size of the constellation can only be so large. This means that the symbol points must be very close together, making it *very* difficult for the receiver to accurately distinguish which symbol was sent. To operate, therefore, the modem must “downrate” to a constellation which uses a smaller number of symbol points. In reality, there are other types of channel impairments, such as available

bandwidth, which also limit operation at these higher rates, but noise, both Gaussian and quantization, plays a large part.

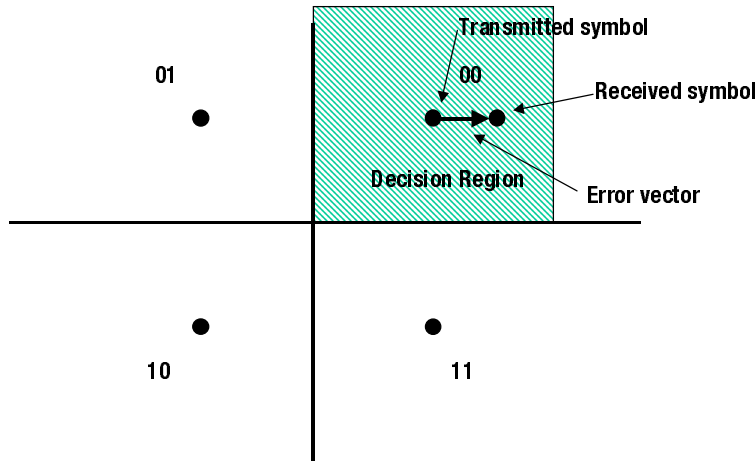


Figure 3: A four point constellation showing the values assigned to each symbol and the receiver decision region.

The error vector has been used in inventive ways by modem designers over the years. One of the earliest “tricks” was to use the error vector to transmit a low bit rate control channel. For example, suppose you have a channel with acceptable noise margin and want to create a 100 bit per second control channel to send diagnostic information between the two modems. You can do this by using the error vector as follows. First, pick a symbol in one of the most robust parts of the constellation so that there is the minimum chance of having that symbol perturbed by noise. Next, define perturbations of this symbol to represent bits in the control channel. For example, if the symbol is perturbed to the left, let that represent a “1” in the control channel. If perturbed to the right, let that represent a “0” in the control channel. Since the symbol can be perturbed by noise, send the same bit multiple symbol times.

Suppose you choose three times as the number of times to send a control channel bit and the symbol was perturbed by noise to the opposite side once during those three transmissions. A simple voting technique would allow you to recover the correct bit. A better technique would be to use a Viterbi decoder which saves the actual error vectors and sums them over the three transmissions. It can be shown mathematically that this technique, where the actual analog values of the error vectors are used, gives superior performance over a simple voting scheme where a decision is made on each error vector, after which voting is applied. The actual bit rate of the control channel depends upon how often you transmit that specific symbol which in turn depends upon the incoming bit pattern. It’s actually fairly easy to maintain an average bit rate in the control channel because a scrambler is applied to the incoming bits, producing a pseudo random transmitted bit pattern.

An engineer at AT&T Paradyne, Gordon Bremer, took this technique about as far as it could be taken when he had the insight that voice samples could be represented by this error vector,

allowing simultaneous voice and data transmission in the analog domain. His technique has been standardized by the ITU as V.61.

But back to modem fundamentals. How many symbols do you need in order to achieve a certain bit rate? In the figure shown above, with four symbols, each symbol represents two bits. If the symbol rate is 2400 symbols per second, the bit rate will be 4800 bps. In the general case, ignoring trellis coding, the number of symbols required to transmit a certain bit rate is given by the following equation.

$$\text{bps} = R_s \log_2 N_s \quad \text{Equation 1}$$

Where:

bps = bits per second

$R_s$  = symbol rate (also bandwidth for QAM)

$N_s$  = number of symbols in the constellation

Solving for  $N_s$ , gives the following equation:

$$\text{bps} = R_s \log_2 N_s$$

$$\text{bps}/R_s = \log_2 N_s$$

$$2^{\text{bps}/R_s} = N_s \quad \text{Equation 2}$$

Using this equation, how many constellation points are required for operation at 9600 bps, if a 2400 symbol rate is used? Substituting into the above equation gives the following.

$$N_s = 2^{\text{bps}/R_s}$$

$$N_s = 2^{9600/2400}$$

$$N_s = 2^4$$

$$N_s = 16$$

Only a 16 point constellation is required for operation at 9600 bps! By today's standards this seems fairly trivial but back in the days when we were first attempting to achieve 9600 bps operation, it was quite a challenge because V.32 also introduced echo cancellation, a DSP intensive technique which I won't describe here.

Another coding technique developed about the time of V.32 was trellis codes, generally credited to Gottfried Ungerboeck at IBM's Zurich Research Laboratory. Trellis codes are especially difficult to explain intuitively, but I'll try. If you look back at figure 2, you'll see how the decision region surrounds the transmitted symbol point. As discussed earlier, as more and more points are

defined, the decision region shrinks until a small amount of noise can perturb the symbol point into another symbol's decision region, causing an incorrect decode by the receiver.

Suppose, however, that you had a way to subset the constellation points, removing half or more of the points to maximize the decision region around each remaining point. This would allow the receiver to more accurately decode which symbol point was sent, in the presence of noise.

This is essentially what's done with trellis codes. A number of constellations are created, each with the maximum sized decision region around each symbol. Since each symbol point represents a unique string of bits, and if the receiver knew what constellation was being used, it could do a better job of decoding what symbol had actually been sent. The problem, of course, is to tell the receiver what subset constellation to use.

This is accomplished in an "after the fact" manner in the receiver by allowing only certain transitions from one state to another. Suppose that we create four different subset constellations. For each starting state, we then establish valid transitions to only two other states and use an extra bit per symbol to force legal transitions, only, through a convolutional encoder. The receiver decodes the symbols as it normally would, assuming only a single constellation. The receiver then does a "traceback" and examines the sequence of state transitions. If an invalid state transition occurred, the receiver then goes back and examines the error vectors for the symbols, computing the distance from the actually received symbol point to valid symbol points (Viterbi decoding). The closest valid symbol point, which also creates a legal sequence of state transitions, is selected.

Nothing is free, however. To accomplish this bit of magic, an extra bit per symbol (for two dimensional codes) or every two symbols (for four dimensional codes) is required. Sending an extra bit per symbol requires that we double the number of symbols and we lose 3 dB of signal to noise ratio (SNR) performance by doing this. We also add delay in the transmission. Since we can't be sure that the symbols were decoded accurately until we receive subsequent symbols and perform a traceback, we have to hold the symbol data in a buffer, adding delay. The trellis code, however, provides about 6 dB of coding gain<sup>1</sup> so we achieve about 3 dB better SNR performance.

## Shannon's Theorem

In the modern digital telephone network, an analog telephone call is carried over the digital portion of the network using 64 Kbps. From the work done with digital subscriber line (DSL) technology, we know that the subscriber's copper wire can carry 1.5 Mbps or more. Why, therefore, can we only achieve 33.6 Kbps (V.34)?<sup>2</sup>

---

<sup>1</sup> Coding gain results from the use of a mathematical technique which allows the modem to operate, with equal error rates, in the presence of greater noise than it could without the mathematical technique.

<sup>2</sup> This question is actually the genesis of 56Kbps modem technology and was attributed in a 1993 IEEE paper to R.W. Lucky of Bellcore (in a slightly different variation).

The answer to this has to do with quantization “noise” and a theorem published by Claude Shannon in 1948.

Quantization noise, or more properly quantization error, has to do with the fact that the analog signal created by the modem has to be converted to digital at the edge of the network. When a signal is converted from analog to digital, the analog signal is sampled at regular time intervals. The problem is that the analog signal can take any value while the digital value assigned to the signal at that sample time can only take certain discrete values. The difference between what the real analog signal level is and the digital value which is assigned to it is known as the quantization error. It is called quantization noise because this difference between actual and digital value is exactly the same as would occur if noise caused the signal to jump to the quantization value.

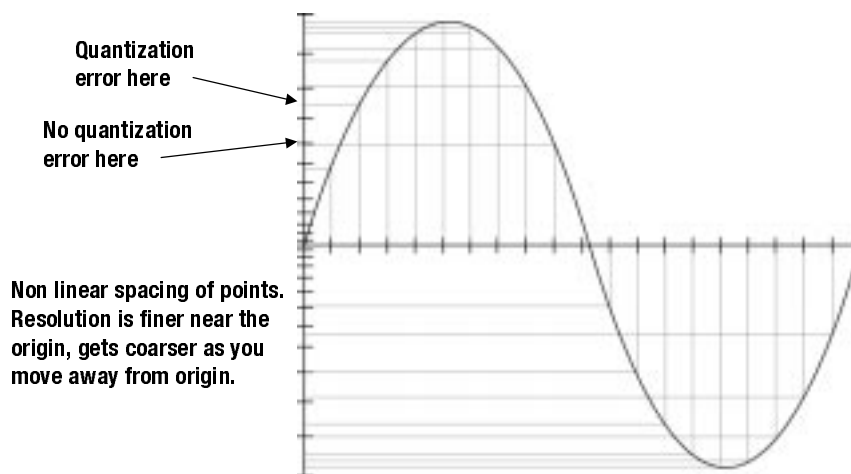


Figure 4: An example of quantizing an analog signal, showing quantization error.

Shannon showed in his ground breaking paper that there was a limit to the amount of information that could be communicated over a channel in the presence of noise. The equation he developed to express this is quite simple but has held up to examination over the years. The equation is:

$$\text{bps} = \text{BW} \log_2 (1 + \text{P/N}) \quad \text{Equation 3}$$

Where:

bps = bits per second

BW = channel bandwidth

P/N= signal to noise power ratio. This is a real power ratio and not a dB ratio.

The codec used in the telephone network in the United States has a theoretical noise floor of 39.5 dB but most real network codecs can only achieve a noise floor of 35 to 36 dB. Since the noise floor is expressed in dB but Shannon’s theorem requires the real signal to noise power ratio, let’s



see what we have to do to Shannon's equation to use dB. Many engineers remember that the real unit of logarithmic power ratio is the Bell, named for Alexander Graham Bell, and is defined as:

$$\text{Bell} = \log_{10} P/N$$

The decibel, or one tenth of a Bell, is defined as:

$$\text{dB} = 10 \log_{10} P/N \quad \text{Equation 4}$$

This is, of course, the reason why the decibel is abbreviated with a lower case "d" and an upper case "B" – the Bell portion of the dB represents Bell's name. Anyway, if we want to substitute into Shannon's theorem, we need to solve the dB equation for P/N. This can be done as follows.

$$\begin{aligned} \text{dB} &= 10 \log_{10} P/N \\ \text{dB}/10 &= \log_{10} P/N \\ 10^{\text{dB}/10} &= P/N \end{aligned} \quad \text{Equation 5}$$

Substituting into Shannon's theorem gives the following:

$$\begin{aligned} \text{bps} &= \text{BW} \log_2 (1 + P/N) \\ \text{bps} &= \text{BW} \log_2 (1 + 10^{\text{dB}/10}) \end{aligned} \quad \text{Equation 6}$$

If we assume a 3000 Hz channel bandwidth, a real world quantization noise floor of 35 dB, and substitute into the equation above, we get:

$$\begin{aligned} \text{bps} &= 3000 \log_2 (1 + 10^{35/10}) \\ \text{bps} &= 34,822 \end{aligned}$$

Thus, because of the quantization noise floor of the codec used in the network, it is not possible to achieve rates above 35 Kbps using ordinary modem techniques. However, if the quantization noise floor of the codec can be eliminated, significantly higher data rates can be achieved.

## **56Kbps Operation**

Now, all that stuff I told you about QAM modulation and amplitude and phase – forget it. 56Kbps modems operate using pulse amplitude modulation (PAM) and not QAM. However, many of the concepts described in the preceding sections still apply.

The network diagram required for 56Kbps operation is shown below. Note that the network is assumed to be an all digital network, with clear 64Kbps transmission between the Internet Service

Provider and the line card serving the subscriber. This restriction will be relaxed later but serves to make the description of the 56Kbps technique easier.

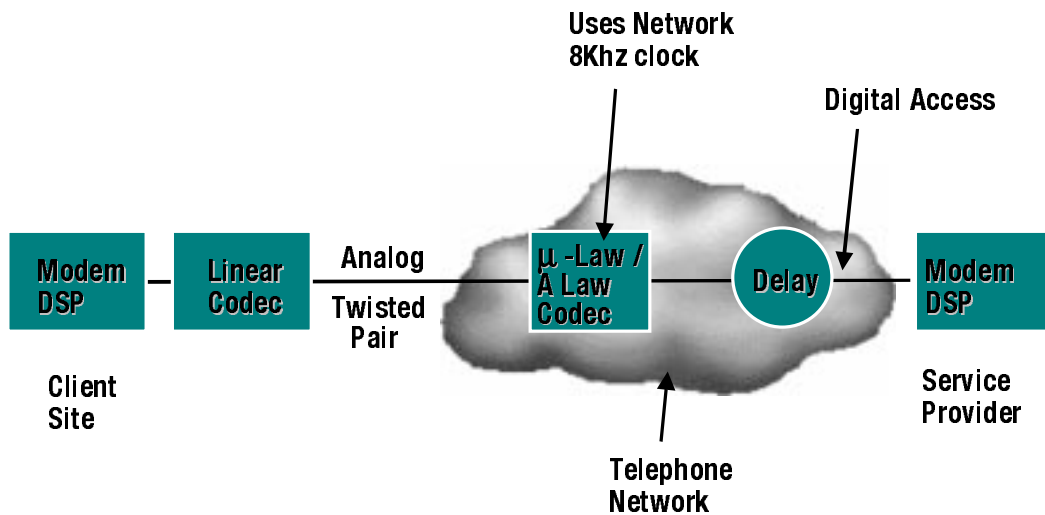


Figure 5: A schematic diagram of the all digital network, showing the service provider with a digital connection.

If we drill down to the next level of detail, we see that the line card consists of a codec, with a low pass filter between the codec and the copper line. The client modem contains a linear codec and a low pass filter, also, but since the modem is supplied to the customer, the accuracy of the codec and the filter characteristics can be optimized for the modem operation. See the figure below.

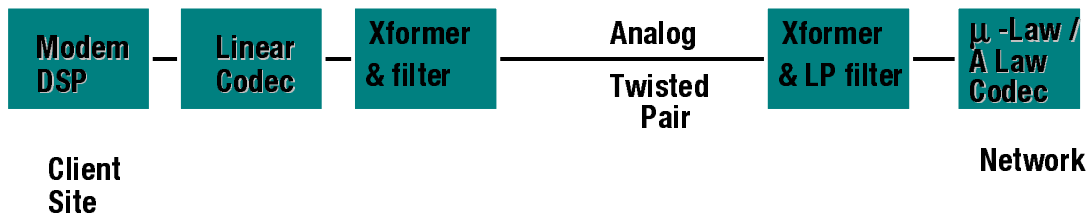


Figure 6: A schematic diagram of the elements in the path from the line card to the customer's modem.

Remember that the PAM technique is used only in the downstream direction, from the line card to the customer's modem. Traditional QAM techniques are used upstream from the customer's modem to the network.

The modem at the Internet Service Provider sends eight bit values to the line card, which generates a specific voltage for 125 microseconds, in response to each eight bit value. When these values represent the quantization values from the sampling of an analog signal, the resulting output is a series of voltage steps which follow the original waveform. Since these voltage steps are put through a low pass filter, the high frequency components are removed, thereby smoothing the voltage steps to a reasonable approximation of the original analog signal.

When PAM is used, however, the eight bit values sent by the ISP modem no longer represent the samples of an analog signal – they are values used as symbols. The codec in the network has the ability to generate 255 different voltage levels. Since the network sampling rate is 8,000 samples per second, 8,000 of these voltage levels will be generated each second. If we want to send 56,000 bits per second, how many quantization points do we need?

Using equation 2, we find:

$$N_s = 2^{\text{bps}/R_s} \quad (\text{Equation 2})$$

$$N_s = 2^{56,000/8,000}$$

$$N_s = 2^7$$

$$N_s = 128$$

Therefore, only 128 of the 255 quantization levels (approximately half) have to be used to transmit 56,000 bps. This modem technique can be described, therefore, as a 128PAM technique. When 56,000 bps cannot be achieved, a smaller number of quantization levels are utilized. For example, at 48,000 bps only 64 quantization levels are required ( $2^6$ ). Data rates between the rates which can be achieved with integer powers of 2 are accomplished with a technique called “fractional bit rates”. Using this technique, essentially any data rate can be implemented. Data rate steps should, therefore, be set based on the expected line impairment steps. Rockwell established data rate steps of 2,000 bits for its K56flex technology, compared to the 2,400 bit steps established in traditional modems.

PAM has been around for a long time and is well understood. ISDN BRI, for example, uses a technique called 2B1Q but is really a four level PAM. T1 lines uses a type of PAM called alternate mark inversion (AMI). So the technique used for 56Kbps modems is not new – only its application to consumer modems is new.

As mentioned above, the codec outputs a fixed voltage level for 125 microseconds (since the sampling rate is 8,000 times per second). This signal can be visualized as shown in the figure below.

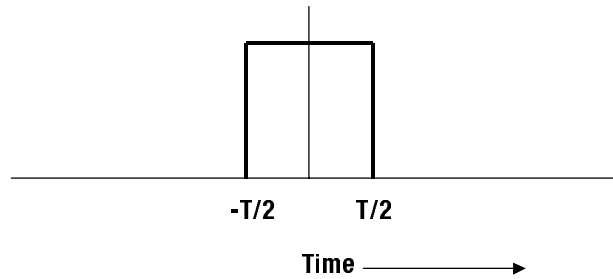


Figure 7: A PAM pulse of duration  $T$  (125 microseconds in our case).

What will happen to this pulse as it flows through the low pass filter between the codec and the copper line (actually the low pass filter is a part of the codec)? Since the higher frequencies will be filtered out, only the lower frequencies will be left. In particular, the low pass filter in the line card rolls off from -3 dB at 3400 Hz and is down to at least -14 dB at 4,000 Hz. This causes the pulse to take on a characteristic shape, crudely represented by the following figure.

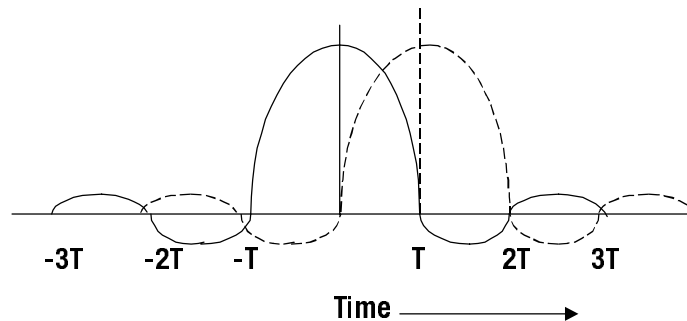


Figure 8: The pulse response of a band limited channel, specifically a channel with a low pass filter. This represents the signal which actually appears on the copper wire, after passing through the low pass filter.

The important thing is that the sample time for the subsequent pulse occurs when the signal from the first pulse is crossing the axis. If this does not occur, a problem known as intersymbol interference occurs.

In reality, of course, the signal on the line is the sum of the two signals shown (in the general case, it is the sum of the actual signals sent). But what will be the actual frequencies on the copper wire?

We know from Nyquist that we must sample at a rate at least twice the bandwidth of the signal we wish to reproduce. Nyquist's theorem also works in reverse. Given a sampling rate, we can not generate a useful signal with a frequency greater than half the sampling rate. Since the sampling rate is 8,000 times per second, the maximum usable frequency of the resulting signal on

the analog line is 4,000 Hz<sup>3</sup>. PAM, therefore, is quite bandwidth efficient, providing a minimum of two symbols per Hz of bandwidth. Note that QAM only provides one symbol per Hz of bandwidth. This, of course, is one of the reasons we can run faster than with QAM techniques.

The second reason we can achieve higher speeds than with QAM is that we eliminate quantization noise from the network codec. We do this by using the codec's quantization levels as the voltage levels representing the symbols. The only quantization noise left is in the codec in the customer's modem.

The codec used in the customer's modem is a linear codec. The equation for quantization noise in a linear codec is:

$$\text{SQR} = 1.76 + 6.02n + \log_{10} (A/A_{\max}) \quad \text{Equation 7}$$

Where:

SQR = signal to quantization error ratio

n = number of bits of sampling accuracy in the codec

$A/A_{\max}$  = The voltage level of the signal divided by the maximum voltage level possible (rail to rail voltage).

For full range signals, each bit of accuracy in the codec represents about 6 dB of quantization noise floor. Thus, a 16 bit codec (commonly used for modems) provides about a 98 dB noise floor for full range signals. This is such a high noise floor that quantization error is essentially removed as a significant impairment. For 56Kbps operation, other types of impairments become dominant and limit the data rate on the channel.

But how much SNR does a 56 Kbps modem need? Let's look to Shannon's theorem again to find out. Let's start with the modified version of Shannon's theorem which we derived as equation 6 and solve it for SNR dB.

$$\text{bps} = \text{BW} \log_2 (1 + 10^{\text{dB}/10}) \quad \text{(Equation 6)}$$

$$\text{bps/BW} = \log_2 (1 + 10^{\text{dB}/10})$$

$$2^{\text{bps/BW}} = 1 + 10^{\text{dB}/10}$$

$$10^{\text{dB}/10} = 2^{\text{bps/BW}} - 1$$

$$\text{dB}/10 = \log_{10} (2^{\text{bps/BW}} - 1)$$

---

<sup>3</sup> In the absence of the low pass filter the maximum frequency on the copper wire would be a great deal higher than 4 KHz, of course, since the output of the codec is a series of square pulses, each 125 microseconds long. And square waves contain very high frequencies compared to the fundamental frequency. Nyquist, however, showed that all frequencies beyond one half the sampling rate are "redundant" (Nyquist's word) and can be removed without affecting our ability to recover the encoded information. In our case, all frequencies higher than 4 KHz can be removed.

$$\text{dB} = 10 \log_{10} (2^{\text{bps}/\text{BW}} - 1) \quad \text{Equation 8}$$

The bandwidth of the channel is not 4,000 Hz because low frequencies must be avoided because of the hybrid transformers and the high non-linear distortion in the transformer at low frequencies. Let's assume that we avoid the frequency band from zero to 200 Hz. This gives us a bandwidth of 3800 Hz. Substituting into equation 8 gives:

$$\text{dB} = 10 \log_{10} (2^{56000/3800} - 1)$$

$$\text{dB} = 10 \log_{10} (2^{56000/3800} - 1)$$

$$\text{dB} = 10 \log_{10} (2^{14.74} - 1)$$

$$\text{dB} > 44.36$$

Thus, the signal to noise ratio on the line must exceed 45 dB in order to achieve 56 Kbps. This is achievable on real lines.

A problem, however, is the limitation of the transmit signal to -12 dBm in the United States. Note the last term of equation 7. This term reduces the SQR of the codec when the signal received by the codec is less than the maximum signal (also known as dynamic range). Each dB given up to dynamic range reduces the noise floor of the codec. For a 16 bit codec with an 98 dB noise floor, once more than about 53 dB is given up to dynamic range, it will no longer be possible to achieve 56 Kbps on the line due solely to the noise floor of the codec in the customer's modem. In reality, 53 dB of dynamic range is enormous – quantization noise is, in general, not the primary impairment on the line.

In order to meet the US FCC limit on transmit power, shell mapping techniques are used to limit the use of the outermost quantization points. However, the use of shell mapping increases the number of quantization points needed for 56 Kbps operation to about 144, further decreasing the decision region size.

This problem of a low received signal is only one of the problems encountered in the network. Other problems have to do with robbed bit signaling and digital pads. Robbed bit signaling is an older technique used on T1 lines in the network to convey call progress indications (logical dial tone, ring, busy, answer, etc.). When using this technique, the network “steals” the low order bit of every sixth voice sample, using this bit for call progress. The problems for the 56Kbps technology are to identify the fact that a connection includes links which utilize robbed bit signaling and then to accommodate it. One way to accommodate this problem is the use of two quantization points (one whose eight bit digital value has a zero in the least significant bit position and one with a one in the same location, with the rest of the bits being the same) to indicate one symbol. Of course, since two quantization points are used to indicate one symbol, this reduces the number of quantization points available and reduces the data rate. However, since this only occurs every sixth frame, the overall effect on the data rate is reduced.

When a telephone call is made, the telephone switch commands the line card to insert some attenuation into the call to reduce the amount of feedback and echo. Common amounts of attenuation are 0 dB (none), 3 dB, and 6 dB. Most line cards in the network achieve this attenuation in the analog domain, by reducing the drive on the line. However, there's a new generation of line cards which utilize a digital signal processor to achieve this attenuation. It does this by converting the signal from the  $\mu$ -law technique used in the network to linear, so that the attenuation can be done easily. The signal must be converted back to  $\mu$ -law and this is where the problem occurs.

When the attenuation is done in the analog domain, all of the signals maintain their relative relationships. Although the signals are attenuated, the relationship remains. When the signals go through the digital pad, they are assigned to the closest  $\mu$ -law point during the linear to  $\mu$ -law conversion phase. This changes the relationship between signals, making it harder to decode at the receiver.

Additionally, when the conversion from linear to  $\mu$ -law is done, some of the points fall half-way between two of the  $\mu$ -law quantization points. Codecs from some companies assign "up" while those from other companies assign "down". This adds additional ambiguity in the decode process at the receiver.

Just like with robbed bit signaling, the problems are how to detect the presence of digital pads and how to accommodate them after they are detected. To operate in the presence of a digital pad, a constellation must be defined which maintains unique points after being processed by the digital pad, i.e., no two transmit constellation points can map to the same point in the resulting constellation after the signal is passed through the digital pad (the received constellation).

As mentioned above, the problem is that different digital pads operate differently so the modems must probe through the pad to determine how candidate constellation points will be processed prior to selecting the final constellation. This is easier said than done with all the variations in digital pads in the network.

## **Into the Future**

It is possible to use PAM in the upstream direction as well as downstream. However, it's a much more difficult problem. To use PAM in the upstream direction, the client modem must determine the characteristics of the line and pre-distort the PAM signals so that they are "equalized" by the filter characteristics of the line. Additionally, the signals must be sent so that they arrive at the codec's analog to digital converter at the right time, synchronized with the network clock.

Although these are formidable problems, they can be overcome and upstream rates above 28.8 Kbps are possible. It is unlikely, however, that the upstream rates will ever equal the downstream rates so asymmetrical operation will continue to be the norm.

### Summary

By now, most people who are interested in how the 56Kbps technique works have read the initial white paper published by Rockwell shortly after the announcement of this technology. That initial white paper provided an intuitive understanding of the technology but did not get into the specifics of how the technology works nor discuss in much detail the network requirements for this technology to operate.

This paper has attempted to go to the next level of detail. For those who wish to go even deeper, I recommend the V.90 specification which covers the complete operation of the link layer portion of the modem.

For a good general description of how the digital telephone network “works”, see John Bellamy’s book *Digital Telephony*, published by Wiley, second edition, 1991.

About the Author: **Mike Henderson** is Division Director, Business and Technology Planning for Network Access products for Rockwell Semiconductor Systems. After joining RSS in 1996, he managed the marketing of central site modems and later, central site ADSL. Prior to joining RSS, Mike spent over 17 years with AT&T where he managed engineering development, most recently of central site modems. He received his BSEE from LSU and his MBA from Nova Southeastern. He is an inventor on 19 patents and patent applications.

**Document End**