SiliconGraphics
Computer Systems

PDH, Broadband ISDN, ATM, and All That:

A Guide to Modern WAN Networking, and How it Evolved.

by

Paul Reilly
MSD Marketing
Silicon Graphics, Inc.

April 10, 1994

SUMMARY: This white paper looks into the wonderful world of ATM and Broadband ISDN. The intent of this paper is to explain these emerging technologies, how they evolved, and explore the impact they might have on the computer industry. Various related technologies such as X.25, ISDN, and PDH are also covered. While it is assumed that you have some knowledge of LAN computer networking, we have tried to keep this paper focused on the non-expert.

Acknowledgments:

We wish to thank the following people who have contributed to this white paper (in alphabetic order):

Nelson Bolyard, Scott Bovenizer, Greg Chesson, John Talbott, Jon Thompson, and Rob Warnock.

PDH, Broadband ISDN,ATM, and All That:
A Guide to Modern WAN Networking, and How it Evolved.

Silicon Graphics, Inc.
Mountain View, California

Section 1 Introduction.

Many people think that Broadband Integrated Services Digital Network (B-ISDN), better known as the *Information Superhighway*, is a computer networking system. It isn't. B-ISDN is not a computer network but the telephone network. Specifically, it is the worldwide digital telephone system currently being installed in virtually every developed country of the world. Once completed, B-ISDN will permit its users to communicate over high-quality, high-speed digital communication channels. The supported media include telex, fax, voice telephone, video telephone, audio, high definition TV (HDTV), and, of course, computer networking. Please note, however, that computer networking is just one of the media and, in many ways, only a minor part of B-ISDN.

It's our plan to give you an overview of B-ISDN, how it developed, and how it works. Then we will explore some of the services that B-ISDN offers that are of particular interest to the computer user. Most specifically, we will look at ATM.

We also realize that the audience for this white paper varies from the neophyte to grandmaster networking guru. While it would be best to have a number of versions of this white paper, each oriented toward a specific range of expertise, such an enterprise is not practical. We are therefore forced to write this paper to the lowest common denominator, the neophyte. This, naturally, causes us to present material many of you already know and understand (how analog and digital transmission works is a good example). What we ask of you is your patience and that you simply skip those sections which are either of no interest or you already know.

In addition, we also recognize that many of you are merely interested in, say, ATM. Please feel free to just go to the appropriate sections of interest and then backfill your reading as necessary or as your interest develops.

For those of you that have the time, we do suggest that you start at the beginning and work your way through this tome. It is organized chronologically, giving you a historical perspective to how WAN (that is to say, the telephone system) technology evolved. The reason we did this is to explain the why as much as the what and how.

Although it might be tempting to cover GOSIP, DCE and a number of other related topics, we will merely look at B-ISDN as the Physical and Data Link layers of the ISO seven-layer model. The other material will be covered in later white papers. Likewise, we will ignore Frame Relay and SMDS. While they can be used to carry computer communications, they are transparent to both the computer and the user.

However, before we get to all the exciting stuff, we need to first clarify some definitions. These include ISDN vs. B-ISDN, B-ISDN vs. ATM, LAN vs. WAN and Connection Oriented vs. Connectionless communications. Knowledgable readers may safely skip this.

1.1 ISDN vs. B-ISDN

We should start with these two acronyms if for no other reason than to point out that they are very different. ISDN, Integrated Services Digital Network, is about ten years old and the forerunner of B-ISDN. (Nowadays, it had become fashionable to refer to ISDN as N-ISDN or Narrowband ISDN. We will use "ISDN" for narrowband ISDN and "B-ISDN" will be used in reference to the broadband ISDN.)

While both are digital telephony standards, ISDN was designed to utilize the preexisting copper wiring that runs from the telephone exchange to our present day analog telephones in order to bring the digital telephone system into our offices and homes, right up to and into the actual telephone.

While ISDN is deployed in Europe, particularly in Germany and France, it has recently been all but superseded by its fiber optic brother, B-ISDN. The reason is speed. Whereas ISDN offers data rates of perhaps hundreds of kilobits per second, B-ISDN gives the user data rates ranging from hundreds of megabits per second to more than two gigabits per second. Clearly, B-ISDN has stolen the limelight from ISDN and has the attention of the world's telephone companies as well as most computer users. It is already colloquially known as the Information Superhighway by many of its potential users and more formally as SONET in North America and SDH in the rest of the world.

1.2 B-ISDN vs. ATM

Another frequently confused pair of terms are B-ISDN and ATM. As noted above, B-ISDN is the so-called Information Superhighway our politicians like to talk about. ATM, Asynchronous Transfer Mode, is often confused with B-ISDN. In fact, ATM is simply a service that can run over B-ISDN. It is literally little more than the specification for a 48-byte packet or cell of information with a five-byte header which tells the telephone system where that packet is going. It can run over a number of different physical media, ranging from UTP (unshielded twisted pair), to something called TAXI, to the B-ISDN superhighway. Think of ATM cells as a string of pickup trucks, each carrying 48 bytes of data, driving over the B-ISDN superhighway, and you will have a fundamental understanding of both concepts.

1.3 SONET and SDH

Although we'll get into these two topics in greater detail later, perhaps a few words are in order here. Both these terms refer to the actual specifications for B-ISDN. Synchronous Digital Hierarchy, or SDH, is the international standard approved by a standards group called the CCITT. Oddly enough, SDH started out as an American proposal generated by Bellcore, who called it Synchronous Optical NETwork, or SONET. Unfortunately, SONET and SDH are not really the same as each has features the other lacks. The reason for this is that until the B-ISDN effort got rolling in the late 1980s, the European and American telephone companies pretty much ignored each other's telephony standards; thus there is a lot of preexisting equipment that each group has that must be accommodated.

Fortunately, there is a subset of mutually compatible specifications which permit the American SONET to interoperate smoothly and nearly effortlessly with the European SDH. Most specifically, these are the specifications for OC-3, OC-12 and OC-48 in America; or STM-1, STM-4 and STM-16 in Europe. The good news is that virtually every implementation of either SDH or SONET complies to these three specifications, Thus, for all practical purposes, SONET and SDH are basically the same thing—if you stick to those three specifications.

1.4 LAN vs. WAN

We all know what a LAN is, don't we? Of course we do. It's a local area network, isn't it? Well, maybe.

For years many of us looked at the type of network and used that as the criterion for judging whether a particular network is local or wide area. For example, we all know that ethernet is a LAN. On the other hand X.25 is a WAN—right?

Perhaps once upon a time this might have been true, but it is not any longer. For example, we ourselves think nothing of using `rlogin` to login into a system literally on the other side of the world and then proceed to work on it as though it was in the same room with us. Obviously, since `rlogin` is a member of the TCP/IP services, and TCP/IP is a LAN protocol, we are working over a LAN. Or are we?

From one point of view, we were working on a LAN—an ethernet LAN to be specific. The magic that permits us to reach halfway across the world is something known as a bridge. Basically, a bridge takes packets for one type of network (i.e. TCP/IP on ethernet), encapsulates them into something else, transports the packets to another part of the world and finally converts them back to their original form and plops them onto another ethernet. Furthermore, it does all of this completely transparent to the user.

From another viewpoint, we were using a WAN although we never noticed it. If you don't believe us, just try to see how far you can get without the magic of the ethernet bridge. According to the ethernet specification, you should get perhaps 1,500 meters.

Conversely, everybody knows that X.25 is a WAN, yet a number of people routinely run it encapsulated in ethernet packets and fire then out onto an ethernet connecting a number of local systems. So is it a WAN or a LAN?

Okay, you might say, let's look at the type of wire and use that to classify the type of network. If it uses a phone wire or microwave or other communication service typically supplied by a company such as the telephone company, it's a WAN. Conversely, if the only physical media involved is an ethernet, FDDI cable, or locally installed twisted pair, then we are dealing with a LAN. And, to clarify the issue of the ethernet connection made halfway across the world via the magic of the WAN bridge, we'll simply call that a bridged LAN.

This classification system works, at least for the present. However, one of the likely long-term (perhaps in ten years or less) results of the introduction of B-ISDN will be the elimination of even this dichotomy.

1.5 Connectionless vs. Connection-Oriented Communications.

Probably no other concepts in networking cause more confusion than these because there is a continuum between the two. Let's look at three examples:

- You receive a letter from the tax people claiming you owe $200 in back taxes, plus $1,204.34 in interest and penalties. After a careful examination of your tax records, you find that they have made a mistake and so you write them a letter. You give full details, include photocopies of twenty documents, stuff it all into an envelope, address it to the tax people, suffix more than enough postage, and last but not least, you mail it. It's now two months later, and you still haven't heard back from them.
- The phone rings. You answer:

  "Hello."

  "Is this Jack?"

  "Speaking. And you are?"

  "Oh, I'm Tom."

  "Hi, Tom. How are you doing?"

  "Just fine. Listen, Jack, I want to ask you if you want to go bowling *buzz-snap-fizz. . . .*"

  "What did you say? We seem to have a bad connection."

  "Okay, I'll speak louder. I asked if you want to go bowling Friday night?"

  "Certainly. Meet you at the alley at 8 o'clock."

  "Great—see you then. Goodbye."

  "Have a good day, Tom."

  You hang up.
- Your fax machine senses the ring of an incoming call and takes its phone line off-hook. After exchanging several packets of information with the fax machine that made the call, your fax machine begins producing several sheets of an important document your attorney's fax machine is sending you. When the two fax machines are finished, they hang up.

1.5.1 Connectionless

The first example, the mailing of letters, is an example of connectionless communication. You send a letter, but there is no guarantee that it will be delivered. And unless you get a letter back saying that your letter was received, you don't know if it got to its intended recipient. Now let's extend the analogy and pretend that you've sent a letter every day for three days (let's assume that these are love letters). Not only is there no guarantee that they will arrive, but there is also no guarantee regarding what order the letters might arrive in. That is to say, the third letter might arrive first, the first letter might arrive two days after the first, and the second letter might even get lost.

There is nothing earth shaking in all this, this is the way the post service works, although it is highly probable that the letters will get to their intended recipient. Our sole point that there is no guarantee.

In the networking world, such "letters" are known as *datagrams*. They are generally "mailed" to one or more addressees. There is no way of knowing if the datagram got to any of the intended recipients unless they choose to send a reply. And if they don't choose to send a reply, it's the sender's responsibility to check. In the case of the letter you sent the tax people, we'd recommend a phone call to check that they got your letter.

In summary, connectionless communications use datagrams which are sent to one or more addresses. There is no guarantee of delivery or even in the order in which consecutive datagrams are delivered. A common example of datagram communications in the computer world is the UDP (Universal Datagram Protocol) which is used by NFS.

1.5.2 Connection-Oriented.

A telephone call is the classic example of a connection-oriented protocol. First, a connection is made between two parties. Unless that is accomplished, communication cannot occur. That is to say, you have to pick up the phone and say "Hello" when the phone rings.

Second, there is an exchange of packets, which in our example are sentences, giving a step-by-step acknowledgment that the communication is occurring between the intended parties. If one of the parties has nothing to say, they usually make some sort of sound like a grunt to indicate that they are still listening.

Since computers cannot speak like humans, they use sequence numbers instead. Typically, each connection-oriented packet is numbered, and the recipient acknowledges receiving them either individually or in groups. In our telephone example, the two participants tended to exchange sentences and so indicated to each other that they received the last packet sent to them. However, one party could have just as easily spoken at length while the other only occasionally made a sound (i.e. "huh-hah") to indicate that he was still listening.

The telephone call also shows an example of error recovery. When there was a line noise, the listening party asked the sender to repeat what he had said. Retransmission occurred almost immediately. This is also typical of connection-oriented communications.

Generally, connection-oriented communications are called *reliable* for these reasons: The information in transferred only when a connection between the two (or more) active parties is known to exist, and the packets are exchanged in such a way that the order is preserved and quick error recovery is possible.

1.5.3 The Gray Area in Between.

The example of the fax machines is arguably both connection-oriented and connectionless. Obviously, a connection was made when the two fax machines established communications with each other. However, there is no guarantee that the document was successfully reproduced on your fax machine (for one thing, it may have run out of paper). And even if you had one of the newest fax machines that does in fact report back to the sender that document was properly received, there is no reason to believe that someone didn’t come along and accidently pick up your fax and throw it away. It’s not until you send a reply to your attorney that you have the contract that he can be certain that you have it. This feature is clearly connectionless. Therefore, this example had features which are connection-oriented and other features which are clearly connectionless.

1.5.4 Summary.

While it is convenient to speak only of connectionless and connection-oriented communications, it’s important to know that some forms of communications have aspects of both. However, we will tend to use the two terms in a more or less black and white manner simply because it is convenient. Generally, we will either use the terms themselves or the word “datagram” to indicate a connectionless communication. From time to time, we will use the word “call” the same way you might use “telephone call.” In this case we will be referring to connection-oriented communication.

Section 2 The History of B-ISDN: How it all began.

It has been said that to understand B-ISDN, one must understand how the global telephone system developed. Such a history could fill volumes and most of it would be as boring as watching grass grow. Hence, we will merely touch on a few highlights, paying attention to only those events and technologies that play a specific role in the evolution of B-ISDN.

(In this section and the next, we will review the development of our present telephone system. Unless you have some knowledge of modern telephony, we urge you to read this section for it sets the stage for B-ISDN for it evolved from the telephone system first built about 100 years ago. If you have a fairly good background in telephony, by all means skip this two sections. However, if you have no idea what telephony is all about, we suggest that you read the following. We have tried hard to make it understandable and easy to read.)

2.1 How It All Began.

It all began with the first telephone call. If the history books are correct, the grand event occurred one day in 1879 when Alexander Graham Bell was working in his makeshift laboratory in the Palace Hotel. He reportedly spilled some battery acid on himself and said, “Come quickly, Watson; I want you.”

The real question is was this, in fact, the first telephone call? We hold that it was not. What Alexander Bell actually did was to demonstrate a practical usage of his invention as an intercom.

An intercom is a point-to-point communications device that can assume that whomever you want talk to is sitting right next to it. Thus, all you have to do is speak. On the other hand, the telephone can be switched to any of a large number of subscribers. This requires the ability to signal that you want somebody to pay attention to the device. Neither switching nor signaling are truly necessary for an intercom—although we agree that both could be useful.

Therefore, the telephone has to have not only a microphone and speaker, but also the ability to selectively connect to any of many other similar devices (switching) and the ability to attract the attention of those at the other end so that we can establish a connection-oriented communication.

2.2 The Development of Signaling.

The first telephones weren’t switched. They were party lines. To place a call, one merely turned a crank connected to a little dynamo built into the telephone to generate a series of long and short ringing patterns. The idea was that only Mrs. Jones, who had the ringing pattern of two longs followed by three shorts, was supposed to answer that pattern.

Fairly soon, you talked to the operator instead. What you did was say something like, “ELmwood 5-3489, please,” or if you were in a rural area, you might simply ask, “Can I talk to Martha?” As we shall see, both of these are examples of in-band signaling.

The next step was that operator physically connected your telephone line to Martha's and made Martha's telephone ring, thereby signaling the end-recipient that you desired to establish a connection-oriented communication.

The next step was the invention of the long-distance phone call. It is basically a logical extension of the local phone call. Instead of asking for "ELmwood 5-3489" you might say, "Give me GIbert 7-3967 in New York City." The operator would politely ask you to wait while she placed the call.

Typically, you never heard what really happened because the operator temporarily disconnected you, although she left your line active so nobody else could call you. Then she started hunting for a path from your local exchange (i.e. switching station) to the GI7 exchange in New York City. This was usually done by her calling an exchange between yours and the one in New York City and asking an operator there to find a line to an exchange even closer to your target. The process was repeated until either the operators found a path or ran out of possibilities. The vernacular used for this path is *circuit*. This is still used today, particularly when we're told that "All circuits are busy."

2.3 Transoceanic Telephony.

The first transatlantic telephone call was made in 1927. Since it wasn't yet practical to install vacuum tube amplifiers in underwater cables for a number of years yet, the phone call was actually a radio-telephone call. It was a milestone for a number a reasons. First, people on either side of the Atlantic Ocean were able to talk to one another. The second issue was that the telephone companies in America and the Post Telegraph and Telephone (PTT) organizations in Europe discovered that they had a problem: Their equipment was incompatible. They had little choice but to design and install conversion equipment at great expense. By the late 1930s, this had become more than a minor problem. However, World War II intervened before anybody could do anything about it.

When peace returned to the world, the United Nations became a center of international diplomacy. One of the many agencies of the UN is the International Telecommunication Union (ITU). One of the more important committees of the ITU is the CCITT or Comité Consultatif International Télégraphique et Téléphonique. It is also known at the International Telegraph and Telephone Consultative Committee, but it is always abbreviated as CCITT.

The CCITT, which was chartered to unscramble the mess that the world's telephone companies had gotten themselves into, has so far spent the better part of fifty years working on the problem. By and large it has succeeded. B-ISDN and its smaller, older brother, ISDN, are its crowning achievements.

However, before we can get into B-ISDN, we're going to have to stop and look at how the modern telephone system works.

Section 3 Modern Telephony.

Although the telephone companies of the world have massive investments in equipment, they have gone through a number of almost revolutionary changes. At first, equipment was designed and constructed to last for thirty to fifty years. Now the expected life-span is much less. The reason is technical advances. And if anything, the time between waves of new technology is decreasing.

This puts the telephone companies of the world into quite a bind: Capital investment made a few years earlier would have to be written off prematurely if the telephone companies leaped on every new technological breakthrough that came along. Yet they have to keep up with customer needs.

In 1989, AT&T, the largest long-distance carrier in the United States, took a quarterly loss of several billion dollars when they prematurely wrote off most of their analog telephone transmission equipment. The question is why? The answer is that it became obvious to the management of AT&T that digital telephony was the only way to go. This decision was certainly not lightly made. In fact, it took well over ten years of debate before it was finally taken.

Since that time, long-distance telephony has been based on digital technology. Let’s look at the reasons why: Basically, they are our old friends switching and signaling, but most of all noise. In order to understand that, we will first look at the analog telephone system and its shortcomings. Then we’ll examine present-day digital telephony—and its shortcomings.

3.1 The History of Analog Telephony.

In many ways, it can be argued that the origins of digital telephony are to be found in the telegraph. This device was clearly digital, using a code of long and short clicks to indicate the various letters of the alphabet. However, it had a very narrow bandwidth—perhaps less than 100 characters a minute.

By comparison, the first telephone systems were all analog. That is to say they used fluctuations in electrical voltages to transmit and reproduce the sound of a human voice and other sounds. The bandwidth of such analog signals were, at the time, many times greater than that of a digital system—at least several thousand times, in fact. There was no comparison between the two media. Analog was clearly superior: The telephone was many times faster than the telegraph.

It wasn’t long before the old telegraph lines were replaced with analog wiring and the telegraph key gave way to the then modern teletype machine. Although by nature digital devices, the teletype machines nevertheless transmitted their signals as a series of tones. The device that converted the digital information into analog tones is known as a MOdulator/DEMdulator, or modem. The modem is still with us. It is commonly used even today to connect remote users to computers as well as computers to each other via telephone lines.

Gradually, as the complexity of the analog telephone system increased, the telephone companies began spending ever larger sums of money on looking for ways to improve service while cutting costs. The most notable of these efforts was the countless man-years of research done at the Bell Laboratories. Many major basic technologies such as the transistor were invented there. The Bell Labs also gave us UNIX®. Less well known is that they also gave use what can only be called the narrow-bandwidth analog telephone.

Although never described as such, the modern analog telephone is a classic example of minimalist design. For one thing, the researchers at the Bell Laboratories discovered that only a narrow part of the human range of hearing is needed for voice communication. That range is generally given as between 300 to 3,400 Hz.

At first, this information was used to design cheaper and better telephones, as well as specifying the cabling running between the subscriber's house and the central office (this known as the *local loop*). Then it had a major impact on the design of long-distance telephony.

The lines (called trunk lines) connecting the various telephone switching centers were capable of carrying much larger bandwidths than required for a single phone conversation. So, eventually, somebody asked if they could find a way of putting several phone calls on a single physical wire? The good folks working for the telephone companies set to work and they did exactly that.

The technique they used was frequency division multiplexing (FDM). It works basically the same way your car radio does. Each of the individual telephone calls had its own private carrier frequency and was known as a circuit. The frequency of the spoken word is electronically added to a fixed carrier frequency, and the composite of the two is then transmitted. Since the bandwidth of sound needed to have clearly intelligible speech is only about 4,000 Hz, this meant that a number of different carrier frequencies could be spaced fairly closely together and so permit the cable to carry a large number of individual messages.

However, there were downsides as well. One was that it was difficult to tease out just one circuit from the many mixed together on a FDM multiplexed trunk line. The only effective way to do this was to demultiplex all of the circuits on an incoming trunk line and convert them back to their audible frequencies, then physically switch each though some sort of switchboard. This made switching a large number of analog circuits expensive. And it also introduced noise. Both these issues were major driving forces in the development of digital transmission.

By the late 1950s, it was clear to many telephone engineers that analog transmission had had its day. Unfortunately, not everybody agreed, and so a titanic struggle developed. However, in time, those pushing digital telephony eventually won.

3.2 Digital Telephony—The Basics.

Some of you may want to skip this section, but keeping to our promise to focus on the non-expert, we will start this section on digital telephony by clearly explaining the differences between digital and analog technology.

Both analog and digital telephone systems use variations in voltage levels to transmit information. The major difference is that analog system uses the entire continuum of the variations in voltages to indicate the signal whereas digital uses only specific ranges of values. That is to say in an analog system, the value of 1.234 volts would mean a slightly larger value than 1.231 volts. On the other hand, a digital system might consider the voltage range of +0.5 volts to -0.5 volts to indicate the binary value of zero, while any voltage greater than +1.0 would be considered equivalent to the binary value of one. In this particular digital encoding system, voltages between +1.0 and +0.5 would be illegal and indicate that a hardware problem had occurred.

(There are many ways of encoding digital data; this is just one. And please don't assume that this is the way the telephone companies do it. It isn't. We chose a relatively simple technique for example's sake.)

Another difference between the two systems is that the time line for analog is continuous while in digital transmission the values are measured only at specific moments in time. In digital telephony, the voltage is sampled at these times only and the voltages levels in between are ignored. The value derived for each point is then transmitted over the circuit as a binary number.

Sounds a little confusing, doesn't it? Well, now that you've been through the verbiage, Let's look at a simple example in figure 1.0. The top part of this figure, figure 1.1, is what the original sound or tone we are transmitting would look like on an oscilloscope. If we were to send this tone over an analog telephone and measure the voltage on the wire, we would see something like figure 1.2. The amplitude of the signal may have been changed to some degree, but it clearly looks like the original waveform. In this particular example, the amplitude was reduced, but it could easily be restored by amplifying it. Otherwise, the analog signal looks like the original.

The procedure for encoding the sound for digital transmission is far more complex. First, the amplitude of the original signal is sampled several times per cycle (in this example, at least) and converted from an analog voltage to a digital number representing the value of that voltage at the instant in time the sample is taken. These points are indicated in figure 1.3 by the vertical bars; the digital values are displayed underneath. For convenience, we have taken these to be +12 volts and -8 volts.

Next, these digital values are converted into a signed 8-bit binary code in which the left-hand most bit is the sign bit which is *on* if the number is negative and *off* if positive, while the remaining seven bits are use to present the number itself. Thus +12 would be 00001100 and -8 would be represented as 10001000. (Those of you who know a bit about computers might note that these are not two's complement representation but simple signed numbers. There is a reason why we did this, but more of that in a moment.)

Finally, this series of binary digits is transmitted over the wire as a series of pulses. If we were to connect the oscilloscope to this wire, we would see something like that shown in figure 1.4. We would see a series of voltage changes that indicate the 0's and 1's of the binary numbers. Please note that a series of 0's are indicated by no voltage, while the 1's are indicated by the positive voltages.

Figure 1 Conversion of a Sound into Analog or Digital Transmission Formats.

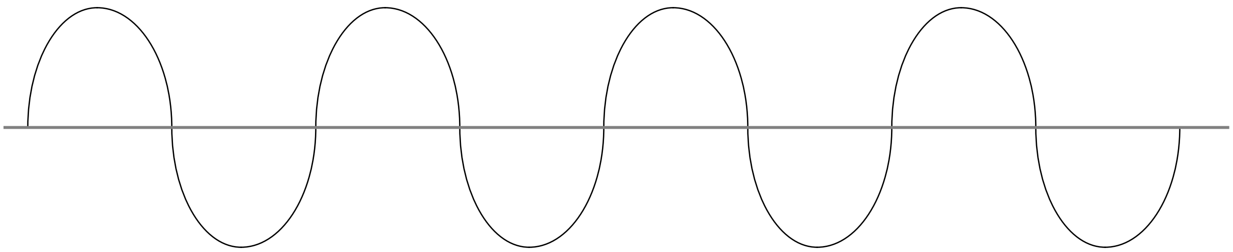

Figure 1.1 Original Sound.

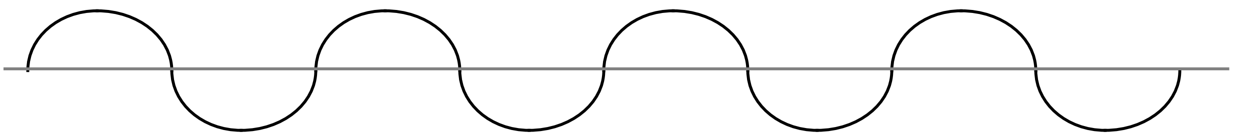

Figure 1.2 Analog Transmission.

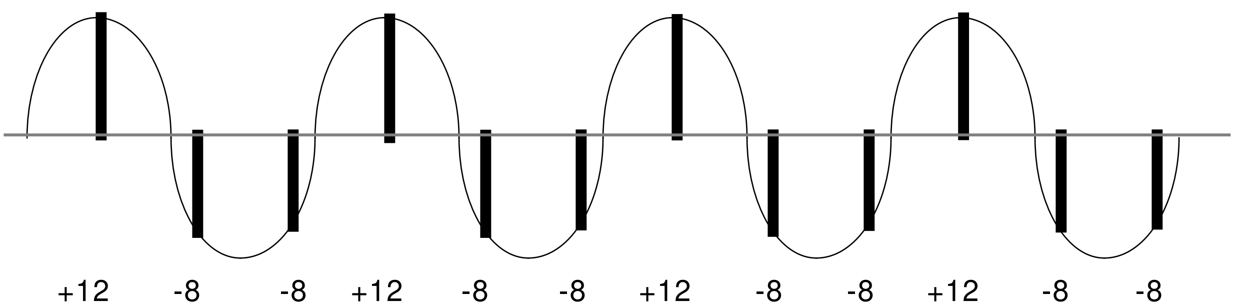

Figure 1.3 Digital Sampling Points.

Figure 1.4 Digital Transmission.

In order to spot the difference between one, two, and three 0's in a row, you have to measure how long the digital signal remains at zero volts. Likewise, in order to know difference between one and two 1's, you also need to time how long the signal remained positive. Obviously, you need a clock to do this. This is a critical part of the design of a digital signal. As we shall see, the clock used in digital telephony is 8,000 Hz.

3.3 Digital Telephony—The Sampling Theory.

What we have described in section 3.2 is how Analog-to-Digital conversion works. It is generally abbreviated as A/D. The reverse process is called Digital-to-Analog, or D/A. (We will not go into how D/A conversion works. If interested, you can look it up in any of a number of basic electronics books.)

The important issue we do need to understand is how A/D conversion permits us to reconstruct the original tone. The first step to understanding that is identifying what is it we need to convert—and that is the dynamic frequency range of human speech actually used in a telephone call.

Okay, time for a ten second quiz. What was the dynamic frequency range of human speech in a telephone conversation?

No fair peeking.

Give up? Well, okay, it's approximately between 300 and 3,400 hertz.

Next, we move to the issue of how many samples per second we need to take in order to correctly measure that range of frequencies. The answer is something called the Nyquist value, which turns out to be slightly more than twice the highest frequency you wish to measure. (If you want to know more about this, we suggest that you find a good book on analog-to-digital conversion.)

What this means is that it is possible to reproduce accurately a human voice as transmitted over the telephone if you sample it at something a little more than 6,800 times a second. In fact, the telephone engineers chose 8,000. Please note that this is not 8,096 or 8K, but 8,000. This, by the way, is an extremely important number. The entire digital telephone system of the world runs at 8,000 cycles or frames per second.

Finally, the remaining question is just how little data do you have to take to measure the amplitude of each data point adequately? Here again, the telephone engineers came up with a clever answer. They treated the amplitude in a nonlinear manner, spacing the steps near the intensity of zero closer together and those at the higher intensities further apart. And since we actually hear the intensity of sound in a non-linear manner (i.e. in decibels which are logarithmic), this makes a lot of sense. The end result is that they found that they only needed seven bits for information plus a sign bit for a total of eight bits of data. This, by the way, is a signed binary number, not a two's complement number.

With that final bit of information, we can now answer the question of how much information you really need to reproduce the sounds of the human voice digitally? When we multiply eight bits per sample by 8,000, we find that we can make a digital telephone reproduce human speech as well if not better than the old analog telephone with 64,000 bits per second. Remember this number. Write it down on the back of your hand if need be. It is the basis of everything we will look at from here after.

3.4 Why Bother?

So far we have spent a number of pages on how a digital telephone works, and we even figured the bandwidth that would needed for you to talk to your auntie in Peoria. The nagging question is why bother? After all, the analog telephone worked great for at least fifty years, so why bother to change it?

The primary answer is noise.

3.4.1 Noise Reduction in Digital Transmissions.

The major problem with analog transmission is that once noise got into the signal, it was almost impossible to get it out. And noise leaked in all over the place. Sometimes there was a poor connection. Other times we heard two or three other conversations as well as our own. These were caused by cross talk between adjacent pairs of wires. In addition, human speech was slightly distorted each time the signal was amplified, and since this had to occur every 800 miles or so, very long-distance phone calls often became very distorted.

The major reason why noise was such a problem in the analog system was because it was impossible to tell which part of the signal was human speech and which part was noise or distortion. As for distortion, who knew what the original speech really sounded like? Each and every one of us has our unique patterns of enunciation, pitch, and tonality. For all the telephone engineer working on the problem knew, maybe the customer really did sound like a chipmunk. Generally any attempt to reduce noise and distortion once it got into the analog transmission system usually led to even worse problems.

On the other hand, it is easy to reconstruct a digital signal even if it was badly distorted by noise. Let's look at figure 2 to see why. At the top of figure 2, we have figure 2.1 which is nothing more than a repeat of figure 1.3. (We already had the artwork, if you can call it that, and so reused it.) What we see is a series of +12, -8, -8, sequences being transmitted digitally. All the zeros go out as exactly zero volts, and all the ones are transmitted as two volts. This is a highly idealized representation, for the pluses are never so neat.

We have a close up of the first two values in figure 2.1 presented in figure 2.2. In this figure we see the actual voltages as well as a time line for the clock that is used to read these pulses. We have placed, for your convenience, the binary value of each pulse over the little vertical bar which indicates the strobe time at which the pulse is to be read.

Figure 2.3 is figure 2.2 after the pulses have traveled through a wire for some distance. The original pulses are still shown in gray, but the black line shows what arrived. All sorts of noise and distortion are now present in the signal, but

Figure 2 Resistance of Digital Transmission to Noise and Distortion.

Figure 2.1. Digital Transmission from Figure 1.3.

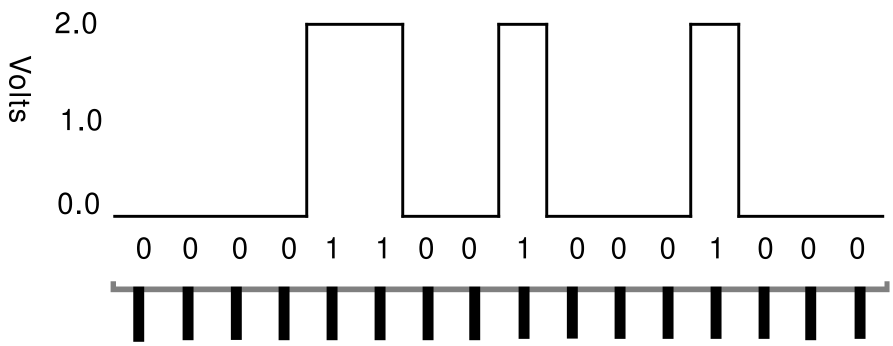

Figure 2.2. +12 and -8 as Transmitted Digitally.

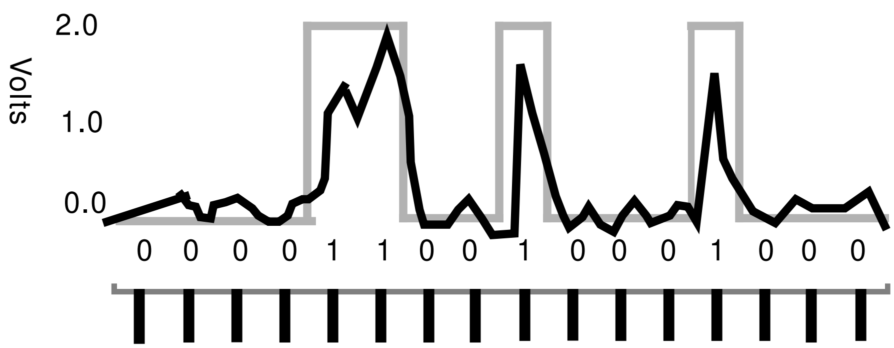

Figure 2.3. +12 and -8 as Received Digitally.

the original values of each binary digit can still be determined at the strobe times because those pulses that are supposed to be 1's still are greater than one volt, and those that are supposed to be 0's are still in the range of -0.5 to +0.5 volts.

Thus it is quite simple to remove noise from the digitalized signal by this technique. There are other tricks that can be used, but we need not worry about them. All we are interested in you understanding is why digital transmissions are so free of noise and distortions.

3.4.2 *Digital Switches are Cheaper.*

As we indicated in section 3.1, it's necessary to demux an analog phone call on one trunk line in order to transfer it to another. Actually, all the calls on the same incoming trunk line need to be demuxed, and then they must be switched individually to whatever trunk line they need to go to next.

The only reasonable way of switching an analog telephone call is by what is known as space division switching (SDS). In other words, each wire is plugged individually into the appropriate outgoing line. A good example of this is the old-time telephone switchboard we have seen in the movies—or perhaps in real life. More modern versions of this technology are the cross-bar and step-by-step switches that were used in telephone exchanges for many years. There was even something known as a panel switch, which is now quite obsolescent. All of these devices filled large rooms with many rows of relay racks, lots of cables and tons of test equipment.

This isn't to say that it was all relay equipment, for near the end of the analog era, electronics were developed to replace the clicking and clacking relays which can still be found in the oldest telephone exchanges. Naturally, electronics reduced the price of the analog switching; however, analog switches still remained expensive because you still had to handle each incoming line as a separate entity.

Digital transmissions are also multiplexed onto trunk lines, but instead of using FDM, digital transmissions can be easily time division multiplexed (TDM). In this technique several separate messages are mixed together according to a time slot. We will look into this technique in greater detail later, but for now imagine that six phone calls are being sent on the same wire with a time slot of one millisecond. Thus, for the first millisecond, several bits of the first phone call are transmitted. Then, in the second millisecond, several bits from the second phone call are sent. This would continue until a packet of bits was sent from all six calls. This brings us to the seventh one-millisecond slot. At this time more bits of the first phone call are sent. Figure 3.1 gives an pictorial example. In this figure we separated out the six phone calls that are being digitally transmitted for clarity. However, you should really think of them as little railroad cars following one another in a train as they head down the railroad track. This is what we did in Figure 3.2.

The actual switching of one call into or out of the train of time slots is fairly simple. All one needs to know is what time slot the call is in. Fortunately, this is readily available information. Figure 3.2 shows this diagrammatically. A trunk line is shown coming into a switch and the packets that make up the fifth telephone call (shown as the cross-hatching) are being copied out and replaced

Figure 3 How Time Division Multiplexing Works and How Circuits are Switched.

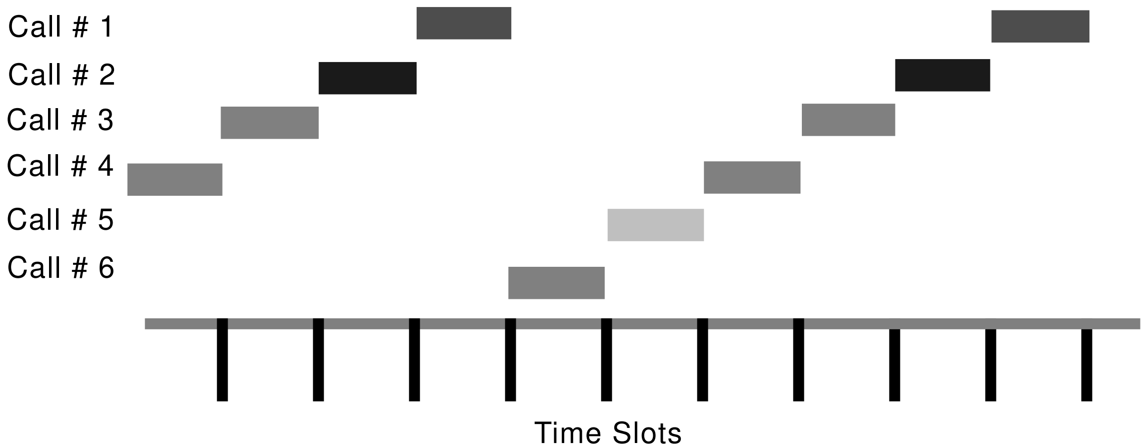

Figure 3.1 Time Division Multiplexing Six Telephone Calls on a Trunk Line.

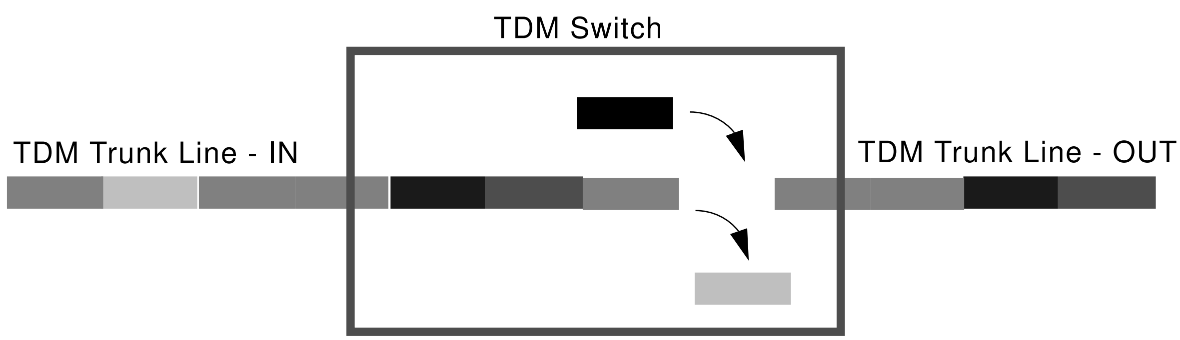

Figure 3.2 Switching out the fifth call on a TMD trunk line and replacing it with another call. This is an example of Synchronous Transfer Mode (STM) because the same call is always in the same time slot.

with a new telephone call (shown as solid black) that is headed in the same direction as the output side of the trunk line. This type of switching can be easily done electronically, and because the amount of stuff needed to build a TDM switch is much less, the equipment is cheaper.

There is a subtle point here as well; each phone call always uses the same slot. That is to say, it is synchronous. This technology is known by some as STM or Synchronous Transfer Mode. As we shall see, ATM, Asynchronous Transfer Mode, permits the same phone call to use different time slots and so is asynchronous. We shall return to this later.

3.4.3 *Digital Transmission is Cheaper.*

Another advantage of digital transmission is that the repeaters needed to amplify the signal are simpler. All they have to do is recognize the incoming pulses and then merely regenerated them. They do not have to amplify the original signal at all. This also reduces noise. It's a real win-win situation.

3.4.4 *Signaling is More Secure.*

Because we have repeatedly mentioned signaling as a primary issue in telephony, we should spend a little time looking at this function. It is the heart and soul of a telephone system. Without it nothing would happen, including billing the customer. Since we are going to need to know some of the terms associated with signaling, we will take advantage of this section to explain some of the more commonly used terms.

Originally, back when Mr. Bell was getting started, signaling consisted of turning a crank to get the operator's attention and then telling her to whom you wanted to talk. She, in turn, might have a chat with several other operators to piece together a circuit for a long-distance phone call.

This sort of signaling is called *in-band* and simply means that it happens in the bandwidth of the circuit. Virtually all signaling on analog telephony is done this way—even today because almost all of us are still using analog telephony for local connections. That is to say that the fancy touch tone telephone we have on desk still uses analog transmission. Presently, digital transmission is limited to long-distance phone calls and data transmissions. However, we're getting ahead of ourselves.

In the case of analog telephony, the operator was eventually replaced by sounds. The touch tone telephone does exactly this. We no longer tell the operator, "ELmwood 5-3489, please," but we instead tap out 355-3489 on the buttons. What we hear are a number of tones. Somewhere in the local telephone exchange, the electric device that replaced the operators listens and reacts. These tones are a classic example of in-band signaling.

During the early 1970s, when touch tone phones became all the rage, in-band signally was also used to place long-distance phone calls. A number of technically sharp individuals soon figured out what was going on and invented a new game called "Let's rip off the phone company." They called themselves

Phone Phreaks and had a great time making phone calls all over the world for free.They used tone generators to produce the same sequences of sounds that the switching equipment used for signaling.

In-band signaling, for long-distance phone calls at least, quickly came to an end. The telephone company switched to what is known as *in-channel out-of-band* signaling. They simply moved the tones they used for signaling to above 3,400 hertz, which, in theory, couldn't be heard on your telephone set. Thus the signaling was still *in-channel* because it was on the same circuit as your phone call, but out of the bandwidth of the actual call. In fact, digital telephony was in part designed to handle this. A few sections ago we explored the sampling rate needed to convert a phone call from analog to digital, and we noted that the telephone engineers chose 8,000 samples a second although only 6,800 or so were necessary. Why? Well, the reason was to permit the out-of-band signaling tones to be transmitted over the digitized links as well—or so they thought.

This out-of-band signaling seemed like a great idea, and it kept the Phone Phreaks a bay until one of them discovered that most telephones can pick up and transmit frequencies higher than 3,400 hertz. Thus the so-called out-of-band signaling actually turned out to still be in-band. Very shortly after that, the phone companies were being ripped off again. The only difference was that things were now done at a higher pitch.

Soon after that, digital transmission of long-distance signaling was changed. Most of it became digital. Although the transmission of tones for switching continued for a while, they were usually safely tucked away in another time slot on the TDM and thus in an different channel or circuit.

Moving the signaling out of the same circuit as the actual phone call is referred to as *out-of-channel* and comes in two flavors: One is called *associated* which means that it follows the same physical path as the actual phone call. The other is *non-associated* which means is follows a different path. Even today most signaling for local connections is still in-band and analog, while the signaling for long-distance calls in America and most of Europe and Japan is digital and over associated out-of-channel paths. Such signaling is often referred to as *common channel signaling*, because the switching information for a number of calls uses one channel. A good example is the European E-1 service which uses one of its 32 channels for this function.

3.5 *The Downside of Digital Telephony.*

So far we have done a superb job of telling you how great digital telephony is. Some of you might even be wondering why we still have analog telephones on our desks and in our homes. After all, digital telephony is cheaper, better, less subject to noise and distortion, and the phone companies can even control the fraud perpetrated by the Phone Phreaks.

Well, would you believe that three out of the four are true? We can't claim that digital telephony is cheaper. Unfortunately, digital telephony has two major problems to overcome: One is the last copper mile between your office or home and the local telephone exchange. This is known as the *local loop* in telephone jargon.

The other major problem is the cost of the telephones themselves.

Digital telephony is not cheaper if the subscriber's connection is involved because there are many millions of them. One estimate puts the cost of converting the line and replacing the telephone for each subscriber at $1,000. That's trillions of dollars. Even if it cost just $100 to convert an individual subscriber circuit to handle digital telephony, that's still many millions of dollars. Not even the telephone company can afford that. Thus, digital telephony is presently being used for long-distance connections only.

Originally, the cost of the A/D and D/A equipment was prohibitive for all but the most expensive services, the AT&T Long Lines that interconnected all the major cities of America. What happened at first was that your long-distance conversations were converted to digital only when they reached the regional switching centers or exchanges. Then, after a hop across country as a series of digital pulses, your conversation was converted back to analog long before it got to the local exchange that connected you to whomever you are calling. The conversation then completed its journey as an analog signal.

Over the years, the cost of the A/D and D/A equipment has become cheaper and cheaper. Today, it's not uncommon for digital transmission to be used over relatively short distances. Generally, digital is used for toll calls that use ATT, Sprint, or MCI long-distance lines. The same may be true for even a local call that leaves your local operating company's exchange.

In fact, the telephone companies have already completed most of the conversion to the fiber-optic transmission lines B-ISDN is based on. The conversion to fiber is all but complete—as far as long-distance telephony is concerned. All that remains is converting of the last copper mile of the local loop and the telephones in your office or home.

But it's also at least 80 percent of the cost because there are so darn many local loops and telephones. And how to handle this problem is yet to be figured out, even though there have been serious attempts to do so.

About ten years ago, the first attempt to convert the local loops to digital technology began. It was ISDN, now often referred to narrowband ISDN to differentiate it from B-ISDN. The major difference between ISDN and B-ISDN is that ISDN used the copper wire already in place. This meant that the bandwidth available into the home was roughly 128 kilobits. While ISDN can be found in Europe (mostly Germany and France) and some scattered locations in the United States, this effort has fairly much be stalled by the promise of a newer, better, cheaper, and faster technology—the fiber optic B-ISDN. This, unfortunately, has clouded the issue of how we're going to hook up the millions of homes and offices presently using the copper loop. This is certainly an undecided question but it will be worth billions to whoever solves it. We will speculate about this at the end of this paper.

Section 4 B-ISDN — The Raison d'Etre.

B-ISDN, for those who skipped the first three sections of this white paper, is not a computer network. It stands for Broadband Integrated Services Digital Network. The digital network they are talking about is the world-wide digital telephone network. It is also popularly known as the Information Superhighway.

Although computers can use this telephone network to talk to one another other, the real intent of B-ISDN is to build a digital telephone system for voice transmission—and high definition TV, movies on demand and other such fun things. That's where the money is—entertainment.

Therefore, to understand B-ISDN, we must look at what it was designed to do. As noted in the previous three sections, there are at least three major issues:

- Bring sanity to the international telephone arena. During the first fifty years of telephony, various telephone companies and all the Post, Telephone and Telegraph (PTT) organizations the world over each went their own way in the designing their equipment. Naturally, this led to a great deal of incompatibility. In the late 1940s, the CCITT was tasked with the mission of bringing compatibility to this industry. Its first efforts were directed toward developing international telecommunication standards for the analog equipment then in use. Since the advent of digital telephony, the CCITT has focused on bringing international standards into existence so that all the new equipment needed to build a worldwide digital telephone network is compatible.
- The second issue was to accomplish the first set of goals in a timely but yet economically reasonable manner. AT&T, for one, had no interest in once again prematurely writing off billions of dollars of equipment as they had to do in 1986 when they converted their long-distance lines to digital technology.
- And in the United States, at least, where most people now own their telephones, the new telephone system had to be made attractive enough to encourage many millions of people to scrap their present analog telephones and rush out and buy shiny new digital telephones for who knows how much each. This problem—replacing millions of telephones—also exists in the rest of the world as well. However, there is an interesting wrinkle—in many parts of the world the telephones are still owned by the PTTs. Not surprisingly, this policy is changing. Ownership of the telephones has suddenly become a liability and many PTTs are now permitting their subscribers to purchase their telephones.

However, none of these issues really concerns us as computer users. After all, we are simply interested in what B-ISDN can do for us. Having already reviewed the reasons why a digital telephone network is so desirable and how it works, let's focus on the computer telecommunications aspects of B-ISDN. First of all, we will have to look at some of the historical baggage that B-ISDN must bear. Remember that one goal of B-ISDN is to preserve as much of the equipment already in use as possible. As far as we're concerned, the two historical items of most interest to us are the digital transmission lines and the X.25 computer telecommunications protocols. Both of these greatly influenced B-ISDN's computer-oriented services.

4.1 The T1 Digital Transmission Lines.

T1, as it is popularly known in the United States, was developed in the early 1960s by AT&T and is still used to transmit voice communications digitally over long distances. The actual media used could be a pair of copper wires, coaxial cable, or microwave. Although the physical media may have varied, the fundamental service was for twenty-four concurrent voice conversations.

The way the data is transmitted is in frames, each taking 125 microseconds (or 1/8000th of a second). A frame is made up of one 8-bit digital sample from each of the 24 phone calls plus a single framing bit for a total of 193 bits per frame. Thus each frame is like a little train of 24 freight cars of 8-bit information and a one-bit caboose.

Nominally, this is 1.544 megabits per second. Given that the caboose or framing bit is not available to carry user information, the bandwidth is a maximum of 1.536 megabits. In reality, the actual portion that you might use could be much smaller, depending on the exact nature of the service. We'll get to that in the next section. However, for now, we want to talk about the big picture.

First, it's important to realize that a T1 service was not limited to 24 simultaneous voice-grade phone calls. Remember that the long-distance phone lines are in fact multiplexed, and, in the case of digital lines such as T1, the various 8-bit samples from each phone call are transmitted one after the other like little freight cars lined up one after the other in a train. There was nothing to prevent someone from using two, three, four or even all of the available 24 channels and transmitting other stuff like music, or even computer data over the same T1 line.

Naturally, it didn't take either the telephone companies or the big computer users (i.e. mainframe users at large corporations) to figure this out. Soon, T1 lines were being leased for large sums of money so that the users of such computers could quickly transfer large volumes of data back and forth across the face of the land.

Before long, even 1.536 megabits weren't enough. The big corporations were back demanding ever-larger volumes of traffic. Naturally, the telephone companies were more than willing to oblige. Since even T1 lines are themselves multiplexed between large population centers, the telephone companies were happy to start leasing these larger, higher-density cables. The most common is T3. There are other higher bandwidth services, but they are generally used only by the telephone companies. We'll get back to this in a paragraph or two, but first we need to make a slight detour—to Europe.

4.1.1 *E-1—the European Equivalent of T1.*

Meanwhile, in Europe CCITT was producing its own long-distance digital telephone service. As before, AT&T and CCITT went their own ways. There are many differences between the American T1 and what was essentially the European version of this service, know informally as E-1. Its official name is CEPT-1, but we'll use the informal name.

Although E-1 uses the same frame frequency—8,000 a second, it has more channels than its American equivalent. E-1 is based on 32 channels instead of 24 as in T1. Like T1, information is transmitted in frames, but they are 256 bits long. This gives E-1 a nominal bandwidth of 2.048 megabits per second. However, only 30 of the channels actually carry subscriber information. One channel is dedicated to signaling information (i.e. routing and billing information) and the other for timing (time-division multiplexing needs a clock. E-1 uses this channel while T1 uses the framing bit we mentioned.) Thus the actual bandwidth available to an E-1user is only 1.92 megabits per second.

As in the case of the American digital transmission system, E-1 is itself multiplexed into larger capacity cables. These follow the equivalent American T3 trunk lines but with proportionately larger bandwidths reflecting the extra channels of E-1. We'll look at the differences in a bit. (No pun intended.)

4.1.2 *The Digital Services, or What the Wire Carries.*

So far we have been using T1 and T3 to describe the digital transmission capabilities available in North America. We should be a bit more careful in using these terms for they refer to the physical media and not the format of the frames of information being transmitted over them. Strictly speaking, T1 is just the wire, not the service it carries.

Although it is perfectly acceptable to use the terms T1 and T3 when talking about these services informally, the correct name is Digital Service, or DS. There is a convenient correlation of Digital Service numbers to their transmission media. DS-1 runs over T1 and DS-3 over T3. However, there can be several dozen variations of each DS service, which is why you should know about them. It is not our intention to teach everything there is to be known about Digital Services, but merely to expose you to them. If you want more detail, call your local telephone company's data communications representative. If you are serious about buying a leased line, they'll be more than happy to talk to you.

Below are the most commonly used services available to users in North America:

- DS-0—This is the truly basic rate of 64,000 bits per second used to transmit a single phone call. A number of these bits are used for in-channel signaling, so the actual number of bits available to the user can be quite a bit smaller. For example, one form of DS-0 being used to carry computer data has a user bandwidth of only 56,000 bits per second. There are variants of DS-0 for voice, audio, and computer data transmissions, the details of which would numb your mind. Therefore, we will not go into them. Generally, you cannot buy this as a separate service (except as a dedicated line from some telephone companies), but since it is the basis of everything else, it should be included.

- DS-1—This is what is normally called T1. It consists of 24 DS-0 channels. This service can be set up either as 24 voice lines, or it can be set up to carry 23 channels of computer data rated at 64,000 bits per second. The twenty-fourth channel is used for signaling. As we will see, this particular service has developed into Primary Rate ISDN. It is also possible to glue together two or more of the DS-0 channels to give a single channel of more than 64,000 bits per second. We'll save that discussion for when we talk about Primary Rate ISDN.
- DS-3—Popularly known as T3, this service has a nominal bandwidth of 44.763 megabits per second. It consists of 672 DS-0 channels. If you need to ask how much it costs, you can't afford it. DS-3 can be used to multiplex together a number of other DS trunks. For example, DS-3 can carry 28 DS-1 trunks.

As we indicated, the correct terms for these services are DS-0, DS-1 and DS-3. Having said that, we'll bow to colloquial convention and refer to them as T1 and T3. Because the DS-0 service is the standard phone connection, there is no T0. We'll just call it the local loop or subscriber connection.

Next, let's look at the European version of these services. As noted, CCITT did not have quite the impact on AT&T's thinking during the early 1960s as could have been desired. Thus there are several incompatibilities between E-1 and T1. These range from the way timing and signaling are done, to such exotic details as how long strings of 0's and 1's are handled. Fortunately, almost all of these can be handled by conversion equipment that can interface T1 to E-1.

By and large, the only serious incompatibility is that the European E-1 transmission services have a substantially larger bandwidth. What this means is that if you are planning to build a world-wide corporate network, your best bet is to do your planning using the American standards. This is because you should have no difficulty getting T1's bandwidth piped onto an E-1 because 1.544 megabits is a good deal smaller than 2.048 megabits. However, there is no way you can get the full bandwidth of the E-1 onto a T1.

If all this sounds confusing, it is. And please rest assured that you have seen only the smallest tip of a large and ugly iceberg. In a word, the digital telephone system we are presently using is a mess. There are several primary driving forces behind the advent of B-ISDN. We'll look at them next.

4.2 The Issues.

This alphabet soup of DS-3, CEPT-3, T1, E-1 and so forth actually has a name. It's called the Plesiochronous Digital Hierarchy. We'll concentrate on the "Digital Hierarchy" part of the name first and tackle the meaning of "Plesiochronous" in a moment. As you remember, the European version of telephony standard for B-ISDN is SDH, or Synchronous Digital Hierarchy. Thus “Digital Hierarchy” is a very popular term in telephony jargon, and so we should define it.

“Digital Hierarchy” simply refers to the layers or hierarchy of muliplexing available. That is to say how many levels of multiplexing occur. There can be at least four, although usually there are only three levels in the United States. For example, there are 24 DS-0 circuits in a DS-1, which in turn is multiplexed into either DS-3 (28 DS-1s) or DS-2 (a mere 4 DS-1) for three levels of multiplexing.

4.2.1 *The Grand Plan That Isn’t.*

The ugly part of this is hierarchy was created in an ad hoc manner over a number of years with little thought to neatness. And as noted several times earlier, both the American and European telephone companies tended to ignore what the other did. This “do your own thing” attitude led to a number of interesting problems.

- First, there were a number of compatibility issues regarding international telephone calls. These have already been noted.
- Second, even in the same part of the world, a number of variants of each standard flourished. Telephone equipment from different vendors often wouldn’t interoperate with each other. This generally required that the same manufacturer’s equipment be on both ends of, say, a DS-3 line. This is often called the *mid-span connection* problem.
- Maintenance issues were generally ignored in the PDH specifications. It was assumed that highly trained technicians using expensive test equipment would troubleshoot and repair the equipment. Although this was a reasonable assumption in 1970 when telephone rates were set by governmental groups, it turned into a financial nightmare in the competitive communications environment of the 1990s.
- Higher levels of the Plesiochronous Digital Hierarchy, such as DS-4, DS-5 and DS-6 weren’t defined. Every vendor of telephone equipment went off and invented their own proprietary version of such equipment.

However, the above were almost minor nuisances when compared to the really serious issue which can be found in the mysterious word “Plesiochronous.” Don’t bother to look in the dictionary because you won’t find it. The root “plesio” means “near” and the telephony meaning is generally given as “nearly synchronous.”

While all this might seem like Greek to you (and it is!), there is a very important statement being made. “Synchronous” in telephonic circles has a specific meaning which is that a specific bit or byte of digital information from a specific phone call is always to be found in the same location from one frame to the next. The reason why this is so important is that if you know where the information is, you can tease it out.

If you go back section 3.4.2 and figure 3, you will see an example of this. Generally speaking, DS-1 lines are multiplexed synchronously. That is to say, a time division multiplexer can add and drop a specific phone call out of the muliplexed mass of phone calls. DS-3 and above are not synchronous. They are plesiochronous. And that is not good enough. For a number of reasons that we will look at in a moment, the bits for a particular call link could be anywhere in the DS-3 frame. The end result is if you want to get one single phone call out of a DS-3 multiplexed line, you have to demux the whole mess back down to the individual DS-1 level, and that takes a lot of equipment—equipment that has to be duplicated at each and every switching station a DS-3 trunk passes through. That adds up to a lot of money even for the telephone companies.

4.2.2 *Why Ever Did They Do That?*

We are assuming you asked that question when you read the previous paragraph. Perhaps you didn't, but we're still going to answer that question anyhow because it is a critical point.

The reason DS-3 and above in the Plesiochronous Digital Hierarchy are not synchronous is because electricity (and light for that matter) doesn't travel instantaneously. In fact, the both electricity and light travel at roughly 200,000 kilometers per second through their respective transmission media. However, that's just a rule of thumb. The actual transmission speed varies according to pressure, temperature, type of physical media, the phase of the moon, whether or not you had dinner and several hundred other factors. In a word, it's complex.

Now let's consider what goes into a DS-3 transmission line. It is a multiplexed-multiplexed transmission. That is, it is 28 DS-1 lines which in turn consist of 24 individual phone calls. That's 672 different phone calls. As you may recall, the 28 DS-1 lines are time division multiplexed which means that there is a clock in each multiplexer, each ticking merrily away 8,000 times a second. Have you ever tried to set 28 watches to exactly the same time? And don't bother to even start working on schemes that use complex networks of wires running from muliplexer to multiplexer, because you can't do it. Remember that there is a propagation delay for the electric pulses to transverse the wiring.

In fact, the propagation delay actually makes things more complicated. For example, let's put these 28 DS-1 multiplexers all over the place—perhaps hundreds of miles a part. Now we have to worry about the weather, particularly the temperature as it makes the phone lines become shorter or longer. While a coefficient of expansion of 0.01 percent per degree doesn't sound like much, it adds up to many feet over a 100-mile span. Finally, temperature is one of the many things that affect the transmission speed of light and electricity. So every time the sun came out from behind a cloud, the bits moved faster. This is often called *jitter* and is very descriptive, as is the phrase migraine headache.

The only practical solution the telephone engineers were able to come up with was bit stuffing, inserting additional bits here and there in each frame to make up for these differences in timings. And since the temperature effects could occur rapidly (like at sunset) the number of additional bits varied from frame to frame and where they were in each frame. This made the actual data move back and forth in an unpredictable manner. Thus while the DS-3 and above multiplexed lines were nominally synchronous, the stuffing bits made it plesiochronous.

The bad part is that you no longer knew *exactly* where anything was in the frame. This, in turn, meant that you had to get those @#$#& (as they were known to telephone engineers) bits out in order to get to the next lower level in the digital hierarchy. Even though it was possible to do this, it was only at the expense of completely decomposing the entire frame. This is expensive because to get to a DS-1 trunk, the entire DS-3 frame had to be decomposed at each and every switching station, and all the various DS-1 trunks had to be individually routed inside of the exchange and then recombined into a new DS-3 frame again.

Naturally, the telephone engineers understood all this and what it meant. They spent a number of years thinking about it. And although the concept of digital hierarchy is still with us in B-ISDN, they did come up with a solution.

However, we are again getting ahead of ourselves. If we are to understand some basic concepts of B-ISDN and then ATM, we will have to first study telephony history a bit more first. Next stop, X.25.

Section 5 X.25

Before you wrinkle your nose and skip this section, be forewarned that if you are interested in understanding ISDN and B-ISDN for computer communications, you need to understand something called X.25. Another way of putting it is that X.25 is the granddaddy of B-ISDN as far as computers are concerned. And in a way, it was also a grandparent of ATM.

5.1 Why X.25?

In a few words, X.25 is wide area networking for the masses. You see, one of the subtle points we glossed over in the section on T1 and its European cousin, E-1, was that you have to lease the whole line to get to it. True, in America there are companies that lease a T1 line and then sell some of the channels to other people. Even the telephone companies do it. It's called fractional T1.

Generally, a T1 leased line is an end-to-end transmission service. That is to say, the twisted copper pair sticking out of the computer room's wall in company A's New York office is hard-wired to the similar twisted pair of copper wires sticking out of the wall in the same company's computer room in San Francisco. This is known as a dedicated leased line. No switching in a telephone exchange is need nor even possible.

The nice thing about a dedicated T1 line is that you can transmit your computer data digitally. Special T1 interfaces are available which permit you to blast megabits of data cross country, and you don't even need a modem.

What—no modem?

Well, perhaps you have one, but that depends on what you're doing. Remember that a T1 can be set up in a number a ways. One option is to put a expensive T1 interface on your computer and let it talk digitally end-for-end to another computer many miles away. In this case, you don't need a modem.

Of course, you do need a lot of money to do this.

However, perhaps you're like most of us and don't want to spend several thousands of dollars a month on such a luxury. Or perhaps you don't have enough data to keep a 1.5 Mbps circuit busy for 24 hours a day, seven days a week. What are your options?

Until recently, the only answer was a modem using dial-up lines and going through the telephone exchange. And this is where the last copper mile (or 1.6 km), known as the local or subscriber loop, mucks ups the works. You see, until ISDN, you had to use an analog interface to the telephone system to get into the local telephone exchange. This is why a modem is necessary with today's (e.g. pre-ISDN) telephone system. The modem takes perfectly fine digital information, converts it to analog tones, which are most likely reconverted back to digital to be transmitted over the digital long-distance trunk lines. Then, once the data reaches the other end, it's converted back to analog so it can go through the local loop on the other end.

Sounds inefficient, doesn't it? Well, it is. The word most computer hackers use to describe this sort of mess is "kludge." But remember that the telephone system was built to carry voice messages, not computer data. And originally, things were darn right ugly.

About twenty-five years ago, when acoustic couplers (that's what modems were called back when core memory was still all the rage), you were fortunate to have gotten 300 bits a second through a circuit that is actually rated at 64,000. That was because even the best telephone service back then was pretty poor when compared with today's. The connections were noisy, often filled with squeaks, squeals and hisses that made it difficult to hear even normal voice communications. It was even more difficult for the poor computers: They didn't have the real-time signal-processing capabilities of the human brain. However, the demand was there, and so everybody tried to come up with a workable solution. Fairly soon, there were literally dozens of different techniques and procedures available—roughly one for each computer or minicomputer manufacturer. Naturally, none of them talked to one another.

In time, the problem was dumped into the CCITT's lap. In 1976, it came up with a clever solution. It proposed that packets of data be shipped over the telephone network. While the idea of packets of data was nothing new, the CCITT proposed what was in effect an integrated data network which coexisted on the telephone systems with standard voice communications. However, there was a significant difference. They decided against making a physical connection between the two end-points as you would in a normal voice telephone call. That is to say there was no pair of wires dedicated to the connection. Please note this last point. It is quite important and central to everything else that follows. We'll even repeat it: There is no dedicated physical phone connection between the two end points.

Now for the ringer—there is a phone connection.

How can this be? Well, the answer is the same as how can a program not be in memory, but yet be in memory. The answer is that the program is in virtual memory. And in the case of the X.25 connection, it is through a virtual circuit. (Please note the last point: Virtual circuit is a vital concept in modern telephony-based networking. We will see it time and again in this paper.)

A virtual circuit has all the characteristics of a standard telephone circuit, except that it doesn't use any physical resources unless it needs them. Thus, when there isn't any actual communications activity between the two computer systems connected by an X.25 session, almost all the telephone equipment involved is being used for other work. This is analogous to the concept of swapping in virtual memory computers. When the program pauses for some reason, it is moved out of memory and another program is given the physical memory to use. In the case of virtual circuits, it's the telephone lines that are shared.

5.2 How X.25 Works.

While most of the Europeans reading this white paper are well aware of X.25, most of the Americans aren't. For a number of reasons, all having to do with cost of telephone bills, X.25 did not become popular in the United States. In contrast, the European PTTs set the pricing of X.25 so that it was an attractive way to do wide area networking. However, as we have already indicated, X.25 is the progenitor of most modern telephony-based networks, including: ISDN, Frame Relay, SMDS, and ATM. Therefore, we will take the time to explore it with the intent of explaining several other concepts which have found their way into these later protocols.

First, although X.25 uses the same long-distance telephone circuits that a voice telephone call might use, it uses them in quite a different manner, as already noted in the above section.

A normal voice telephone call uses a switched circuit. That is to say a pair of wires is "physically" connected between the two telephones is dedicated to that call when it is made and kept that way for as long as it the two phones are connected. The reason why we put quote marks around the word "physically" is the technical nit that long-distance phone calls are multiplexed. But even then, bandwidth on the multiplexed line is preallocated to that call. On the other hand, if this is a local call, one can actually trace this physical connection.

Thus, when you dial a number, telephone switching equipment finds an appropriate path between your telephone and that of whomever you are calling and allocates the path to that call. If telephone equipment can't find an appropriate path, you get a busy signal. If it can make the connection, the switching equipment locks the circuit so that nobody else can use it (i.e. other people now get a busy signal if they try to call either of you).

This, however, is a terrible way to connect two computers or even a computer user to the computer. The reason is that computer-oriented communication tends to be *bursty*. That is, it occurs in bursts of activity with long periods of no activity in between.

For this reason, X.25 switches packets of data instead in what is called a Packet Switched Data Network (PSDN). A long message, such as this white paper, is divided up into a number of packets of between 128 bytes and 4096 bytes each and treated as individual messages, each to be transmitted and managed by the network as separate entities. In its day, this concept was revolutionary. And in a conservative and staid group such as the CCITT, it was considered radical. However, the reasons for doing so were compelling.

- First, as already noted, the reason why the CCITT proposed a PSDN is because computer communications occur in bursts of activity.
- Human-to-human phone calls also tend to be short, averaging a few minutes each (with the notable exceptions of teenage daughters and lonely lovers). Computer connections usually last for long periods of time, usually lasting hours.

When you look at the two types of communications, it is reasonable to tie up valuable telephone equipment for the sole use of two humans because they are using it in a fairly continuous manner for few minutes. On the other hand, doing the same thing for computer communications is a patented waste because they are connected for long periods of time and usually inactive for most of that time. Thus, the CCITT decided to use packet switching for computer communications because it was appropriate for the bursty nature of such communications. It chose to use virtual circuits to permit more than one pair of computers to use the same telephone equipment more or less at the same time.

5.3 *The X.25 Network Architecture.*

Now that we have all the basics out of the way, let's look at the architecture of this network as shown in figure 4. On the left is a user hooked up through something called a PAD, which means Packet Assembly and Disassembly. On the right side is the host computer—also hooked up to a PAD. The PAD itself may or may not be present in a X.25 network, but the two ends are always connected as the Data Terminal Equipment (DTE) to the network cloud shown the middle of the page. This "cloud" is known as the Data Communications Equipment (DCE). Normally, the DCE is shown exactly as that—a cloud—because for most presentations about X.25, the discussion is focused mainly on the various messages that are sent into or received from this nebulous mass. Inevitably, little concern is given to what actually happens inside the cloud.

An excellent analogy to how X.25 works from a user viewpoint is to look at it as though it were just a telephone call. We, as users, are merely concerned about knowing how to dial a call, picking up a telephone when it rings and hanging it up when we are done. Most of us care little about how the telephone system works and so it might as well be a cloud. However, for this discussion, we are more interested in how the telephone system works and therefore interested in what does happen inside the "cloud." So we have turned on an X-ray machine and discovered that inside the cloud there are a number of network nodes (NN) linked together by one or more network links (NL). In fact, each NN is a small computer of some kind located at a telephone exchange and the NLs are nothing more than dedicated telephone lines. They are usually T1 lines in the United States and E-1 lines in Europe. Nothing mysterious here at all.

Now for a more detailed explanation of the X.25 network.

On the left side of figure 4, we have a user who is using what is typically called a dumb terminal. Nowadays, few such terminals are still in use, but since they were the primary user interface to X.25 when it was first designed, we are showing it to explain the functionality. As most of you may remember from the dark days of computing before the invention of icons, GUI, and 24-bit color graphics, these terminals communicated one character at a time through something called an RS-232 interface. However, as already noted, X.25 is a packet-oriented network. Thus there is a needed for some sort of interface to collect these single characters and combine them into a packet. This is the function of the PAD, which stands for Packet Assembly Disassembly. These were usually a microprocessor-based device, often a Z80 with 64 kilobytes of RAM and a couple SIOs.

The PAD collects the characters being typed by the user into a buffer and when something like 128 characters are available, the PAD packages the buffer of characters in something called a LAPB frame and ships it off into the X.25 DCE cloud. Eventually, it arrives at a destination PAD, where the process is reversed and the host system sees the 128 characters dribble in through its serial port more or less as though the remote user had been directly attached to that serial port.

Over time, the dumb terminals gave way to modern workstations, and the hardware PAD became a program running inside the workstation. So it is possible to have a X.25 network with no PAD, but there is still a DTE. This, of course, is the workstation's serial port connected to the DCE. We will use the acronyms PAD and DTE more or less interchangeably in the following discussion. And what we really mean by PAD or DTE is the computer the user is using as well as the computer he is talking to. The DCE is simply the stuff in the cloud, the telephone wires and network nodes.

The actual interface from the DTE to the DCE is usually a modem. The original interface was supposed to be a bit-serial synchronous digital interface called out by the CCITT X.21 specification. This, as we will see, is a forerunner of ISDN. The reality is, however, that most X.25 users use a X.21*bis* interface which specifies either a V.28 or V.35 type modem. This is an analog interface. Remember that most of us still have only an analog connection to the telephone system. However, the *intent* was to have a digital connection. That was to come a few years later with the advent of ISDN, which we shall get to shortly. However, we still have a few more points of interest in X.25.

5.4 Making an X.25 "Phone Call."

Now that we understand the basic functioning of the X.25 network, let's look at some of the details. For example, how does one place a virtual phone call?

The answer is that the user's DTE sends a message to whatever host DTE he wishes to be connected to. It works very much like a regular phone call. The difference between a X.25 phone call and a normal one is that X.25 sends a packet of information into the DCE; while in the case of the phone call, you simply tap out the number on the touch-tone pad. There is a phone number in both cases, but it the case of X.25, it is embedded in the information packet. Since the way you make a phone call under ISDN is almost exactly the same as you do under X.25, it is worthwhile to examine what happens when placing of an X.25 phone call.

Normally, what the user of a X.25 network does when he sits down at a terminal is to first identify himself to the PAD or DTE. This is a usually a login procedure. Then the user can request a connection to the host. While the user might enter the name of a system, what is sent is something that looks like a telephone number. In fact, it is a telephone number. It is based on the CCITT X.121 standard. The number is 12 digits long with an optional two digit suffix. This number is similar to the nation and city code familiar to anyone who regularly calls Europe. The optional two digit suffix is used by the receiving DTE. It is typically used to select a particular printer or perhaps terminal attached to the host computer. This suffix is also available under ISDN and permits you to do interesting things like calling a particular telephone in your house.

What happens in the DCE "cloud" is that the Network Node (NN-2) attached directly to the sending PAD receives the packet and responds in two ways. First, it acknowledges the packet to PAD-1 and establishes a Logical Channel Number

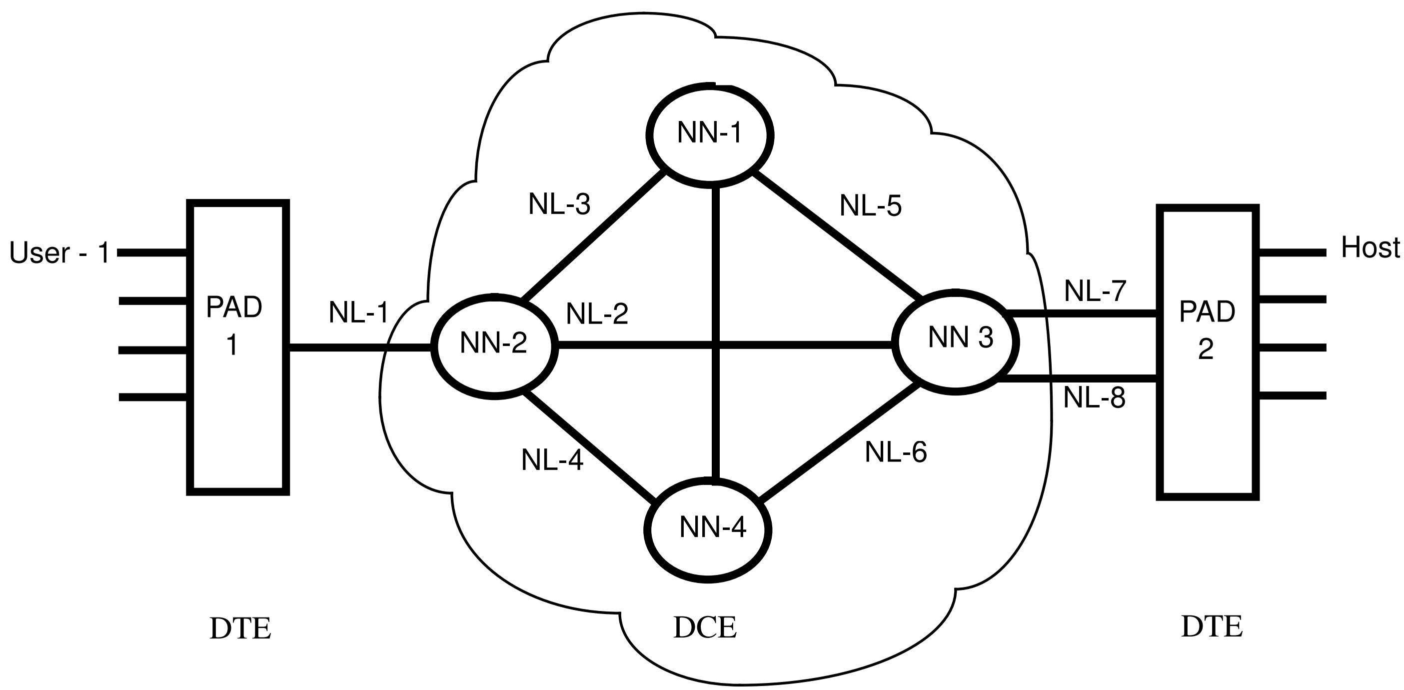

Figure 4 Basic Architecture of a X.25 connection between two computer systems.

(LCN) between itself and PAD-1. By the way, this LCN is strictly local between the sending PAD-1 and NN-2. It should be considered to be nothing more than a shorthand way for the two to communicate about this particular "phone call."

The second thing NN-2 does is try to find a way through the cloud to the destination PAD. It is a routing problem. It might have to hunt and peck through the various combinations of NNs and Network Links (NLs) until it establishes a path from one PAD to the other. For simplicity, let's assume that the route is direct: from PAD-1 through LN-1 to NN-2, through LN-2 to NL-3, through LN-7 to PAD-2.

It is of interest that each of the three legs of this connection will undoubtedly have different Logical Channel Numbers. Remember that the LCNs are simply shorthand tokens to identify a virtual circuit from each computer to the next. A second point to remember is that this hunt-and-peck search for a route from PAD-1 to PAD-2 might give a different route from one call to the next. That is, it is switched from one NN to the next. For this reason it is called a Switched Virtual Circuit (SVC). Please remember this acronym, it is founded in all modern telephony-based networking.

(For those who wish to avoid the chance of a busy signal if all the network links are filled to capacity and have the money to do something about it, they can still lease a T1 (E-1 in Europe) line and have a dedicated circuit. Since the latter are not broken down after each call, these are called Permanent Virtual Circuits or PVCs).

Eventually, the original "phone call" request (known as a call request) reaches the PAD-2. It not only sees its "phone number" in the packet, but it also sees the sender PAD's "phone number" as well. This feature is available on ISDN telephones as well. The receiving PAD could, if it wishes, refuse the connection request. We'll assume, however, that the caller has paid his bills, and so the PAD-2 acknowledges the request and the phone call is established.

The next phase is that the two PADs communicate through the LCN they have established with their respective NNs. And to confuse this even more, NN-2 and NN-3 might be using yet a third LCN, all for the same "phone call." For example, PAD-1 and NN-2 might think of this call as LCN 120, NN-2 and NN-3 might think of the call as LCN 100, and NN-3 and PAD-2 might refer to the call as LCN-88. The reason why this can happen is that there are no physical connections, and each step through the "network cloud" is separately established.

This is another point we need to clarify at this point, and that is about how the packets are sent through the network. Whenever a PAD or an NN receives a packet, it stores it locally until it has time to send the packet onto the next stage of its journey. This is called a *store and forward* system. Once an NN receives a packet, it owns it until it can successfully send the packet on to the next NN or to the receiving PAD. This means that not only must it send the packet, but it must also receive confirmation from the next node that it has successfully received it. Only the actual destruction of one or more of the NNs along the way will cause data to be lost. This makes for a extremely reliable communications system. Information, once in the X.25 network, will almost always come out—eventually.

This also explains why they call this a virtual circuit. Except for the brief time that a packet is being sent over the wire, it is either in the memory or disk drive of one of the NNs. The physical telephone line between the two X.25 nodes can be therefore used to handle a large number of virtual circuits in this manner.

Moving on to the X.25 protocol itself, is a complex, with something like 20 different types of commands, so we won't bore you with the details. If you are interested, there are a number of books on X.25. The types of messages range from call request, to call accepted or refused. There are also messages for ending the connection, either gracefully or forcefully. Other messages permits the DCE to refuse or terminate a connection because of networking problems. Collectively, this protocol is called LAPB. Since there are many books that go into the LAPB protocol in detail, we will only take a passing glance at it. The reason is that it is the basis of the LAP-D protocol used by ISDN.

5.5 The LAPB Protocol.

LAPB, which means Link Access Procedures Balanced, is a packet-oriented protocol with the a general layout shown in figure 5. The packet is delimited by a a flag, one of which is on either end of each LAPB packet. It consists of a fixed pattern of eight bits set to 01111110. This octet, as it is called, can only occur in this one application because special hardware inserts a "0" bit whenever it spots five "1" bits in a row in any other part of the packet. The procedures for doing this are called *bit-stuffing* and known to some of you as HDLC. We will not go into HDLC any deeper than this for our interest in LAPB is generic—we are merely looking at it to understand generally how it works. A generic version of the LAPB packet is shown in figure 5.

Figure 5 The contents of the LAPB packet. The bit position in each byte is shown at the top of the figure.

Bit	8	7	6	5	4	3	2	1
Byte 1	GFI				LCGN			
Byte 2	LCN							
Byte 3	PTI							
Byte 4	[Rest of packet varies according to packet type]							
Byte n								

Only the first three bytes (or octets, if you are a purist) are common through all types of X.25 messages. The byte is divided into the Logical Control Group Number (LCGN) and the General Format Identifier (GFI). The second byte is the Logical Channel Number, while the third byte is the Packet Type Identifier. Their functions include the following:

- LCGN — Originally, this was supposed to help route calls by bundling them to the appropriate trunk line. When combined with the Logical Channel Number, it permits you to have as many as 2^{12} or 4,096 virtual circuits. In reality, it is not used (see LCN). However, as we will see, this concept reappears in ATM.

- GFI — The General Format Identifier has a “Q” bit used to indicate whether the packet contains *higher level* control information or not. The “D” bit tells how the confirmation of delivery is to be made (either PAD to PAD, also called end-to-end, or merely to the PAD and the NN it talks to. The last two bits determine how the packets are to be numbered—either in modulo 8 or modulo 128.
- LCN — The Logical Channel Number is an arbitrary number between 1 and 4,095 (if the LCGN is used with it) to indicate between the various network entities in the X.25 which *logical channel* this packet belongs to. It generally only uses the 8-bits of the LCN and so is usually only in the range of 1 to 255. It is similar in many ways to the file descriptor number used in C or Logical Unit Number used by Fortran to indicate various disk files. The important thing to remember is that the actual value is local between any two nodes in the X.25 network. It has no end-to-end significance. The actual mapping of the input LCN and output LCN on a particular network node is known only in that node. Thus, in our example above, the LCN from PAD-1 and NN-2 is LCN 120, and the LCN for the same “call” from NN-2 to NN-3 is LCN 100. Only NN-2 knows this mapping information.

The remaining bytes of the packet depend on the nature of that packet type. For example, the call request packet contains the telephone numbers that identify which DTE or PAD you want to call. These look like regular telephone numbers and are at least 12 digits long so you don’t want to repeat them all of the time. This is another reason why LCNs are used—they are only one byte long.

5.6 The Strengths and Weaknesses of X.25

Although our overview of X.25 was somewhat casual, there were a number of important issues that need to be highlighted. The good points were:

- First and foremost, X.25 uses digital signaling to establish a connection between two users.
- It uses packets of information to pass control information as well as user data.
- X.25 can be digital end-to-end (if you use X.21 instead of X.21*bis*).
- It uses virtual circuits so that valuable resources (i.e. the long-distance phone wires) can be used more effectively.

Some of X.25 bad points include:

- X.25 can be very slow. The latency for a packet to arrive at the other end of the connect might be minutes if there are networking problem.
- Thus X.25 is a terrible way to talk over the telephone.

5.7 Important Concepts in X.25

This is a mini-review of the last few sections on X.25, but we'd like to point out again a number of the more important concepts presented that you should feel comfortable with. If you don't feel comfortable with them, you might want to reread the appropriate paragraphs for these concepts will reappear and be quite central to our sections on B-ISDN and ATM. These concepts are:

- VC — Virtual Circuits.
- PVC — Permanent Virtual Circuits.
- SVC — Switched Virtual Circuits.
- Packet Switching.
- Digital Signaling.

Section 6 ISDN—The Grand Plan

Now that we have taken a five-minute tour of X.25, we can turn to the issue at hand, which is building an all-digital telephone system from end-to-end, as they say in the telephone industry. This means from the telephone in your hand to the telephone in your mother's (or whomever you are calling) hand. It also assumes, of course, that your voices are converted to and from digital signals within the telephones themselves.

Naturally enough, you might wonder why the telephone industry would be willing to spend literally billions of dollars to do it? Originally, back in 1980, when ISDN was proposed, it solved a large number of issues for the telephone companies. These are basically the same ones we covered before in section 3 and, as you remember, basically boiled down to costs. In addition, there were all sorts of new services such as videotext, digital fax and even audio services such as piped music; all of which would enable the telephone systems to gain additional revenues.

And, as you will remember, the fly in the ointment was the last copper mile, or the subscriber loop, which is the twisted pair of 22- to 26-gauge copper wires going from your telephone to the local telephone exchange. These were originally designed to handle a rather narrow band of frequencies. That is, they were designed to handle analog voltages in the 20 to 50-volt range with frequencies of less that 4,000 hertz. This sort of wiring cannot be easily convert to carrying even voice-grade digital communications at 64,000 bits per second.

Since about 80 percent of the capital investment of the telephone systems of the world is tied up in these wires, it became important to find a way around this problem. The goal was to keep the wires, but change the electronics, if necessary, to get the necessary bandwidth to bring the home telephone into the digital age.

And they succeeded. In 1988, ANSI standard T1.601-1988 was approved which defined something known as a Basic Rate local loop in which the preexisting subscriber loop wire (that is, one pair of twisted copper wires) could be used to drive 160 kilobits per second over a distance of about 18,000 feet. Just how they perform this magic is not important to us, but the fact that they could do it is important for it is the basic enabling technology of ISDN. It meant that it was now possible to connect our homes and offices to the wonderful world of ISDN.

There were, of course, a number of other problems to overcome. One was selling it to the user. As noted earlier, all the telephones would have to be replaced and that cost was obviously going to come directly out of the pocket of the user. Another question is how do you sell this when the going-in cost is probably several hundred dollars in new telephones per household?

Well, you can start hyping all the neat new things you can do, such a digital fax, and perhaps have the ability to call your microwave and have it start dinner at 8 p.m. instead of 7:30. You could also come up with all sorts of exciting new features such as being able to see the telephone number of who is calling you (a feature now disabled in the United States unless the caller wants it). Generally, in the United States, most common folks really had no use for all these new capabilities and so ISDN did not get pushed by the local telephone operating companies. They had no interest in selling ISDN.

How can this be? After all, it was these same telephone companies that started the whole thing, wasn't it? Well, not exactly. Remember that, in 1983, AT&T divested itself of the 22 local telephone companies it owned, and these are now operating as completely independent entities. Thus the economic pressures of the late 1970s and early 1980s which gave rise to ISDN were gone, in the United States at least. In fact, the new realities now made it much less desirable for the local telephone operating companies for they now would have to foot the conversion bill all by themselves. It had also become obvious by 1986 that B-ISDN would blow ISDN away in a few years, anyhow. So the local telephone companies did nothing if at all possible.

Therefore, ISDN became popular only where there was a unified end-to-end telephone system much like AT&T had once been. Because these are by and large the European PTTs, it is not surprising that ISDN had become popular in Europe, most specifically France and Germany. These two countries were rebuilding their telephone systems in the 1980s anyhow, making ISDN a logical thing for them. On the other hand, ISDN has not been a overwhelming success in the United States because the cost for conversion was far greater than the payback to the average telephone user. This is not to say ISDN is not available in the United States, for it is, but it is generally confined to large companies which own their own telephone exchanges (PBXs) and a number of computer users who work from home. ISDN makes the 9,600 baud analog modem obsolete and gives the computer user at least two 64,000 bit per second digital transmission channels.

However, even though ISDN may appear to be a nonstarter in the grand technology race, it is nevertheless an important evolutionary step. Thus we should look at it much as a paleontologist studies fossils to help understand how life evolved.

6.1 *ISDN, the Technology.*

ISDN is basically a digital user interface to the digital telephone system built during the late 1970s and 1980s by the various long-distance telephone companies of the world. This first digital system is generally referred to as the PDH or Plesiochronous Digital Hierarchy. And it is still the dominant digital telephone system of the world. As noted earlier, PDH is presently being replaced by something called Synchronous Digital Hierarchy or SDH. SDH is also known in the United States as SONET and is the basis of the Information Superhighway.

Because it is meant to be a user interface to the digital phone system, ISDN focuses a large part of its specification on this aspect of the network. And since this aspect of ISDN is of most interest to us, it is what we will look at.

Like many other networking protocols, ISDN covers the bottom three of the standard ISO levels we all know and love. These are the Physical, Data Link and Network Layers. We'll start from the Physical Layer and work up.

6.1.1 ISDN, Physical Layer.

As expected, ISDN's physical layer is oriented toward telephone wiring. There are two basic versions of this physical layer users can have access to. These are the Basic and Primary Rates. These are also known as the local subscriber loop and the T1 multiplexed long-distance telephone line.

6.1.1.1 Basic Rate.

This service is designed to run through the preexistent twisted pair of wires running from your house to the local telephone exchange. The interesting thing is that there are three logical circuits. Two, each rated at 64,000 bits per second, are known as B channels. These are the so-called *bearer* channels for they carry the actual digitized data. Although a number of people talk about the user-bandwidth of the Basic Rate as being 128 Kbits, this is misleading. There are two separate 64,000 bit channels, and although there are those who have combined them using ad hoc technology, there is no official specification for doing so. If you think about it even briefly, the design of Basic Rate gives you two channels of voice communications that are almost exactly what is carried today over the long-distance telephone system. Naturally, this is also used for other communications such as computer data links as well as videotext, fax and even audio (i.e. music).

The third channel in Basic Rate is the D channel. Just what "D" stands for, if anything, is not generally defined. Most speculation is that is means "Data" channel. What it is used for, however, is quite clear. It is an out-of-channel signaling and control channel. That is to say, the memory of the Phone Phreaks is still with the telephone companies of the world and when it came the design of ISDN, they made certain to keep all signaling functions out of the actual customer (i.e. bearer) circuits. In the case of Basic Rate D channel, the bandwidth is 16,000 bits per second. And although nominally available for use to carry such information as burglar and fire alarms, the channel is in practice used exclusively by the telephone companies for telephony signaling.

The interesting point about the physical layer of Basic Rate is that it is defined as a four-wire system. That is, four wires are needed to carry the signals—at least in your home or office. This means that your present telephone wiring ether needs to be converted from the present two sets of two-wires, or additional wiring needs to be installed. However, this is true only in your house or office for Basic Rate still uses the same old two-wire circuit the for the subscriber loop itself.

As noted above, the ANSI standard T1.601-1988 defined the technology of the Basic Rate local loop in which the preexistent subscriber loop wire (that is, one pair of twisted copper wire) could be used to drive 160 kilobits per second over a distance of about 18,000 feet. Thus, the two B channels and the one D channel easily fit inside this bandwidth. However, this is for only a distance of a little more than three miles. Fortunately, this covers almost everybody who lives in cities or near a town or large village. Beyond that, you need repeaters, which are expensive. Thus, ISDN is not something for the rural folk.

6.1.1.2 Primary Rate.

Primary Rate is designed explicitly for the American T1 and European E-1 long-distance phone circuits. In the case of T1, Primary Rate support twenty-three 64,000 bit B channels and one 64,000 bit D channel for at total of twenty-four channels. This, of course, is exactly what T1 is. The E-1 version uses thirty 64,000 bit B channels and one 64,000 bit D, plus the thirty-second channel which is still used for timing.

Unlike Basic Rate, you can officially combine several B channels to form larger capacity bearer channels in Primary Rate. These are the H channels. They come in three sizes: H0 at 384,000 bits, H11 at 1,536,000 bits and H12 at 1,920,000 bits per second. These translate into 6, 24 and 30 standard B channels respectively, and they are designed to work either on the American T1 (H0 and H11) or the European E-1 (H0 and H12) cables. Other combinations of H channels are possible but as yet none others have been defined. (And given the emergence of ATM/SDH, it is not likely that they ever will.)

6.1.2 ISDN, Data Link Layer.

Although the B channel is effectively the same thing as the switch virtual circuit we first saw in X.25, none of the telephony signaling is done through it. That is the sole domain of the D channel. As far as we (and the telephone companies) are concerned, the user can do whatever he wants on the B-channel—as long as it's digital information and meets the electrical specifications of the B channel. In fact, this is a bit of an overstatement because the B channels also have a protocol akin that on the D channel. In fact, there are at least three:

- First, one can run X.25 over the B channel (and in Europe the D channel as well), so you can actually send switched packets over B channels—but they all have to basically go the to same phone number (but remember that two digit suffix we talked about in X.25. You can use it to select different devices or logical terminations at the same "phone number").
- In addition, there are two digital frame formats. Remember that the B channel is digital and so frames of information are transmitted through it. In fact, the data of the D and B channels are Time Division Multiplexed (at 8 kilohertz) on that pair of wires we discussed a few paragraphs ago. Thus the simple Basic Rate ISDN telephone not only is now doing analog to/from digital conversion but it is also doing TDM as well. And because it is possible to have two phone calls coming to the same telephone (there are two B channels), it must also be able to select between the two. This, of course, is done via the two suffix digits on the telephone number we talked about earlier. And, in fact, each ISDN telephone or device (i.e. computer) in your home will have a unique suffix. As for non-packet digital communications, there are two basic versions. In Europe, the protocol generally used on the B channel is V.110, which is a fairly simple synchronous transmission technique. For example,

there is no Cyclic Redundancy Check or CRC unless the user embeds it inside of his data. This is best suited for voice communications but can handle computer and video text data.

On the other hand, the United States uses the much more sophisticated V.120 protocol. This, like LAPB, uses HDLC or bit stuffing to avoid excessively long strings of “1” or “0” bits and allows up to eight channels of multiplexing. In addition, it even has a CRC. While this might seem excessive, it should be noted that ISDN is a telephone system in Europe and is used more for computer communications in the United States so all the extra functionality of V.120 is needed and desirable. All this means that your ISDN telephone will have a lot of electronics in it. Imagine the cost of a telephone with all that stuff built into it. It certainly isn’t the simple $9.95 analog telephone you can pick up at your corner drugstore.

Getting back to ISDN, the point is that what happens on the B channels is fairly much up to the user, as it should be. He can use it to send X.25 packets, HDLC data streams or even talk over it. That is to say, not only is it a switched virtual circuit, but you can also switch packets over it as long as they are all going to the same basic telephone number (remember those suffix digits!) And in the future other functions are planned for it such as videotext and slow-scan video conferencing.

In contrast, what goes on in the D channel is of vital interest to the telephone company. It is how the telephony signaling is done. And guess what they use? It’s something called LAP-D which is similar to our old friend LAPB from X.25.

That’s right, the D channel uses *packets* to send and receive switching information. The ISDN telephone system uses computer network-like packets to establish phone call connections, send billing information, plus all sorts of other good stuff such as who is calling, charge reversal requests and so forth.

Our outline for this paper originally has us going off and exploring LAP-D at this point, but we won’t bother. It is enough like LAPB that if you want to understand how it works, read section 5.5 if you skipped it. What we are more interested in is not what the similarities between ISDN and X.25 are, but the differences for they show the evolution of the modern B-ISDN telephone system most clearly. However, before we do that, we need to look at the third layer of ISDN, the network layer.

6.1.3 *ISDN, The Network Layer.*

While LAP-D is a fairly obvious packet-oriented protocol, it is merely that, a protocol which defines a number of different packet formats which permit various kinds of data to be shipped back and forth between ISDN telephones and ISDN switches as well as the terminal ends of the circuit. Thus, to place a phone call, your ISDN telephone now sends network-like packets out the D channel to the computer-like telephone switch which figures out how to establish a connection to the other ISDN telephone owned by whoever it is you want to talk to. That telephone receives the set-up packet and can decide that it doesn’t want to talk to your telephone (i.e. call blocking). This higher level of logic that is new and unique to the telephone system although the rudiments of it can be found in X.25.

We need to spend some time on it for as we shall see, it is a major step toward ATM/SDH. It's called Q.931. Many people also refer to it as Digital Subscriber Signaling System Number 1. For some reason it is abbreviated as DSS1 and not DSSS1 as you would expect.

Please note, however, that DSS1 refers not only to Q.931, but also to the specifications for additional higher-level functionality found in another specification called Q.932 as well as a large number of other additional goodies now found in the Q.95X series of CCITT standards. Thus, from a pedantic viewpoint, DSS1 is not a layer but the entire upper half of the networking protocol stack. Therefore, having noted the existence of DSS1, we shall ignore it and try to retain some form of lip service to the ISO networking model by focusing in on Q.931. It is the specification that defines the *basic* functionality, which is all we need for the purposes of this white paper. All we really want to know is how to make a phone call on ISDN.

6.1.4 *Making a Phone Call on ISDN, Q.931 and SS7.*

By now you have a good idea of what happens when you pick up your analog telephone and dial a number. In the section on X.25, we also explained how you can make a virtual telephone call over the X.25 network by sending packets back and forth to your intended recipient. The procedure for ISDN is similar to that used by X.25, but two protocols are used. One is Q.931, which is used between you and the telephone company and sometimes to your recipient. The other protocol is whatever the telephone company is using to set up your virtual circuit. The plan is that telephony network control to be done by Signaling System 7. Whether or not that comes about is open to debate, but since there is *something* out there doing that function, we'll pretend that it's SS7. Our only caution is don't assume that it is SS7 for it could be something else. Another point is that SS7 is different from DSS1, which is used strictly between the end-points (i.e. ISDN telephones) and the local telephone exchanges. While it might seem inefficient to use two separate protocols, there is a good reason for doing it. The telephone companies want to keep *their* switching logic as far away from the Phone Phreaks as possible.

With that overview, let's get on with the way an ISDN telephone works. Basically, it lies to you a lot. You pick it up and you hear a dial tone. Then you happily tap in the telephone number you want, and you hear a reassuring beep each time you tap a button. Then, depending to how your telephone was programmed by the local telephone switch, it sends the first packet of information to that switch. Until this time, the communications have been strictly between you and your telephone; all the noises it made so far were it merely pretending that it is a good old fashion analog touch-tone telephone so that you would feel comfortable with it. If you listen carefully, you'd notice that all the tones for the various keys you tapped are exactly the same while there is quite a range of tones emitted by a real touch tone.

That is because the ISDN telephone is actually waiting for you to tap enough keys for it to figure out what it is you are trying to do. This will, naturally enough, depend on where you are. For example, at the office you might tap only four or five keys to get another employee on the PBX, while at home, you might have to tap seven or eight. Or if you start off with “01”, the telephone might expect a much longer string for an international direct dialing call, while if you tap “0” and wait, it will eventually figure out that you want to talk to an operator (remember those?). Only when it is convinced that you have tapped the proper number of digits will the telephone or other ISDN terminal end device (such as your workstation) try to set up the telephone call. This is called *en-bloc* dialing. A single packet is sent to the local telephone switch with all of the digits of the telephone number you want in it. This is particularly useful if you are using a particularly smart ISDN terminal-end device such as a workstation.

Naturally enough, there are those who didn’t like this technique for there are a large number of special cases such as 911 (emergency hotline in the United States) which have to be individually keep track of, and that is asking a lot of your telephone. Therefore, mainly for telephone use, the Q.931 specification also permits *overlap* sending of digits in which a string of packets, one for each digit is sent by your telephone to the local telephone switch. This permits the telephone to be somewhat simpler (and cheaper) and places all that logic for sorting 911 out from area code 901 in the switch.

In either case, the local telephone switch eventually knows what the number you are dialing is and so goes about the business of finding a route. This is exactly like the procedure used by the human operators seventy years ago when you picked up the phone and made a long-distance telephone call. The difference is that your local telephone switch either passes this function off to another bit of telephone equipment or sets up the call itself by using SS7 to communicate to a series of telephone switches along the way until a virtual circuit for your call is found.

Please note that we used the word “virtual,” meaning that the entire route from your handset (or terminal end device) to the other telephone is over multiplexed circuits. This is true even of your local loop because it is now also multiplexed with two B channels and one D channel. No longer are specific copper wires set aside for the sole use of a single communication. What has not changed, however, it that the bandwidth needed for your telephone call all along the virtual circuit has been preallocated to your use. Thus, in the case of ISDN telephone call, you have a guaranteed 64,000 bits per second.

Because SS7 is transparent to us, we are not going to go into it beyond saying that it is the proposed network for the all the telephone systems in the world to interconnect and make international direct dialing much neater and cleaner for the telephone companies. All it does, from our viewpoint, is set up the virtual circuit from our telephone to the one we are calling.

At this point, Q.931 comes back into play. The first switch, the one our telephone is connected to, sends a message to that other telephone. This is a Q.931 set up message, which contains the number you are calling, your telephone number and other information, most of which is not important to us. It is of interest that the recipient knows who is calling and can choose to reject a call even before it rings the bell. Another feature is that the telephone of the receiving end can display that number on a LCD screen so that the called person has some idea who is calling (however, a court decision in the United States permits the caller to remain anonymous if he wishes).

Assuming that the recipient's telephone was instructed to let your calls through, it starts generating a ringing sound and sends back an *alert* packet to your telephone, which now also generates a ringing sound so that you know that the other telephone is ringing. Once upon a time, you heard the ringing from the recipient's local switch, but now it's all faked, just to make you feel like something is happening.

Eventually, the person you called picks up the phone and says "hello." And so you talk by sending digitized information back and forth in packets. This goes on until one or the other of you hangs up your telephone. This action was once use to break a electrical circuit but no more. What happens instead is that the telephone being hung up first sends a special Q.931 *disconnect* packet. This is seen by both local switches, and they in turn terminate the ISDN connection. Then whichever local switch initiated the phone call sends out a message to the telephone network that the virtual circuit you were using is now free and the bandwidth allocated to your phone call is now free. This, naturally, uses SS7.

Before going on to the summary about ISDN, there are a few points we want to stress that SS7 is in no way tied to ISDN, B-ISDN or any other single telephony protocol. It is totally separate and something none of us is ever likely to have access to. It is the control network for the new generation telephone system. It is meant to be used only by telephone companies and PTTs for their needs. The reason why we even mentioned it to show you how much networking concepts and technology are being integrated into the design of the world's telephone system. This trend first started with X.25 and has been accelerating ever since. In fact, many of the concepts we shall see in ATM and SDH/SONET came from the networking world.

6.2 ISDN, a Critical Summary.

As indicated earlier, we believe that ISDN has had its day in the spotlight. That spotlight has obviously since shifted to ATM and SDH/SONET. However, there are many features of ISDN which are important to us.

- First, it is an all-digital telephone system, end-to-end. This is critical to us as computer users as it makes computer communications much faster and easier.
- It got around the last copper mile problem without replacing several billion dollars of wiring. Unfortunately, the limited bandwidth it could attain was a serious shortcoming. Thus the last copper mile problem is still with us and a big problem for ATM and SDH/SONET. It is, however, an opportunity to make a billion or so if you can come up with the answer.
- It is the first example of packet message control not only of a WAN like X.25 running over the telephone system, but also of the telephone system itself. Indeed it uses a protocol (Q.931) which is based on the control logic of X.25.

- However, unlike X.25, there is no store and forwarding. There is also no guarantee that a packet pushed into the network will get through. This is a logical outcome of the fact that ISDN is designed to handle voice as well as computer data. Thus, real-time response is a requirement for voice communications.
- The flow control of X.25 has is simply not there. The Media Access Control, as networking people like to call it (in ethernet it is collisions and in FDDI it is the token), is nothing more than pre-allocating sufficient bandwidth. ISDN is really still just a phone call. It is designed to be set up for some number of minutes, give whoever wants to use it a fixed bandwidth and that's that. Although there are a number of tricks and schemes for mapping, say, six 9,600 bits per second data streams into a B channel, there is no good way of handling bursty communications typical of the computer networking world beyond buying enough bandwidth to handle the worst case, and that's expensive.
- ISDN is switched. Unlike LANs, which by and large share a physical media such as the ethernet or FDDI fiber in a party line fashion, ISDN is a switched network. That is, connections are established between two end-terminals (or telephones) and the complete bandwidth of the media is available to just those two entities. (Well, we agree that this is a fairly obvious point; one of the hallmarks of LANs is that they use shared media while WANs are switched. The reason why we make this point will be clear when we get to ATM.)
- Last, but not least, ISDN is a fairly narrow bandwidth capability. Officially, there is no 128,000 bit per second channels on the Basic Rate although a number of ad hoc schemes are already in place. While even 64,000 bits per second would have been a god-sent ten or perhaps only five years ago, it is no longer sufficient for the modern graphical user interfaces and gigabit files.

In summary, we consider ISDN a very important stepping stone. It is, however, the victim of bureaucratic lethargy. ISDN was first proposed in the late 1970s, and the CCITT spent the better part of ten years working on the specifications. Had ISDN been generally approved in the early 1980s as originally planned, it might have taken off. However, it wasn't until 1988 that all the important parts of it were approved standards. By that time, ATM and SDH/SONET and the pure sex of fiber optic transmission had become the next obvious step in the evolution of the world's telephone system. And so ISDN has been left to a lingering demise, unloved and unwanted by all but a few.

Fortunately, the tragedy of ISDN was not lost on the world. Those behind ATM saw what happened to ISDN and moved quickly to take ATM out of the hands of the CCITT by forming the ATM Forum and driving the development of ATM themselves.

We shall get to that story shortly; however, we have to take a quick look at SDH/SONET and see what it's all about first.

Section 7 Synchronous Digital Hierarchy—The Need.

As we saw at the end of section 4, the first attempt at a long-distance digital telephone system was not an unqualified success. Although extremely effective and certainly a vast improvement over the nearly century-old analog telephone system, the Plesiochronous Digital Hierarchy (PDH), as this digital system is generically known, certainly had its faults, shortcomings and peculiarities. For those of you who skipped that section, these were in brief:

- The American and European telephone systems had little in common. Expensive conversion equipment was need for transatlantic telephone calls.
- Equipment built to the “same standard” by two different vendors often didn’t interoperate, thus requiring the same manufacturer’s equipment to be on both ends of a particular long-distance link.
- There were virtually no self-checking and maintenance features in the switching equipment. This required expensive technicians and expensive test equipment to be located at virtually every telephone exchange.
- There were no standards for very high bandwidth links. These were all ad hoc or proprietary designs.
- The network was not synchronous at greater than DS-1 bandwidths.

This last point, which is surely the most obscure, was also the most important one. To understand it, you need to know that the word “synchronous” has a specific meaning to a telephone engineer. Synchronous means that bits from one individual telephone call are always located in the same position inside each digital transmission frame. DS-1 digital lines, for example, are synchronous because the twenty-four telephone calls being carried by it are always in the same time slots (see section 3.4.2). Hence, it is easy to remove or insert a telephone call in the DS-1 frame as it moves through a time division multiplexed switch.

Because DS-2 and above digital lines were almost always multiplexes of multiplex lines (i.e. DS-2 is four DS-1 lines, each of which consist of 24 DS-0 lines), it was impossible to exactly synchronize all the signals because of the vast number of sources involve and their individual variations. This is generally called jitter. The way the telephone engineers corrected for it was by stuffing extra bits here and there as necessary to make everything come out straight. Thus the system became known as plesiochronous which means “almost synchronous” because these bits were stuffed in just about anywhere, and you didn’t know for certain where a particular phone call was located from one transmission frame to the next because the number of stuffed bits could vary.

This means that the only way to find a specific telephone call in the tangled mass of DS-2 and DS-3 multiplexed lines was to demultiplex everything in exactly the reverse order they were originally multiplexed in and filter out the extra bits that were stuffed in. Then after breaking the DS-3, for example, into seven DS-2s, and these DS-2s into four DS-1s, it was possible to go into the DS-1 (which is synchronous) and tease out a specific phone call and replace it with a new one.

Therefore megabucks of equipment are needed at each and every telephone exchange simply to switch individual phone calls on and off the various multiplexed links they have to ride to get wherever they are headed. And all because of those bits!

7.1 Pointers—The Solution.

Naturally enough, the telephone engineers struggled with this for a number of years. Part of their problem is that they were in love with HDLC (which is a bit-stuffing network protocol) and used it in various applications: X.25, LAP-D, and Frame Relay are good examples. They had stuffing bits firmly lodged in their thinking. They had to break that habit, but how?

Then one of them must have taken a C language course and discovered pointers. Overnight, their thinking changed. Instead of stuffing bits to offset the inevitable jitter problems, why not simply let the contents of the whole frame shift back and forth a byte or so and have a pointer in the header of the frame that points to where the data is in that particular frame? While not truly synchronous, the pointer idea gave them exactly what they wanted. It was now possible to go directly into even the most heavily multiplexed link and, by simply using the offset supplied in the header, pull out whatever they wanted. Even better, you could even replace it on the fly. This was a multibillion dollar savings by itself.

Now that they had the basic technology, the telephone engineers began looking at all their other problems and headaches. It was clearly time to replace PDH with something better, and the engineers at Bellcore (which is the group that defines specifications for telephone equipment for the baby Bells) came up with a great set of ideas. They called it Synchronous Optical NETwork or SONET.

Since one of their major headaches was handling the ever increasing international telephone activity, Bellcore even went so far as to try to establish SONET as an international standard. In 1987, Bellcore took its proposal to the CCITT. The Europeans hated it. There were dozens of objections. The most important was that the original proposal for a 49.9 Mb/s link, which was fine for encapsulating DS-3, was not useful for the European CEPT standards. Let’s look at the European and American PDH standards again. We also threw in the Japanese PDH as well as the international transmission rates to show how complex this got.

Table 1 The Plesiochronous Digital Hierarchy for America, Europe and Japan, with the international transmission rates shown as well. The data rates are in thousands of bits per second.

Hierarchical Level	American DS-x	European CEPT-x	Japanese	International
0	64	64	64	64
1	1544	2048	1544	2048
2	6312	8448	6312	6312
3	44736	34368	32064	44736
4	139264	139264	97728	139264

As it turns out, the above table is misleading as the American DS-4 rate shown is actually DS-4NA, which exists only because the international transmission of level 4 is agreed to be at 139 Mb/s. That gets converted to three DS-3s the instant it hits America because DS-3 is the most important transmission rate in the United States. On the other hand, CEPT-4 is the dominant long-distance transmission rate in Europe while their CEPT-3 is also common. Naturally enough, the Europeans insisted that any new standard support those two bandwidths in a reasonable fashion.

To complicate things, the Japanese for some reason chose to follow the American standards for hierarchy level 0 through 2 and then do their own thing for the two highest levels. Naturally, they had their own set of complaints.

For a while, it certainly seemed that the American proposal was going to be thrown out. Everybody knew, of course, that if it had, the Americans would simply mutter some nasty words, go back from where they came from, and do exactly whatever they wanted anyhow. After all, it that was what they had done for the past seventy years, so why should they change now?

Things looked ugly. Then reason prevailed, and in an extremely short period of time (just a few months), a number of compromises were reached.

First, the basic rate for SONET was increased from 49.920 megabits per second to 51.840 megabits per second. This was basically a concession to the Europeans, who insisted on a large amount of bandwidth for operations, administration, and maintenance (OAM). If anything, they were ahead of the Americans in automation of maintenance and reducing the required staff. In hindsight, the Americans are now quite happy that they agreed to it because the American communications environment has gotten extremely competitive in the last few years.

The second set of compromises were from the Europeans. They dropped their insistence that their levels 2 and 3 be directly supported (that is to say, the 8.448 and 34.368 megabits per second rates). This left the American DS-3 and the European CEPT-4 rates (the Japanese were simply out-voted on their 97.728 megabits per second fourth level). From this came SONET/SDH, the marriage of the two sets of requirements.

Although it would be tempting to branch off at this point into all sorts of diagrams showing the mapping of all this into the actual frames, it would serve no purpose. If you are interested, there are many fine books on the subject. What we are interested in are the concepts and what they mean, not the bits. So let's look at the common set of SONET and SDH standards in terms of bandwidths. There are only three. We are also showing how many DS-3s or CEPT-4s each can support. The "OC" stands for "Optical Carrier" while "STM" means "Synchronous Transmission Module."

Table 2 The transmission bit rates (in megabits per second) for the SONET/SDH international standard, showing the hierarchy level the SONET OC-x standard, the number of DS-3 links supported by that OC standard, the ETSI-SDH STM-x standard and the number of CEPT-4 links supported by that standard.

Hierarchy Level	Bit Rate (megabits per second)	SONET	DS-3s	SDH	CEPT-4s
O	155.52	OC-3	3	STM-1	1
1	622.08	OC-12	12	STM-4	4
2	2,488.22	OC-48	48	STM-16	16

As you can see, the three defined bandwidths support multiples of DS-3s and CEPT-4s. That means that SDH (which is its correct international name) is truly a high-bandwidth communications network. Because of that, it runs only on fiber optic (except during switching operations). The interesting—if not absolutely amazing—thing, is that the same frame format (yes, we did say *same*) is used to support either the DS-3 or the CEPT-4 transmission frames. These are encapsulated within the user payload of the SDH frame, and two sets of pointers are supplied so that either format can be carried. This makes conversion at international boundaries much easier and simpler, although not transparent. There are still a number of minor differences between CEPT-4 and DS-3 encoding such as how strings of zeros or ones are handled. However, these are trivial when compared to the original nightmare.

The official CCITT standards for this subset of SONET and SDH are G.707, G.708 and G.709. These specify the above three bandwidths based on the conjunction of DS-3 and CEPT-4 bandwidths, with a lot of extra bits left for OAM and other support features. Please note, however, that there are aspects of both the SONET and the European SDH standards (called ETSI-SDH) which go well beyond the CCITT standards and are not compatible with one another. However, the above three specifications define the *subset* of both which are to be used internationally and are therefore intercompatible.

While this may seem limited, it has made the SONET and ETSI-SDH variants of the standard the basically the same because the CCITT subset is what the manufactures are building. A good example is the Bellcore SONET specification. It defines OC-1, OC-3, OC-6, OC-12, OC-18, OC-24, OC-36 and OC-48. The transmission equipment actually available, however, supports only OC-3, OC-12 and OC-48. The reason for this is economics. With relatively simple changes in the firmware (none in hardware), the same equipment can handle either SONET or ETSI-SDH. In fact, at least one company already has an ATM switch that can run either ETSI-SDH or SONET according to the position of a switch. Thus while the CCITT SDH specifications may be a subset, they are nevertheless the de facto standard for the entire world.

7.2 What Can SONET/SDH Do for Me?

Now that we have covered how and why SONET/SDH came about, let's look at what it can do. Most people immediately say, "It runs ATM!" This is true but misleading. As you can see, it was designed to handle the old DS-3 and CEPT-4 PDH standards. ATM was supported later.

First of all, as far as design is concerned, SONET/SDH is completely independent of ATM as ATM is independent of any physical media. As we shall see in the next chapter, ATM is like a string of little pickup trucks streaming down the SONET/SDH Information Superhighway. That is because SONET/SDH is simply the Information Superhighway, not the traffic running on it. Thus, although ATM and SONET/SDH are almost inevitable used in conjunction with each other, they are separate things.

In fact, SONET/SDH can do many things besides carrying ATM. For one thing, it carries DS-3 and CEPT-4. This is a more important requirement than ATM because the PDH equipment already in place has to interoperate with SONET/SDH for it will be years before all of the old PDH telephone equipment can be replaced. Now, at least, the old PDH transmission protocols can ride on the optical superhighway.

One interesting thing about the SONET standard is that there are many mappings of the user load. For example, besides stuffing a SONET/SDH frame full of ATM cells, or DS-3 transmission frames, you can also put a number of FDDI frames into it. This would permit somebody with a need to run FDDI thousands of miles a way of doing it. (Admittedly, however, it would be far cheaper to use ATM.)

Finally, there is what is called *virtual container* which is a way of mapping a number for T1 lines and other low speed communications into the user space of a SONET/SDH frame.

In summary, SONET/SDH is really nothing more than a road: It is the Information Superhighway. Perhaps a better analogy would be an Information Super Railroad for SONET/SDH is more like a long train of containers cars (i.e. frames) that move around the world over fiber optical rails. And what is inside of each container can be about anything: DS-3, FDDI, CEPT-4, or ATM.

We'll leave our exploration of the Information Super Railroad here because we, as computer networking folks, are interested in what is inside those container cars. In particular, we are most interested in the type that carries ATM.

Section 8 Asynchronous Transfer Mode—The Need.

Asynchronous Transfer Mode, or ATM, as it is better known to us, is more of a revolution than an evolution. Although steeped in the traditions of digital telephony, ATM is in many ways a break from the past. Many cherished concepts and requirements once thought to be inviolate were simply scrapped by ATM. However, before we look at ATM in detail, let's look briefly at the cultures of the telephony and local computer networking worlds and see how they influenced each other. It is from this interaction that ATM was molded and, indeed, is still being molded for ATM is not yet a complete specification. There are several glaring holes which limits ATM's effectiveness in the LAN environment. The battle over the resolution of those holes is currently raging in the ATM Forum. How these are resolved will determine what ATM can do for you. Therefore, we will look at to the two sides of this debate. Not surprisingly, the two sides are the telephone engineers and the networking engineers.

As recently as five years ago, those in the telephony and local computer networking worlds had almost contrary needs and cultures. The telephony world, on the other hand, saw communications as wires strung from point A to point B so that two individuals could communicate with each other for a short period of time. Even though the telephone people did permit the networking people to use their wires, it was on their terms—lease a telephone line and do what you want, but you're still going to have to pay for it 24 hours a day, seven days a week. Their culture was and still is very businesslike. They also saw everything from a connection-oriented viewpoint. The reason, of course, is they wanted to bill for the services used.

The local networking counter-culture, on the other hand, was much more oriented toward group behavior and sharing the network, and they were horrified by the thought of charging money for using it. The network was seen as a group resource, as a way to communicate and commiserate with each over via e-mail, bulletin boards and interest groups. It was there to be used when you wanted, shared with others much like a public road. Thus when you had a large file to copy to your friend's system, you started the task to do so, and if others were doing similar work, you simply shared out the available resources on a fair and equitable manner. Eventually everyone got their work done. Time was not of the essence.

Over the last few years, however, the computing world changed. No longer was it the domain of the computer hacker. Other needs for other activities have developed. One is the multimedia revolution. Instead of sending e-mail which were simply text files that could take hours to arrive at their destinations, we are now into video conferencing, where something close to real-time performance is required. Next came the idea of movies on demand. This technology not only requires real-time performance, but also a guaranteed-rate-no-matter-what performance.

Likewise the telephone culture has changed. Just a few years ago the attitude was a somewhat arrogant "lease a line, they come in three sizes." Not much choice was given to the user. Either you used what was offered or you did without. Deregulation changed that attitude. No longer was AT&T the monolithic Ma Bell who knew what was best for you, but now a leaner and hungrier company glancing anxiously over its shoulder at Sprint and MCI. Customer service and slick advertisements have come into vogue.

This new competitive environment forced the telephone companies of the world to finally look at the real needs of their customers. Three sizes of leased lines didn't do what was needed. Besides, only the rich could afford them; and in any case, even they were bitching about the cost. Clearly what was needed was some convenient way to offer each user the bandwidth he needed when he needed it. Furthermore, voice communication was no longer the sole focus. The telephone companies became aware of the multimedia revolution and realized that they had a golden marketing opportunity. All they had to do was find a way to capture it.

ATM is the child of that environment. The primary goal was to find some way of easily dividing up the incredible bandwidth now available on the SONET/SDH Information Superhighway in such a way so that everybody could get what they wanted, when they wanted it.

Clearly the original approach to digital telephony, PDH, wouldn't work. The hierarchy of DS-0 through DS-4 (or CEPT-0 through CEPT-4) was based on the concept of 64,000 bit per second digitized telephone calls which were multiplexed layer upon layer. Certainly you could use the telephone line to send computer data (as in ISDN) to whomever you wanted when you wanted, for the line was switchable. But ISDN was limited to 64,000 bits per second. If you needed more bandwidth, you could use the multiplex levels—if you had the money, but in the United States, you could only get T1 (1.544 megabits per second) or T3 (44.726 megabits per second) and they weren't switchable. What do you do if somebody wants 8 megabits per second?

A number of ad hoc solutions were offered not only by the telephone company but by entrepreneurs as well. They sold fractions of these services as fractional T1 and T3, but still these services were what we know as Permanent Virtual Circuits, better known as leased lines. You couldn't call up your mother and show her a video of your baby taking her first steps unless you paid for a month of service by leasing the line and having it installed. And you are not likely to do that even if you had the money.

The telephone engineers who had just recently discovered the cure for their bit-stuffing blues by using pointers in the new SONET/SDH fiber optic telephone system again looked at what the networking people were doing. To their surprise, they found that the networking people had long ago solved the problem of dynamically sharing out bandwidth with something called packets.

The idea is simple. You put the data into nice little containers of, say, 1,500 or 4,352 bytes (ethernet and FDDI, respectively), then you pump them over the network to wherever they were going. If there were a number of other people trying to use the network, you got to send fewer packets per second so that others could also have access to the network as well; but you still got the file there. Suddenly, it became clear. The issue wasn't how many bits per second you sent, it was the number of packets per second. It sounds obvious, but the idea was quite revolutionary. The mind set of the entire telephone industry had to be changed. And they sort of succeeded in solving the problem. But, unfortunately, it wasn't a complete break with the past. They still haven't gotten away from their mind set that bandwidth must be preallocated and remain static for the duration of the connection.

Actually, this fantastic discovery should have come as no surprise to the telephone folks for packets were not exactly new to them. Long ago, they invented X.25. However, it was strictly a computer network, with store-and-forward capabilities that absolutely guaranteed delivery of the data. True, it might take hours for a packet to arrive at its destination if there was congestion, but the X.25 network was absolutely reliable because an X.25 node never deleted a packet until it was positive that the next node in the chain had correctly received that information.

This, of course, is totally useless in the real-time world of video conferencing and movies on demand. You needed timely response. The telephone engineers noted that little short coming as a problem to be solved.

Next, the telephone folks looked at their experiment with ISDN. In this environment they used packets to set up telephone calls and regulate the system. However, that was on the D-channel. While ISDN did give the user two 64,000 bit per second B-channels, he couldn't legally connect them together although many users often did using proprietary protocols. But still even 128,000 bits per second wasn't enough for video or movies. In addition, the B channels were still plain old telephone lines. True, they were digital; and true, you could send packets over them if you wanted, but they were still the basic DS-0 service which was an integral part of the PDH everyone now wanted to get rid of. So while ISDN was an interesting experiment, it wasn't the way.

Finally, in what must have been an interesting meeting, somebody suggested that they, the telephone system, convert everything to packets—including voice communications. We can see it now: A long table surrounded by telephone engineers, white shirt sleeves rolled up to their elbows. They're all staring in utter shock at the bright young person who just proposed such a radical idea. Grumbles could be heard. However, reason won the day. The packet was clearly the solution. The only problem was how to clean up the mess those networking folks had made of the idea with their complex protocols and regulating mechanisms. This is the crux of the current debates—the networking people feel strongly that those complex protocols and regulating mechanisms are vital.

Turning back to the networking world, we should remember that their primary focus was on reliability in an unreliable environment. The world of networking engineering is filled with thousands of others doing their own thing without much outside regulation. True, there were standards. However, these were subject to countless different interpretations. Interoperability was nightmare. Eventually solutions were found, and from them came a mind set which, if anything, was more conservative than that of the telephone engineer. And for good reason, for interoperability—the ability of equipment from many different suppliers to function together in a heterogeneous environment was critical.

In addition, the network engineer strove mightily to overcome the chaos to be found on the LAN. It was often overfilled with bursts of activity when dozens of people decided to use the network at the same time. Data could get lost as packets collided and routers choked.

Thus a number of checks and balances were developed over the years. One is the idea of Media Access Control (MAC). What that translates to is a traffic cop who regulates the activity on the network so the traffic can flow smoothly. One well known version of MAC is the collision of packets in ethernet. When two or more people try to use the ethernet at the same time, the packets collide and everybody backs off and then tries again until they get a clear channel. Some get to do their work before others, but eventually everybody gets their turn. Another example of MAC is to be found in FDDI and token ring: They both use a token that is rotated around the various systems connected to the network, and only the system that has the token may transmit.

A second common feature of computer networks is that there are built-in checks of data integrity. These are the so-called CRCs or Cyclic Redundancy Checks. Thirty years of experience has proven that these are necessary. Strange things can go bump in your network and trash your data. Without a robust CRC, you might never know that your data was trashed.

From the telephone engineers' point of view, however, all that was wasteful overhead—mainly because they never experienced it. For example, the telephone engineer had the luxury of qualifying the equipment that was connected to the telephone system. Second, the telephone engineer's view of a network is not of a shared bus or ring, but of a switched network with individual wires that are temporarily connected exclusively from one end-point to another end-point. Furthermore, the bandwidth need for that connection is fixed, dedicated and preallocated. This, of course, is the telephone call. This implies several things:

- The full bandwidth of the physical media used is available between those two end-points. Sharing the media with others is not an issue, even in the case of multiplexed long-distance line. During set up (i.e. "making a phone call"), each individual phone call is still allocated a virtual circuit and 64,000 bits per second is set aside all along the path between the two end-points for that phone call to use. Thus the bandwidth of the virtual circuit is known. It is also fixed, which is wasteful from the networking engineer's viewpoint. But, traditionally, the telephone engineer wasn't concerned about that because the customer was paying for it, regardless.
- Contention, if any, is also controlled by those end-points in a telephony view of the network. That is, one person listens while the other speaks, then they exchange roles according to their own set of rules. In other words, it's the customer's problem to figure out how they speak to each other.
- Lost data is not that important. The typical user can either do without that data or fill it in themselves. And if there is a really bad disruption, one end-point can simply ask the other to repeat what was said. This is what is commonly called higher-level recovery.

In short, the telephone engineer is merely interested in supplying a wire from point A to point B. What happens on it is the customer's problem. With this mind set and with the intention of making packets work in a nearly real-time environment, the telephone engineers set to work and came up with a packet switched network the likes of which had never been seen before. In many ways the design was brilliant. It was certainly revolutionary, for it basically abolished many of the long-held concepts about telephony. Two of the most important were:

- Everything including voice telephone calls would be transmitted in packets. There would be no more physical connections.
- The digital hierarchy concept such as twenty-four DS-0 lines being multiplexed onto a DS-1, which in turn is multiplexed onto a DS-2 and then onto a DS-3 and DS-4 would go away. This is particularly interesting seeing that SONET/SDH is just such a hierarchy. However, as we noted, SONET/SDH and ATM have little in common in their design.

Unfortunately a third, long-held concept which was *not* abolished is the concept that fixed bandwidth should be allocated for the duration of the connection. This is not surprising, for the telephone companies bill you on the class of service (which includes the bandwidth as a primary parameter) and for the duration of the call. In addition, this procedure would be all but impossible to modify because it is regulated by governmental organizations.

The network engineer, in contrasts, unencumbered by tariffs, rules and billing procedures, has long looked at networking as truly dynamic bandwidth—it was shared on an as needed and as available basis.

From a networking viewpoint, the decision (or requirement) for fixed bandwidths during a call was a waste. That is because the ATM view of SONET/SDH could be that the various levels of the digital hierarchy are simply progressively larger boxes (i.e. transmission frames) into which a larger number of ATM cells can be packed. This is because multiplexing can be done solely at the ATM cell level. However, it isn't. ATM connections are presently fixed bandwidth for the duration of the connection. If you have nothing to say, you are supposed to send null cells (i.e. empty cells). Although it is possible to make the bandwidth allocated anything that may be desired, it remains fixed at the value as long as the connection exists. If you want to change it, you must break the connection and make a new one. After all, that's the way the telephone system works.

In addition, the maximal bandwidth possible for an ATM connection is presently 155 megabits per second (we'll see why near the end of this white paper). This is, in many ways, the true schism effecting ATM today. The network engineers want to make the utilization of bandwidth on the Information Superhighway truly dynamic, while the telephone companies want to keep it fixed for each connection. However, before we go into this further, we need to look at ATM in greater detail, which is done in the next section.

Section 9 ATM, the Solution.

As noted in the beginning of section 8, there was a clear need to get rid of the whole concept of a digital hierarchy because of the widely varying needs of the modern telecommunications environment simply didn't fit into it any longer. These needs range from the simple voice telephone call to movies on demand. What the telephone engineers came up with was ATM.

9.1 What ATM Means and Why It Was Invented.

As repeatedly noted, ATM stands for Asynchronous Transmission Mode. Just what does that mean? Since we are near the end of this white paper, we should probably define it before we are done.

To start with, almost anyone will admit that "ATM" is a terrible name to give a product because of the instant confusion with those little machines at the bank that give us money. It is bad marketing.

Actually, "ATM" was never meant to be used by the general public anymore that Plesiochronous Digital Hierarchy was. It's telephony jargon for a system in which you don't know what is where in the transmission frame. Because we have already covered why the telephone people are so eager for a synchronous digital system, we should explain why the same people went out and invented a totally asynchronous (from their view of point) transmission system. The answer is the need for variable bandwidth.

If you think about ATM as a string of pickup trucks, each with some destination stamped onto its header, you can see several interesting possibilities. One is bandwidth allocation. If Mr. X needs ten times as much bandwidth as Ms. Y, then you simply give him ten times as many pickup trucks per unit of time. The idea is fairly obvious.

As we have seen, the SONET/SDH system was designed to replace the older and somewhat inefficient PDH system that has served us for so many years. However, the basic problem of an easy and convenient bandwidth allocation scheme is still lacking in SONET/SDH. Instead of having a choice between three relatively low-speed cables as in PDH, you now had a choice between three relatively high-speed cables in SONET/SDH. As far as bandwidth allocation is concerned, nothing changed; it simply got faster.

The reason this happened was that the whole design of PDH and SONET/SDH is based on the concept of switching telephone lines, not packets. And since they were using time division multiplexing (TDM), the telephone engineers strove mightily to make everything synchronous (that is say, in the same place from transmission frame to transmission frame) so that they could use TDM conveniently and easily.

There is an alternative, however. It is called statistical time division multiplexing (STDM) which gives the multiplexer the ability to give one user more bandwidth than another. Most versions of this technique require a little flag attached to each packet of user information so that the receiving multiplexer can figure out where to send each and every packet. And if this sounds like an ATM cell with a five-byte header to you, you are absolutely right. ATM is nothing more than a form of STDM. The difference between it and other forms of STDM is that the little header is left on for the whole journey, from end-point to end-point. The reason why it is called Asynchronous Transfer Mode is simply a telephone engineer's way of saying that all the cells traveling to wherever they are going are all mixed together in any order. And for once, the telephone engineers don't care because all the switch has to do is look at the header and send the cell down whatever line or circuit that is appropriate. It's basically a sorting function.

Thus all an ATM switch does is take the incoming frame, split it open and sort all the ATM cells inside into piles according to where they are headed. Then on the output part of the cycle, it packages each pile of ATM cells up in a new transmission frame and sends it on its way. This all happens very fast, in 1/8000th of a second.

A good analogy is the way mail is handled. You drop your letters in a mailbox. Somebody comes along, collects them into a bag and takes the bag of letters to the local post office. There the clerk opens the bag and sorts all the letters by, say the first two digits of the zip code (a.k.a. postal code in Europe and the rest of the world). Eventually each pile of letters is bundled up, packed into a bag, and sent to another post office closer to the ultimate destination of those letters. There the processes is repeated, but let's say they are now sorted by the second two digits instead of the first two. Again the new piles are bundled up and shipped to a third post office, where the final sort according to the last digit occurs.

9.2 ATM as a Hierarchy-Free Telephony Standard.

Clearly, ATM is a way to get away from the digital hierarchy concept that permeates long-distance telephony. But SONET/SDH does not yet permit this.

Presently, SONET/SDH is strictly a hierarchial environment at the OC-12 and OC-48 levels. In other words, OC-12 only carries four OC-3 "pipes" and OC-48 only carries four of these OC-12 "pipes." That is, there is no version of SONET/SDH at this time that makes the entire 622 megabits per second of OC-12 or the 2.4 gigabits per second of OC-48 available for a single channel of information.

On the other hand, there are a number of different mappings of OC-3. Some are designed to carry three DS-3 trunks or other relatively low speed telephone communications links. And as in the case of OC-12 and OC-48, OC-3 these are a multiplexed communications channel. However, there is a special mapping of the OC-3 frame that gives the entire frame over to one user or channel. This is called OC-3C for OC-3 Concatenated, and exists mainly to carry either ATM or FDDI over SONET/SDH. Thus, today the fastest SONET/SDH service supporting ATM is OC-3C.

However, there is no reason why there can't be a OC-12C. Indeed, a number of people are already pushing for it because it would give them a single data stream of 622 megabits per second. And if that can happen, then why not a OC-48C?

Well, the answer to both proposals is that the hardware to support them doesn't exist—yet. But that will be resolved before long. Thus should the hoped-for OC-12C and OC-48C specifications ever become reality (and we are betting that they eventually will), then the ATM cells can view the SONET/SDH Information Superhighway as a number of progressively larger bags each cell has to live in while in transit and effectively convert the digital hierarchy into three sizes of packages.

True, the switches to support this will have to do a phenomenal amount of work, but—in time—they should be able to open these frames full of ATM cells, sort them out like letters in the post office, repackage them into OC-3C, OC-12C and OC-48C sized bags, and then send them on their way without the hassle of having to multiplex and demultiplex multiplexed lines as we presently have to.

However, please remember that a hierarchy-free digital transmission system is a future promise of ATM, one that is not likely to be realized for many years yet.

9.3 The ATM Cell.

We suppose some of you are interested in at least a brief overview of just how ATM works. This section is for you. Those of you not interested in the bits and bytes can safely move on to section 9.6. We will start with the cell, as it is called and then work our way through the other parts of ATM.

Starting with the cell header, it is simplicity itself.

Figure 6 The contents the ATM Header. The bit position in each byte is shown at the top of the figure.

```
Bit           8    7    6      5    4     3    2    1

Byte 1   |         GFC or VPI     |       VPI            |
Byte 2   |         VPI            |       VCI            |
Byte 3   |                     VCI                       |
Byte 4   |         VCI            |       PT       | CLP|
Byte 5   |                     HEC                       |
```

In the ATM header,

- GFC means Generic Flow Control. Basically, this was put into the specification to support such things as Metropolitan Area Network. By in large, it is not used, or it is used to expand the VPI.
- VPI is the Virtual Path Indicator. This is from our old friend X.25, where it ended up not being used. However, it is used in ATM. It is the virtual path through the series of ATM switches an ATM cell has to follow. In many ways, it is the same as the Virtual Circuit Number of X.25. The actual VPI changes from one node to the next, and the actual values are local to each ATM switch the virtual path traverses (just as in X.25).

- VCI is the Virtual Channel Indicator. This is a bit more subtle than a virtual circuit. It is more like a lane number on the VPI roadway and usually remains the same from one ATM switch to the next. In fact, a number VCIs are predefined. This field permits many channels of information to be concurrently transmitted over the same Virtual Path. Some of these channels are used for user data, and others are used for network control, much as the D channel in ISDN was used. However, there are a large number of OAM (overhead, administration and maintenance) and control channels in ATM instead of just the one (i.e. the D channel) used in ISDN. These usually have preallocated VCIs.
- PT means Payload Type. These three bits define a number of features that help sort out the various OAM features in ATM.
- CLP is Cell Loss Priority. If set, then an ATM switch can discard the cell if necessary. If set to zero, then the ATM switches should not toss the cell, although they still might.
- HEC is the Header Error Control. It’s an 8-bit CRC of bytes 1 through 4.

That’s it. That’s all there is to ATM. Well almost, because there are some five variants of the cell, not to mention how you get everything stuffed into cells and then reassembled at each end of the connection. First, we’ll look at the ATM Adaptation Layer (AAL), which defines some details of the ATM cell and how it is used, and then we’ll look at the way the user’s data is segmented into and reassembled from these cells.

9.4 *The ATM Adaptation Layer, or AAL 1 Through 4.*

Before the ATM Forum literally usurped ATM, the CCITT defined four types of ATM cells that were designed to handle the various types of data envisioned on the ATM network. They ranged from voice calls, to audio and video, to computer data. The different requirements imposed by these needs were originally to be handled by four different services called Class A through D. Specifically, these are as follows:

- Class A: This service offered a constant bit rate in connection-oriented circuits that also required some level of reliability and error detection. Therefore, there is a sequence number associated with each cell like the one found in TCP/IP. Sequence numbers permit quick and easy detection of lost cells and hence give the receiving station the ability to immediately recognize the problem and request a retransmission, if desired.

  As part of the constant-bit rate requirement, there is also a requirement for timing synchronization between the transmitting and receiving station. This means that the delays going each direction on the circuit are minimal and constant. Typical uses for this service would include voice over an ISDN (yes, we mean narrowband ISDN) or a yet to be defined ATM voice service where excessive delays would intolerable. In addition, because the ISDN telephone would be limited in functionality compared to a computer, it would not have much buffering and so requires a constant bit rate.

This service could be also handle other types of data, but it is expensive because of the constant bit rate feature. The AAL format that handles this service is the called AAL type 1. It uses one byte of the 48 bytes in the user payload for the sequence number, leaving the user with 47 usable bytes.

- Class B: This is service similar to Class A but is designed to handle video and audio. As such it does not require the constant bit rate if the receiving device has even a modest amount of buffering (a few hundred bytes). However, it would still require the clock synchronization, and it would be connection-oriented.

  The AAL type 2 packet supports this service. Here the CCITT went a little overboard for they added quite a bit of overhead in each cell for the eventuality of lost packets, scrambled data, and even variable-length data streams. Thus there are three bytes of overhead in the type 2 AAL format, giving such information as sequence numbers (as in Class A), data length, and even a 10-bit CRC. Planned uses for this service were, as noted, audio and video. In addition, high quality computer data transmissions such as those required by real-time tasks (i.e. aircraft simulation) could use this class of service. Unfortunately, the overhead left only 45 bytes of user information in each cell.

- Classes C and D: These two services are designed explicitly for computer data, particularly data that is sensitive to loss but not delay. These two classes of service do not support real-time or timed activities. Class C is connection-oriented and Class D is connectionless. They are handled by the AAL type 3 and 4 formats respectively. The two services are similar and use almost the same AAL format which uses four bytes of overhead, leaving the computer user with only 44 user bytes per cell.

9.4.1 The ATM Forum and AAL Type 5.

As noted several times before, a number of people watched ISDN get lost in the bureaucratic maze of the CCITT for ten years and arrive virtually still-born, overwhelmed by the technological promises of SDH/SONET. They weren't about to let the same thing happen to ATM. So a number of "concerned users" of ATM formed an ad hoc committee called the ATM Forum and simply took ATM away from the slow-moving CCITT committee. It is of interest that the founding fathers of ATM Forum consisted mainly of long-distance telephone companies. Later, the computer manufacturers joined the ATM Forum when it became obvious that ATM would be important to computer networking. As we shall see, the late arrival of the computer and network router manufacturers in the ATM Forum has caused some friction which is still unresolved.

By and large, however, most will agree that the ATM Forum has been a good thing. Many issues which the CCITT could not come to grips with, such as network management, were settled in weeks if not days by the ATM Forum. And many other, but not all, serious issues left unresolved by the CCITT were soon settled by the ATM Forum.

One of the first of these issues was the performance of ATM. Just about every computer manufacturer found that the proposed Class C and D services, now better known as AAL type 3/4 were inadequate. They therefore invented a fifth class, known as AAL type 5. Simply put, it is a raw cell. There is no overhead in the cell itself as in the other types, leaving all 48 bytes available to the user. This leads to an immediate 10 percent performance increase because the overhead in AAL type 3/4 has been eliminated.

In addition, the cell-by-cell overhead of the 10-bit CRC was also removed. Since the cell level CRC proposed by the CCITT (it was only 10 bits) was inadequate protection (only one bit could be corrected), the computer networking folks simply got rid of it. This is not such a harebrained idea as it may seem for the expected ATM transmission media were all highly reliable as far as bit errors are concerned (fiber optics usually have less than one error in every one billion bits). So why waste space in every cell for a generally useless CRC when you can do it at a higher level instead? Thus, the now preferred AAL format for computer data is type 5, mainly for these two reasons.

9.5 Sending Computer Data via ATM.

Now that we know what the format of the ATM cell is for computer data (i.e. AAL type 5 with five bytes of header and all 48 bytes of the user payload available to the user), we can now move to the question of how the user's data gets from his computer system through ATM and into the recipient's system. This is done in several steps, most of which are old hat. The example we'll use is a TCP/IP transmission. We'll skip the details of TCP/IP because they are common to ethernet, FDDI, token ring and a number of other physical layers, including ATM. We'll start in the user's program.

9.5.1 ATM's Convergence Sublayer.

The user's program does a write, or equivalent system call, which tells TCP/IP that he wishes to write data out on the network. As it has since the days of thick net ethernet, TCP/IP does its thing and eventually IP sends the frame of user data wrapped up in a TCP and IP headers to the ATM driver. (If you are curious about this, the document that defines the TCP/IP ATM encapsulation is RFC 1483. It was published in July 1993.) The entire package of encapsulated user data is usually referred to either as a packet or a frame. We'll use the term frame.

These TCP/IP encapsulated frames are about 64 Kbytes long (i.e. 65,535 bytes) or less when delivered to ATM by IP. In ATM terminology, the layer of ATM that first sees the frame is called the Convergence Sublayer. All it really does is to add its own header and trailer so that the overall appearance of the ATM TCP/IP packet looks something like an onion with the ATM layer, the IP layer, the TCP and finally the user's actual data being layered from the outside to the center.

(Actually, there are more layers to this onion than this, but most of them are transparent to the average user so we left them out.) It would look something like figure 7.

Figure 7 The layers of protocol information surrounding the user's data in a ATM data packet in the Convergence Sublayer.

| ATM Header | IP Header | TCP Header | USER'S Data | TCP Trailer | IP Trailer | ATM Trailer |

In the case of ATM, the actual format of the header and trailer information varies according to AAL type. In the case of ATM AAL type 5, the trailer information for the frame includes a 32-bit CRC for error detection, which covers the entire 64 Kbytes of the IP/TCP/User's data frame inside. On the other hand, in the case of AAL type 3/4, there is no ATM frame level CRC. Thus, AAL type 3/4 depends solely on the individual 10-bit CRC included in each ATM cell. Therefore, although both AAL 5 and AAL 3/4 both have CRCs, the one for AAL 5 is both easier to calculate (there is only one per frame) and more rigorous (it's 32 bits long instead of 10 bits).

In summary, the Convergence Sublayer is simply wraps whatever it is given by higher layers (we were only using TCP/IP as an example, GOSIP, X.25 or just about anything else could have been given to the convergence level) in a header and trailer so that its colleague in the receiving system can determine if there were any transmission errors. In the case of AAL type 5, this also includes the calculation of a 32-bit CRC, which is almost exactly the same functions you find in the code that handles the encapsulation of ethernet packets and FDDI frames for those two media.

The next stop down the chain is the Segmentation And Reassembly Layer of ATM.

9.5.2 ATM's Segmentation And Reassembly (SAR) Layer.

This layer of ATM is usually handled in special hardware for a number of reasons. Generally, the functions of this level are to slice and dice the ATM frame delivered by the convergence sublayer into neat little 48-byte cells. It also does the reverse procedure, gluing a large number of ATM cells back into a ATM frame. It is thus for these reasons that this layer of ATM is called Segmentation And Reassembly (SAR).

Additional features of SAR include calculating a CRC per cell in the case of AAL type 3/4, as well as filling in the information in the five-byte header. At this point the ATM cell is complete and ready for shipment by the physical layer. However, before we bounce down to that, we will explore some of other features of SAR. Most specifically, why it is done in hardware.

9.5.2.1 The SAR Chips, Implementation Features, and ATM Performance.

One of the favorite complaints one reads in the popular networking magazines and newspapers is that ATM is inefficient because all those 48 byte cells create too much overhead. This, unfortunately, can be true in brain-dead or brain-damaged implementations of ATM.

The issue is based on the long-held observation that the larger the frame, the more efficient the communications. This is usually true. Many of these same writers then latch onto the 48-byte cell size ATM and assume that its frame is only 48-bytes long. As we have seen, this is not at all true. The ATM frame is as much as 64 Kbytes, which is considerably larger than FDDI's frame.

Whether or not ATM is efficient actually has little to do with the size of the cell if the overall implementation is reasonable. This is the major reason why the SAR level of ATM should be in special hardware so that it doesn't create millions of interrupts per second on the main processor as some ATM implementations do. This activity can and should be handled by the SAR chips. It's only when the frame has been completed or is determined to be defective that the main processor should be involved. As noted, some early implementations of ATM computer interfaces don't use such smart SAR chips, and thus their performance is woefully inadequate (about 20 megabits per second) because they overload the main CPU with interrupts. However, if intelligently designed, there is no reason for ATM to be "inefficient."

Another favorite complaint of some networking writers, particularly those pushing some other networking technology, is that there is "incredible overhead" in the ATM header. This is based on the observation that if five bytes out of 53 are used for the header, only 48/53rds or 90.5 percent of the cell is usable by the user. This is absolutely true, but they usually fail to point out that ATM is basically physical layer free and one of the trade-offs for doing that is each cell has to have some sort of address attached to it. We shall look at this particularly neat and wonderful feature of ATM right after the next section. However, let's first look briefly at why ATM cells are 53 bytes long.

9.6 Why ATM Cells Are the Size They Are.

In a word, the answer is politics. ATM is not the first attempt at using packet data for voice transmission. In the late 1970s and early 1980s, a number of researchers were experimenting with this idea as part next-generation telephony. For example, in 1982, the French built a digital voice transmission system called *Prelude*. It used an 18 byte cell, with a two byte header and 16 byte data area. This worked reasonably well, but the French became convinced that, for voice transmission, a 32 byte data field was best.

At about the same time, both the United States and Australia were experimenting with a Metropolitan Area Network now generally known as DQDB, or Distributed Queue Dual Bus. This system has gone into general use in Australia, but for legal reasons is languishing in the United States. The important thing for us is that it is characterized by a 64 byte cell.

So, when it came time to settle the size of the ATM cell in the CCITT, the French wanted a 32 byte cell, and the Australians and Americans were pushing for a 64 byte cell. Guess what happened? They split the difference and made it 48. So, sometimes hard bargaining takes the place of scientific reasoning in such things as networking specifications.

9.7 *ATM's Physical Layer.*

As noted many times already, ATM does not have a specific physical layer associated with it. In many ways it is analogous to TCP/IP, which runs over ethernet (there are three types if you look at thin net, thick net and twisted pair), FDDI, HIPPI, token ring, and of course ATM. The difference, however, is that ATM is almost a physical layer, whereas TCP/IP is a higher layer that must be encapsulated into a physical layer. Thus for a TCP/IP frame to move from, say, ethernet to FDDI or back, the entire frame must be captured, re-encapsulated and then retransmitted. This is a lot of work.

An ATM cell, on the other hand, is an ATM cell, no matter what it's running on. Although a number of them may be repackaged in the conversion from a DS-3 transmission frame to an OC-3C transmission frame, the individual cell is not changed (except that the Virtual Path Identifier might be renumbered. However, that is solely a routing issue and has nothing to do with the change-of-media issue. Therefore, one can say that a ATM cell is not changed when it changes transmission media).

This is fantastically efficient for it means that an ATM cell can start off on a relatively low-speed interface and travel through a double pair of twisted wires to a local hub. There it can be easily inserted into a high-speed fiber optic system and shoved out on the Information Superhighway for a trip anywhere in the world. Thus the analogy that an ATM cell is like a little pickup truck driving out of your driveway, through the local streets of your town and onto the interstate highway is accurate. It is also why it has caught everybody's attention.

And if we have to put up with a costly five-byte header (as some pundits harp) to have this convenience, we'll take it. The trade-off is that you have to reincapsulate your packet at every media change—and that is really expensive.

9.7.1 *ATM's Many Physical Layers.*

At this moment in time there are four or five official ATM physical media as defined by the ATM Forum. These are:

- SONET/SDH: In this media, better known as the Information Superhighway, the ATM cells are herded like cattle into a SONET/SDH transmission frame and are shipped in bulk. It is potentially the highest bandwidth media because OC-48 (STM-16 in Europe) is rated at over 2.4 gigabits per second. However, for now, ATM over SONET/SDH is limited to OC-3C, or 155 megabits per second.

There are two forms of this specification. One is a WAN version (a.k.a. your telephone company) that uses telephone equipment and single modal fiber. The other form is a LAN version which uses multimodal fiber. Most recent implementations of ATM use this latter version at the OC-3C level. That means it uses multimodal fiber. However, you can plug the LAN version into a WAN version if you have the correct termination equipment from the telephone company. (They insist on this, by the way.)

- T3: This is the first media to be brought into use. Correctly known as DS-3, it has a nominal rating of about 44.5 megabits per second. It is strictly a telephone system media due to its cost, but it is fast. Having used this media, we can say that it's great. Fortunately, a telephone company was paying the bill!
- Fiber channel: We suppose that the ATM Forum put this in to make IBM happy. Nominally, this media is rated at 155 megabits per second. It basically uses the fiber channel physical layer.
- TAXI: This media has a confusing name, but it is the name of the chip used to encode the bits. It is basically FDDI's fiber optic physical layer with all the silly overhead of tokens and Station ManagemenT ripped out. There are several forms of this media, rated at a number of different bandwidths. The official rating is 100 megabits per second, and it uses all the physical plant (cables, MIC connectors and so forth) of FDDI. This gives the FDDI user a way to upgrade to ATM. This is a local area implementation and until recently, all ATM equipment designed for the LAN market used this media. The present trend is to SONET/SDH.
- Token ring or STP: Yes, you can convert your token ring to ATM, using all that thick and hard to handle shielded twisted pair (type 1 and type2) cable. Although rated at 155 megabits per second, we do not expect this to be popular except for those who have a lot of thick and hard to handle STP cable already in place.
- Unshielded twisted pair (UTP): If anything puts ATM on the map of the networking world, it going to be the UTP standards currently being developed. There are many: One is rated at 51 megabits per second. IBM, however, is pushing hard for a 25 megabits per second standard, and everybody is talking about 155 megabits per second. The differences between all these are not only the cost of the interface electronics but also the type of UTP cable you already own or want to install. The slower speed implementations are designed to run on category 3 (a.k.a. level 3 or LVL III) wire which is the most common UTP for ethernet. The higher speed versions, such as 155 megabits per second, will require category 5 (a.k.a. level 5 or LVL V). Both types of wiring will be good for a 100 meter run to the local ATM switch.

- T1: This is an unofficial standard as yet, but a number of telephone companies are talking about it. Since this includes AT&T, it will happen. It will give you about 1.5 megabits per second. The only reason why this is a reasonable media is price and popularity (there are a lot of T1 leased lines out there).
- Others: Since ATM runs on anything that can carry electrons or photons, there will undoubtedly be other media. You can already count microwave in this as it is a carrier for both T1 and T3, although no longer popular with the long-distance telephone carriers. The Europeans have also proposed E-3 and E-4 ATM services similar to DS-3. Another probable ATM media will be wireless. About the only media that doesn't make sense to somebody someplace is two tin cans and a length of string. And undoubtedly somebody is working on that anyhow.

9.8 *ATM: The Promise and the Reality.*

This section is the most susceptible to change and may very well be out-of-date by the time you read it. This is because we are about to deal with the immediate issues and problems that still surround ATM. These, obviously, are subject to change. And since the ATM Forum is generally a fast moving group, we hope they will have the major issues resolved in a few months.

Some of the issues are rather simple. You may have noticed that we have not yet talked about placing phone calls and that sort of stuff in ATM. Well, there is still some argument over the details, so as of this date (late March 1994), there is no official standard in place for making switch connections. In other words, ATM is presently only a Permanent Virtual Circuit environment. There are, however, proposed standards. The most likely to be adopted is Q.93B, which is a slight variant of our old friend Q.931 from ISDN. Therefore, we don't consider this a real problem.

A much more difficult issue is that there is no Media Access Control (MAC) in ATM, at least in the eyes of the networking engineers. As you remember, MAC is the traffic cop of the network. In the case of ethernet, it is the collision, and in the case of token ring and FDDI, it is the token. Basically, MAC controls traffic so that congestion does not occur. This is an issue because the telephone engineers see no need for it. After all, they argue, you are simply making a phone call between two end points, and as long as you don't exceed the allocated bandwidth, there will be no problems.

The networking engineers disagree vehemently. Consider a file server connected to a number of clients via ATM. What happens if all ten clients ask for data at the same time? The answer is that the ATM switch hooking them all together will be swamped in cells, unless there is a traffic cop.

The second messy problem on the ATM Forum's plate is what to do if there is congestion. The current solution is to throw cells away, an activity that is an anathema to any networking engineer because the ATM frame at the convergence level is some 64 kilobytes long (or about 1,375 cells long), and throwing away just one cell means that you will have to retransmit all 1,375 if you want to have the data.

The telephone engineers reply that you wouldn't have that problem if you used AAL type 3/4 and the sequence number to get the missing cells only. The network engineers then grumble about losing 10 percent of the bandwidth in AAL type 3/4 when they can have it all with AAL type 5. And so it goes on and on. . . .

Presently, the arguments still rage back and forth in the ATM Forum over these two issues. As you might guess, the participants fall into two camps: the telephone engineers and the networking engineers. If you go back and review the section we did on their respective cultures, you will easily understand why the telephone engineers don't consider these real issues and the networking engineers do.

Just what will come out of all is not known. In the short term, it means that ATM is not a very good LAN. If you can have congestion, you will—and if it is uncontrolled, your network will collapse. Thus, while it is okay to hook up a few workstations over one of the LAN versions of ATM that are on the market, we do not recommend ripping out your ethernet and replacing it with ATM at this time—even if the particular manufacturer of ATM switching equipment you chose to buy does have congestion control. Such a system is proprietary and will undoubtedly be incompatible with whatever standard that does arise. And one will, we are willing to bet on that.

The reason why we believe ATM will one day have a viable set of congestion control features is that ATM as a WAN already is a sure thing. Gigabucks are being invested in it by every major telephone company and PTT in the world. It will be our WAN in just a few years. There is no doubt about that because it is already an established fact. Given that, it is obvious that there will be extreme pressure from LAN users to have a neat and clean interface to this WAN and what better way than by ATM itself?

What this means is that if the ATM Forum doesn't get their act together on these two issues, somebody else will. They can simply do what the ATM Forum did to the CCITT and form a concerned users group and take ATM away from them. However, we doubt it will come to that. Sooner or later, they will get together and come up with something that does work. There is too much money at stake.

9.9 The Role of ISDN and Cable TV in ATM.

We would like to end this white paper with some speculation. One of the recurring issues we mentioned is the problem of the "last copper mile" as the subscriber loop is called. It is the really big problem because there are so darn many of them. Literally billions of dollars will have to be spent to replace all that wire with something else. Until now, nobody was willing to spend that sort of money.

As noted, ISDN was an attempt to circumvent the issue by retaining that wire and converting it to a relatively low-bandwidth digital physical media. However, ISDN has had its day in the sun and is no longer being pursued by most telephone companies. That leaves us with the question of how we hook up our homes to the magic of the Information Superhighway. It remains unresolved. One proposal now being bantered about is to use a modified form of ISDN to give the home owner access to the Information Superhighway. Such an idea has technical merit. All one has to do is use the present physical layer of ISDN but instead of having D an B channels, use ATM instead. This will give you about 140 kilobytes per second, which is enough for most telephone calls and computer connections.

It does not, however, have anything like the bandwidth you need for your TV. Even a highly compressed video signal requires at least one megabit per second for each TV channel you are watching (and remember that most homes have several TV sets). That has opened an interesting opportunity for the cable TV companies, who have a separate problem to deal with. Back in the 1970s when cable TV became all the rage, the then start-up cable TV companies wired as many neighborhoods as fast as possible and admittedly did a poor job of it. Much of that cable has now deteriorated to the point that it must be replaced.

The opportunity the cable companies have is to replace that cable with fiber and literally co-opt the telephone company by offering ATM service. This will give you movies on demand, telephone service, computer connectivity and whatever else you can think of. ATM may well no longer be simply a communications system, but become our entertainment system as well. That mean big bucks are at stake.

In summary, it’s going to be an exciting next few years for ATM, for the telephone companies and for the cable TV companies. Keep an eye on the cable TV companies, and what they do or what is done to them. The action is going to be fast and furious, and whoever is standing on the top when it is all over will be a big time winner. You can bank on that. The question is who to bet on.