Fundamentals of SONET/SDH

P. Michael Henderson
michael.henderson@cox.net

September 18, 2001

When data is transmitted over a communications medium, a number of things must be provided on the link, including framing of the data, error checking, and the ability to manage the link (to name a few). For optical communications these functions have been standardized by the ANSI T1X1.5 committee as Synchronous Optical Networking (SONET) and by the ITU as Synchronous Digital Hierarchy (SDH).

This paper attempts to describe how SONET/SDH works, down to the octet level. While it is a tutorial on SONET/SDH, the reader is expected to be familiar with communications concepts such as parity, scrambling, BER, etc.; with legacy plesiochronous channels (like DS-1s and DS-3s); and with the principles of optical communications. If there are some concepts which you feel should be covered here, please send me an e-mail. If I don't cover it in the main body of the paper, I'll try to add an appendix to cover the areas which provide the most problems for readers.

SONET/SDH is not an easy subject. There are lots of things happening in a SONET/SDH system and things can get complex fairly quickly. I try in this paper to "layer" the description, starting with the simpler, overview subjects and then going to the more complex. My hope is that even if you don't read this paper all the way through you'll still get some useful information out of it.

Finally, although there are a lot of similarities between SONET and SDH, there are some significant differences, especially in terminology. In an attempt to avoid totally confusing the reader, I have focused primarily on SONET and SONET terminology. This does not mean that I think SONET is more important than SDH – it's just easier to explain things from a SONET point of view because SONET is a subset of SDH. Once you understand SONET, it's easier to understand SDH. If you try to explain it the other way around, you have to keep jumping back to SONET to explain why certain things exist in SDH. There are a number of things in SDH which don't make sense until you realize that SDH had to do it that way to maintain compatibility with SONET.

Appendix A has been added which explains things from an SDH perspective. It builds on the SONET knowledge you will gain from the main body of the paper so please do not attempt to read Appendix A until after you've finished reading the main body – if you're just learning SONET/SDH it'll probably get you totally lost.

Historical Background

Before I jump into SONET/SDH, let me cover a bit of telephone communications history to review how we got to optical communications and the problem that the designers and engineers were trying to solve.

For a great many years, telephone calls were handled in the analog domain. Long distance calls were routed over twisted pair, coaxial cable, or analog microwave between major switching offices. Any readers old enough to remember these long distance calls will recall the amount of noise and crosstalk in these calls – you always knew when someone was calling you long distance[1].

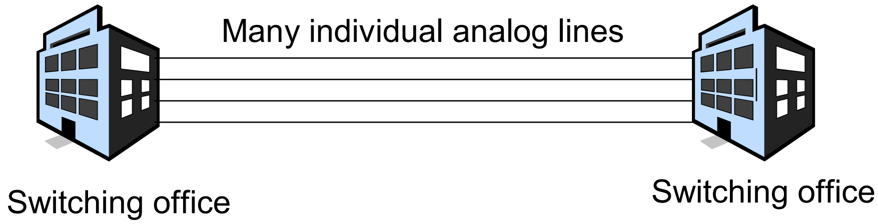

Figure 1: Individual analog lines between switching centers.

In 1962, AT&T began installing DS-1 T-carrier services between long distance switching centers. Basically, these were channel banks[2] which took 24 analog telephone circuits, converted them to digital and then transmitted them over copper to the other switching center, where they were converted back to analog. This worked very well – it reduced the number of copper circuits required between switching centers and improved the quality of the telephone calls (less noise and crosstalk).

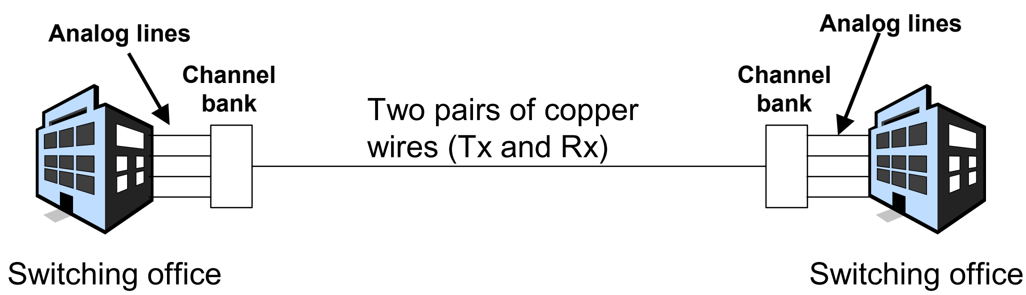

Figure 2: Use of digital communications between switching offices to replace individual analog lines or analog microwave. Note that the channel banks are inside the switching offices and the analog lines are connected to the telephone switch.

As the volume of long distance grew, the number of T-carrier circuits required between switching centers increased. Additionally, DS-1C and DS-2 signals began to be used to increase the capacity of a circuit. In the late 1970's optical communications became feasible, allowing higher speed communications, which meant that one circuit could carry many more telephone calls. For example, one of the first commercial fiber circuits was installed in Chicago in 1977 and operated at 45 Mbps (DS-3 rate).

[1] At least, that's what my grandfather tells me :-)
[2] The first channel banks were known as "D1" channel banks and coded the speech into 7 bit samples. Later channel banks coded speech into 8 bit samples. Perhaps the most common channel bank in use today is the "D4" channel bank, although there's a newer "D5" channel bank.

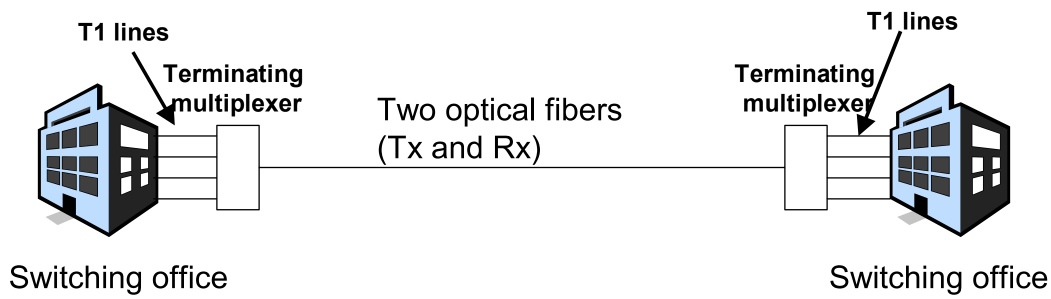

Figure 3: Early use of optical communications to replace DS-1 links between switching offices. The terminating multiplexers will be inside the switching offices. This point-to-point link is subject to failure – rings were eventually used to provide backup.

In those days, the telephone companies looked at optical communications as simply a replacement for the older wire or microwave communications they had been using for years. But then they encountered a practical problem. Vendors of optical communications equipment had used their own framing techniques on the optical fiber. Once you selected a vendor, you were stuck with that vendor for all the equipment in that optical network. Thus was born the concept of standards in optical communications.

It's extremely important to recognize that the first standards for optical communications were focused on handling voice circuits, and especially legacy plesiochronous channels like DS-1s and DS-3s. If you keep this fact in mind, many of the odd things about SONET and SDH will make more sense. At the time these standards were developed, the tremendous volumes of data traffic had not appeared and most people did not foresee it.

Introduction to SONET/SDH

Information is sent over an optical fiber by turning the light off and on in the fiber. For example, suppose that the presence of light indicates a "1" while the absence of light indicates a "0". Just knowing this much we can send and receive bits across an optical link[3].

But how do we extract the information from those bits? This is where SONET/SDH comes in. SONET/SDH defines the low level framing protocol used on these optical links. By "framing", we mean a block of bits (or octets) which have a structure, and which utilize some technique which allows us to find the boundaries of that frame structure. Parts of the block may be devoted to overhead for the network provider to use to manage the network. Other parts will be dedicated to carrying payload, or information we want to communicate.

For example, since I assumed the reader knows communications principles, I'm going to assume you already know something about high level data link control protocol (HDLC). HDLC is the protocol used on lower speed links, perhaps over a POTS modem link. HDLC has a framing character at the beginning and end of the frame, some control information, the payload, and a cyclic redundancy check (CRC) field which is used to check if any errors occurred in the transmission. See Figure 4. SONET/SDH is simply the HDLC for optical links.

[3] See Appendix B.

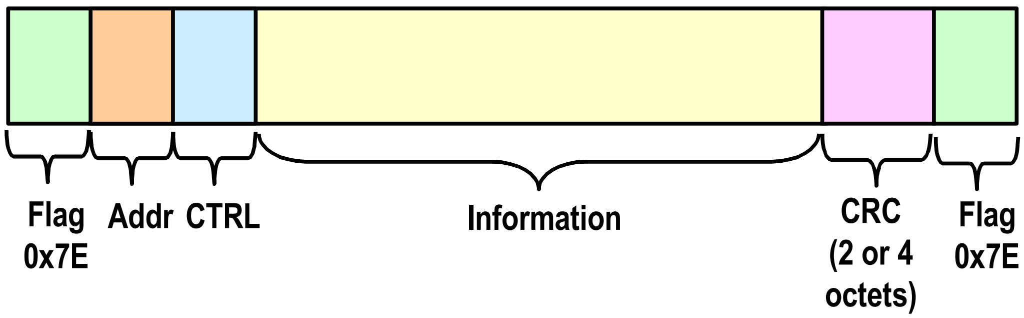

Figure 4: An HDLC frame.

HDLC uses the framing character twice, once at the beginning of the frame and once at the end of the frame. This is necessary because, within limits, an HDLC frame can be any size – so we have to indicate both the beginning and the end of the frame. But if an HDLC frame were always the same size, we'd only have to indicate the beginning. Once we found the beginning we'd know where the end was because we'd know the length. When we get to the SONET/SDH frame we'll see that these frames are a fixed size for each SONET/SDH rate. Because of this, SONET/SDH uses framing characters only at the beginning of the frame.

Note that the way we represent an HDLC frame is essentially one-dimensional – we just string the octets of the frame along from left to right. Later, we'll see that the SONET/SDH frame uses a two-dimensional representation.

Now, let us step back in history and see how SONET/SDH developed. SONET was developed in the United States through the ANSI T1X1.5 committee. ANSI work commenced in 1985 with the CCITT (now ITU) initiating a standardization effort in 1986. From the very beginning there was conflict between the US proposals and the ITU. The US wanted a data rate close to 50 Mbps in order to carry DS-1 (1.544 Mbps) and DS-3 (44.736 Mbps) signals. The Europeans needed a specification which would carry their E1 (2.048 Mbps), E3 (34.368 Mbps), and 139.264 Mbps signals efficiently. The Europeans rejected the 50 Mbps proposal as bandwidth wasteful and demanded a base signal rate close to 150 Mbps.

Eventually a compromise was reached which allowed the US data rates to be a subset of the ITU specification, known formally as Synchronous Digital Hierarchy (SDH).

As mentioned earlier, since SONET is a subset of SDH, I'm going to focus my discussion on the SONET frame. I'll attempt to point out where SDH differs and have provided an appendix which addresses SDH specifically. Additionally, ANSI has a report which outlines the major differences between SONET and SDH [T1Rpt36]. But for now, we'll talk SONET.

Before we jump into the SONET/SDH frame, there's some other terminology we need to talk about. The end-to-end connection through a SONET/SDH network is always called the "path." The connection between major nodes, such as between add/drop multiplexers is called a "line." And the link between an add/drop multiplexer and a regenerator, or between two regenerators, is called a "section." See Figure 5 which shows this graphically. Remember that back in the mid-'80s, the only kind of signal amplification available was electronic regeneration – optical amplifiers had not yet appeared.

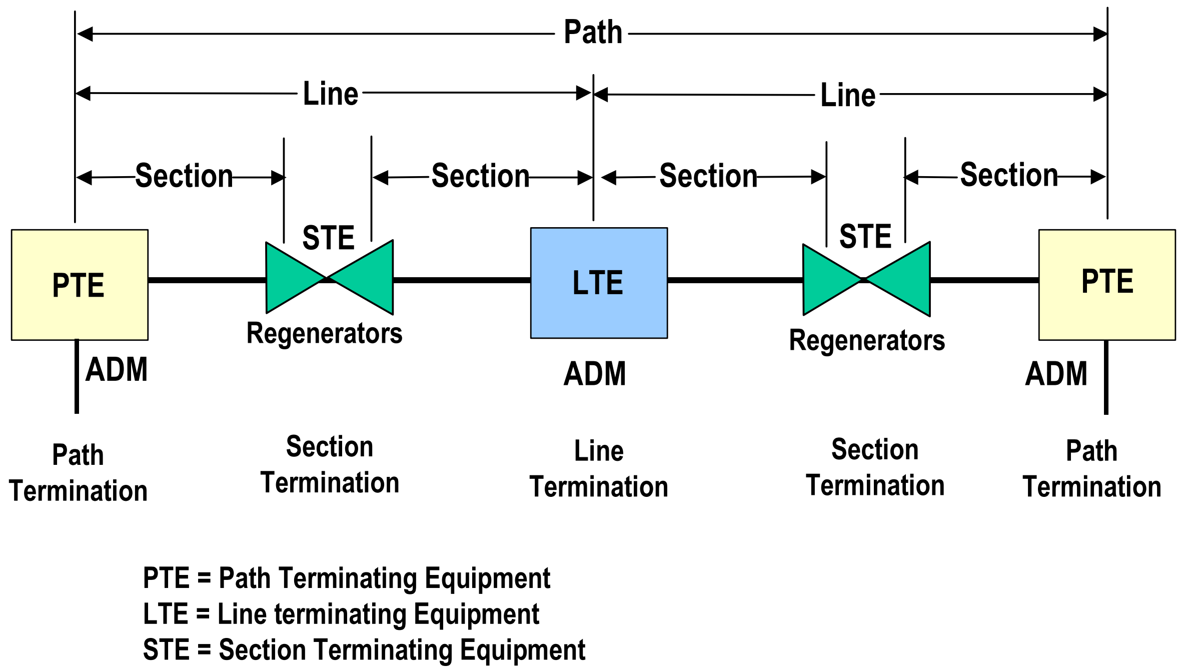

Figure 5: Terminology used in SONET/SDH.

The basic SONET frame is set up as shown below, in Figure 6, as 9 rows of 90 octets. It is transmitted from left to right and top to bottom. That is, the octet in the upper left corner is transmitted first followed by the second octet, first row, etc. When we get to octet 90, we come back and start with the first octet of the second row.

It's important to realize that this two-dimensional representation is just for convenience. The bits are simply transmitted one after another in a serial stream. We could also represent this SONET frame as a linear sequence of 810 octets. Every 90 octets we'd have three overhead octets. The two-dimensional representation is more convenient, allowing the whole frame to be shown on a page (without making each octet tiny).

There's nothing wrong with two-dimensional frames, or even three-dimensional frames – we just have to understand what's being represented and how to interpret it.

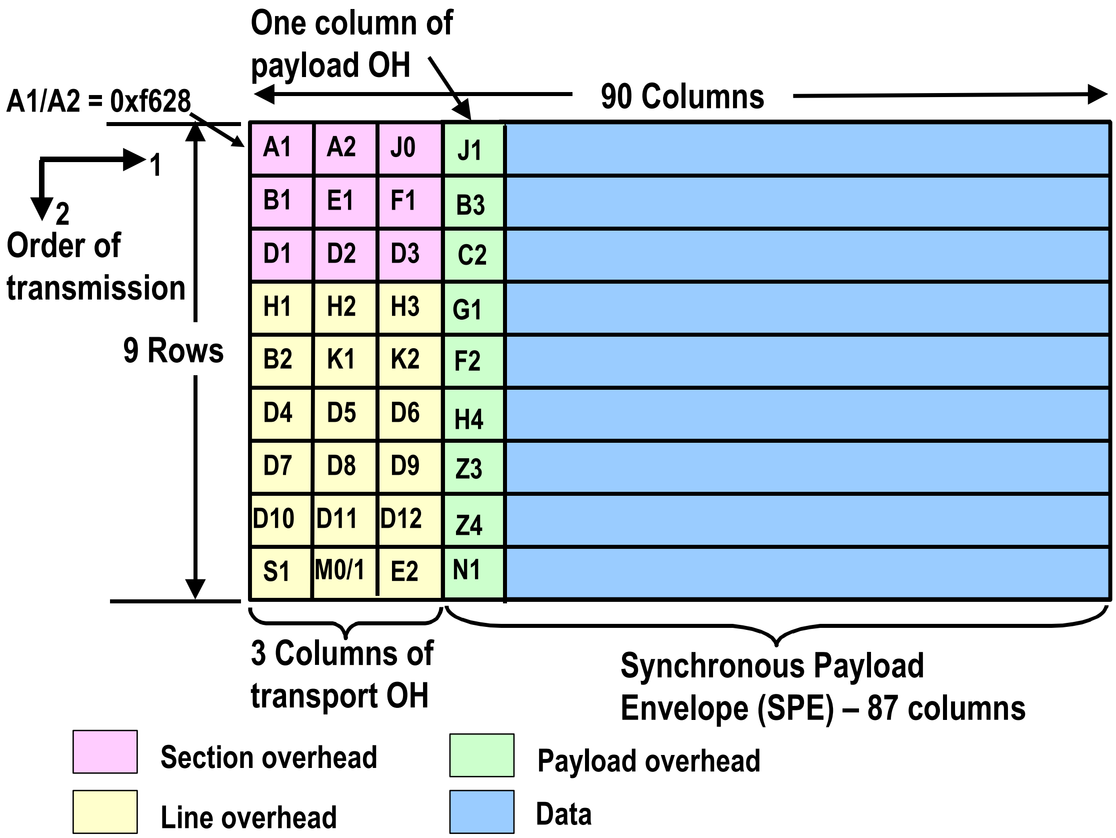

Figure 6: The basic SONET STS-1 frame.

Framing is accomplished by the first two octets, called the A1 and A2 octets. When the frame is transmitted, all octets except A1, A2, and J0 are scrambled to avoid the possibility that octets in the frame might duplicate the A1/A2 octets and cause an error in framing. The bit pattern in the A1/A2 octets is 1111 0110 0010 1000 (0xf628)[4]. The receiver searches for this pattern in multiple consecutive frames[5], allowing the receiver to gain bit and octet synchronization. Once bit synchronization is gained, everything is done, from there on, on octet boundaries – SONET/SDH is octet synchronous, not bit synchronous.

The first three columns of a SONET frame are called the Transport Overhead (TOH). The 87 columns following the TOH are called the Synchronous Payload Envelope (SPE). Within the SPE there is another column of overhead, called the Payload Overhead (POH), whose location varies because of timing differences between networks. We won't explore this aspect right now, but will later on. This leaves 86 columns by 9 rows for usable payload in this basic frame.

[4] I have not been able to find out why this particular pattern of bits (0xf628) was chosen for framing. The pattern was used for the E3 frame so the reasons for its choice probably go back to that standardization effort. If you know the answer to this, please let me know. It's a "unique" string and is DC balanced but many other patterns could meet those conditions – why was this particular pattern selected?

[5] The framing hardware searches for the boundary between A1 and A2 for octet synchronization. It must find the A1/A2 pattern for a certain number of consecutive frames before it leaves the seek state and enters the synchronized state.

Later we'll learn that the frame for the lowest SDH rate, STM-1, contains 270 columns by nine rows, and that it contains 9 columns of transport overhead. If nothing were done, this would leave 261 columns for payload, including payload overhead. However, one form of the SDH payload, known as the virtual container 3 (VC-3) has a structure very similar to the STS-1 87 columns by 9 rows of payload. When a VC-3 is combined with it's associated H1, H2, H3 pointers, it is known as an administrative unit 3 (AU-3). Of course, there are three AU-3s in the payload area of an STM-1.

Remember I mentioned earlier that the SONET designers were interested in carrying their legacy plesiochronous traffic. Because of this, they tied everything to the existing voice traffic. And in the digital telephone network, voice is digitized according to ITU specification G.711. That is, the granularity of the voice samples is one octet, and samples are taken 8,000 times per second, or every 125 μseconds. So this basic SONET frame is designed to carry voice conversation. This means that a SONET frame has a period of 125 μseconds.

This is an extremely important point. It turns out that *every* SONET frame repeats every 125 μseconds, no matter how fast the line speed gets. As the line rate goes up, the SONET frame gets bigger by some number of octets, just sufficient to keep the frame rate at 8,000 frames per second.

For this first level basic SONET frame, this gives a data rate of 51.84 Mbps (90 columns times 9 rows, times 8,000 times per second, times 8 bits per octet). This signal is known as a Synchronous Transport Signal - Level 1 (STS-1). Once the scrambler is applied to the signal, it is known as an Optical Carrier – Level 1 signal or OC-1. In this paper, I'm going to refer to STS-*N* rates. Just remember that these are the same as the OC-N rates.

Since there are 86 non-overhead columns of 9 rows, a SONET STS-1 frame has a usable payload rate of 49.536 Mbps, sufficient bandwidth to carry 774 simultaneous voice conversations[6]. This is in excess of the 672 simultaneous voice conversations carried in a DS-3, allowing one DS-3 to be easily mapped into a SONET STS-1 channel.

Lower rate plesiochronous digital hierarchy (PDH) traffic (DS-1, and E1) is encapsulated with additional framing octets, designed to allow the PDH traffic to be carried within a SONET/SDH channel. This is known as a virtual tributary (VT) in SONET and a virtual container (VC) in SDH. Multiple DS-1 circuits, for example, may be combined into a single SONET channel, up to 28 DS-1s in an STS-1. The techniques used to map plesiochronous traffic into SONET/SDH will be covered in much more detail later in this paper.

I mentioned earlier that the ITU established a base rate close to 150 Mbps for SDH. Specifically, the rate the ITU established is three times the SONET STS-1 rate (three times 51.84 Mbps or 155.52 Mbps) and is called the Synchronous Transport Module – Level 1 (STM-1) signal. SDH also uses a 9 row frame but an STM-1 signal has three times as many columns as the STS-1 signal (270 octets instead of 90 octets). This pattern repeats for higher level of SONET/SDH – an STS-12 signal has 9 rows but 1,080 columns (12 times 90 columns), etc.

The overhead grows in the same proportion. In an STS-1 signal we have three columns of transport overhead. An STS-3/STM-1 signal has 9 columns of transport overhead. An STS-12/STM-4 signal has 36 columns of transport overhead. And an STS-768/STM-256 signal has a whopping 2,304 columns of transport overhead.

---

[6] Later, we'll see that not all of the payload area of the STS-1 signal can be used to carry voice traffic. And if the voice traffic is mapped as DS-1s or a DS-3, it can only carry 28 DS-1s or one DS-3 for a total of 672 voice channels.

Not all levels of SONET/SDH signals are used in the network. SONET goes from STS-1 to STS-3 because the STS-3 rate matches the lowest level of SDH, the STM-1. After that, both rates go up by factors of four. There's nothing magic about the factor of four. Making changes to the network is expensive and difficult and network providers only do it when it provides a significant gain or advantage. The basic concept for data rates has been "four times the data rate for twice the cost." The most common rates for SONET and SDH are given in Table 1 below.

SONET name	SDH name	Line rate (Mbps)	Synchronous Payload Envelope rate (Mbps)	Transport Overhead rate[7] (Mbps)
STS-1	None	51.84	50.112	1.728
STS-3	STM-1	155.52	150.336	5.184
STS-12	STM-4	622.08	601.344	20.736
STS-48	STM-16	2,488.32	2,405.376	84.672
STS-192	STM-64	9,953.28	9,621.504	331.776
STS-768	STM-256	39,813.12	38,486.016	1,327.104

Table 1: SONET/SDH digital hierarchy.

SONET/SDH Interleaving

An STS-3 can be thought of as three STS-1 bit streams transmitted in the same channel so that the resulting channel rate is three times the rate of an STS-1. And when multiple streams of STS-1 are transmitted in the same channel, the data is octet multiplexed[8]. For example, an STS-3 signal will transmit octet A1 of stream 1, then octet A1 of stream 2, then octet A1 of stream 3, then octet A2 of stream 1, octet A2 of stream 2, etc. (see Figure 7). This multiplexing is carried out for all levels of SONET and SDH, including STS-192 and STS-768. Because of this, SONET/SDH maintains a frame time of 125 μs.

[7] Overhead associated with the transport overhead columns only. Excludes overhead contributed by the POH.
[8] SONET/SDH utilizes octet multiplexing in order to reduce delay. Octet multiplexing is used instead of bit multiplexing because, in SONET/SDH, everything is done in octets instead of bits. Octet multiplexing, therefore, causes the minimum delay (compare it to row multiplexing, for example).

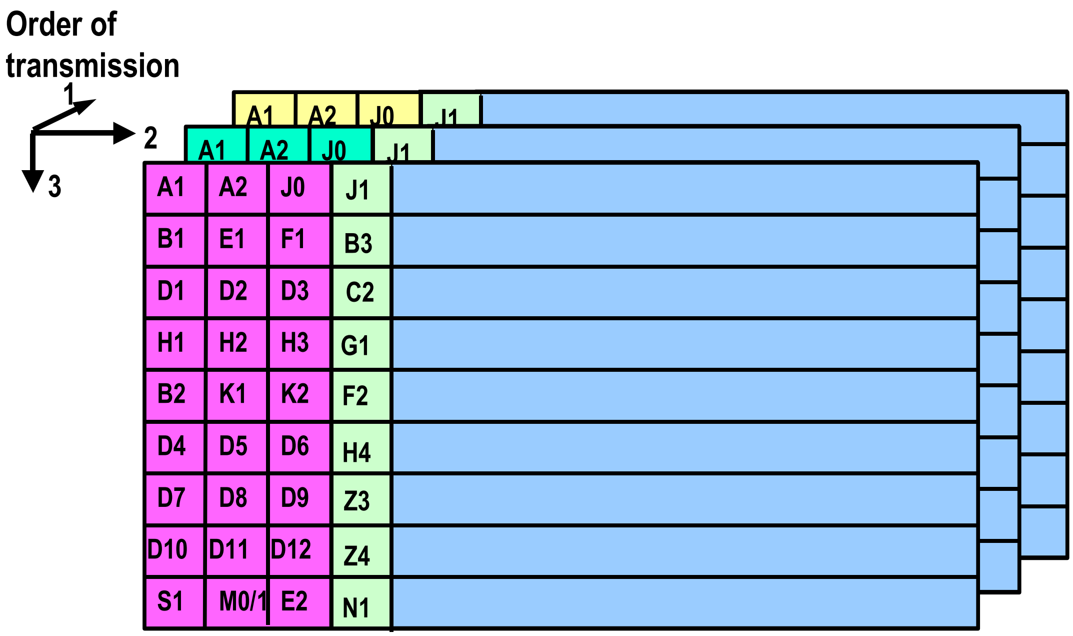

Figure 7: Interleaving of three SONET STS-1 frames into an STS-3 frame

Let's look at this a bit closer. Figure 8 shows an STS-3 frame created from the interleaving shown in Figure 7. It consists of 9 rows and 270 columns, of which 27 columns are overhead. Now, let's zoom in closer on this overhead. We see that the first three octets in the first row are all A1 octets. This is because we took three frames, like the one shown in Figure 6, and octet interleaved them. So we took the first octet from the first STS-1 frame (which is an A1 octet), then we took the first octet from the second STS-1 frame (which is an A1 octet), and then we took the first octet from the third STS-1 frame (which is also an A1 octet). Then we took the second octet from the first frame (which is an A2 octet), then the second octet from the second frame (which is anA2 octet), and finally the second octet from the third frame (which is also an A2 octet), etc.

A1	A1	A1	A2	A2	A2	J0	Z0	Z0	
B1	X	X	E1	X	X	F1	X	X	
D1	X	X	D2	X	X	D3	X	X	
H1	H1	H1	H2	H2	H2	H3	H3	H3	
B2	B2	B2	K1	X	X	K2	X	X	
D4	X	X	D5	X	X	D6	X	X	
D7	X	X	D8	X	X	D9	X	X	
D10	X	X	D11	X	X	D12	X	X	
S1	Z1	Z1	M0/1	Z2	M2	E2	X	X	

First STS-1 Second STS-1 Third STS-1

Figure 8: A SONET STS-3 frame showing how the STS-1s are interleaved.

Another way to look at this interleaving is that we take the first column from the first STS-1 frame and lay it in as the first column of the STS-3 frame. Then we take the first column from the second STS-1 frame and lay it besides the first column of the STS-3 frame which we're building. Then we take the first column from the third STS-1 frame and lay it besides the existing two columns of the STS-3 frame. Then take the second column from the first STS-1 frame and put it into the fourth column slot of the STS-3 frame, etc. While this creates the proper STS-3 frame, remember that the information is transmitted by row, octet-by-octet.

We haven't talked about what all the octets mean yet, but look at the STS-3 frame. You see that the first column of the STS-3 frame is the same as the first column of the STS-1 frame. And columns 4 and 7 of the STS-3 frame are the same as columns 2 and 3, respectively, of the first STS-1 frame. Since we interleaved these columns, this is to be expected. But look at columns 2, 3, 5, 6, 8, and 9 of the STS-3 frame. These are nothing like the columns 1, 2 and 3 of the second and third STS-1 frames (although we see some similarity in row 1 and row 4). What's going on?

It turns out that for many of the overhead octets, only one set is needed. This is not true for the overhead in row one or row four, however. These are required in every column.

Now, let's look at the payload. One type of payload is simply three of the individual STS-1 payloads. This is shown in Figure 9. The payloads would actually be inside of the frames but are shown separately to illustrate the fact that there are three of them.

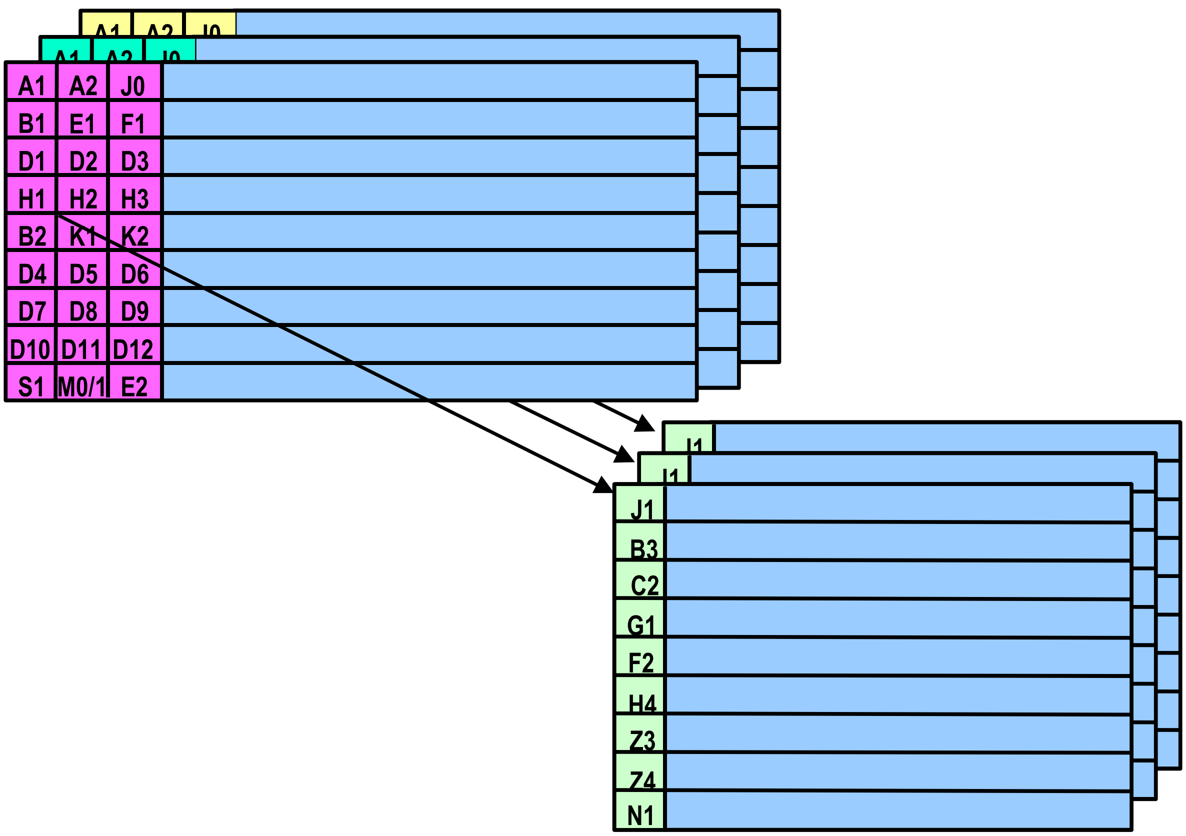

Figure 9: Three individual STS-1 payloads carried in an STS-3 signal.

We haven't said anything yet about how to interpret these interleaved SONET/SDH streams. Suppose you needed to transmit data that exceeded an STS-1 data rate (51.84 Mbps). How would you do it? It turns out that SONET/SDH provides the ability to concatenate a number of STS-1 payloads to create a higher speed payload. This type of data stream is indicated with a lower case "c" following the name (e.g., STS-3c or OC-3c). Concatenated channels have their payloads locked, i.e., when payload adjustments are made (due to differences in clocks), it is done simultaneously for the entire frame and is for N octets (this is explained later in the paper). Thus, an STS-3c is a single payload (SPE) data stream of 150.336 Mbps. An STS-48c is a single payload (SPE) data stream of 2.405 Gbps.

Figure 10 shows the payload for a concatenated data channel. Figure 11 shows the relation of that concatenated payload to the transport overhead. Note that there is only one payload and that it is pointed to by the H1, H2, H3 pointers in the first STS-1. I'll explain how the pointers work in more detail later.

Figure 10 is a bit tricky because it shows two "stuff[9]" columns after the POH. The note on the figure gives the equation for when these stuff columns are used: (N/3) – 1. For N = 3, this equation equals zero. So there are no stuff columns for an STS-3c, although there are stuff columns for all higher levels of concatenated SONET and SDH payloads.

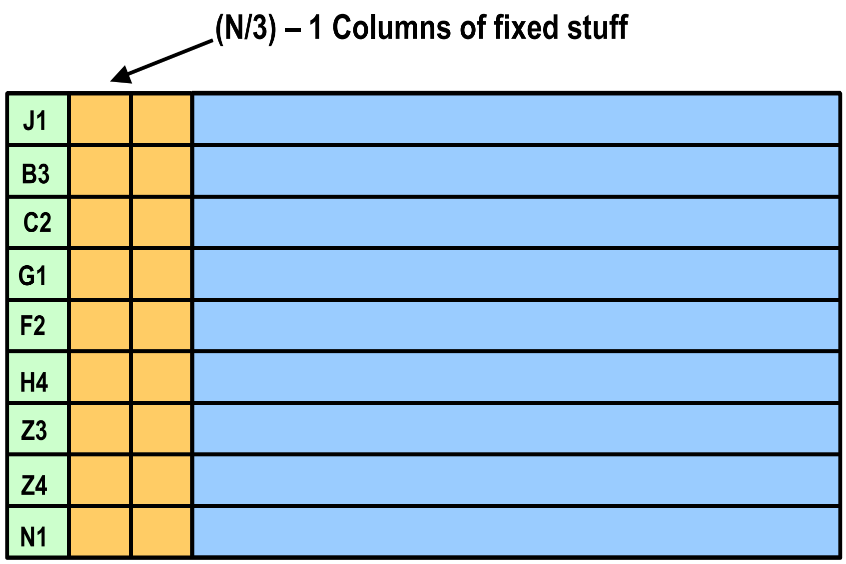

Figure 10: A concatenated payload. Note the stuff columns after the POH. These stuff columns only appear for SONET N>3.

[9] Stuff means "extra", or "unused." Stuff octets are, by definition, filler and not customer data.

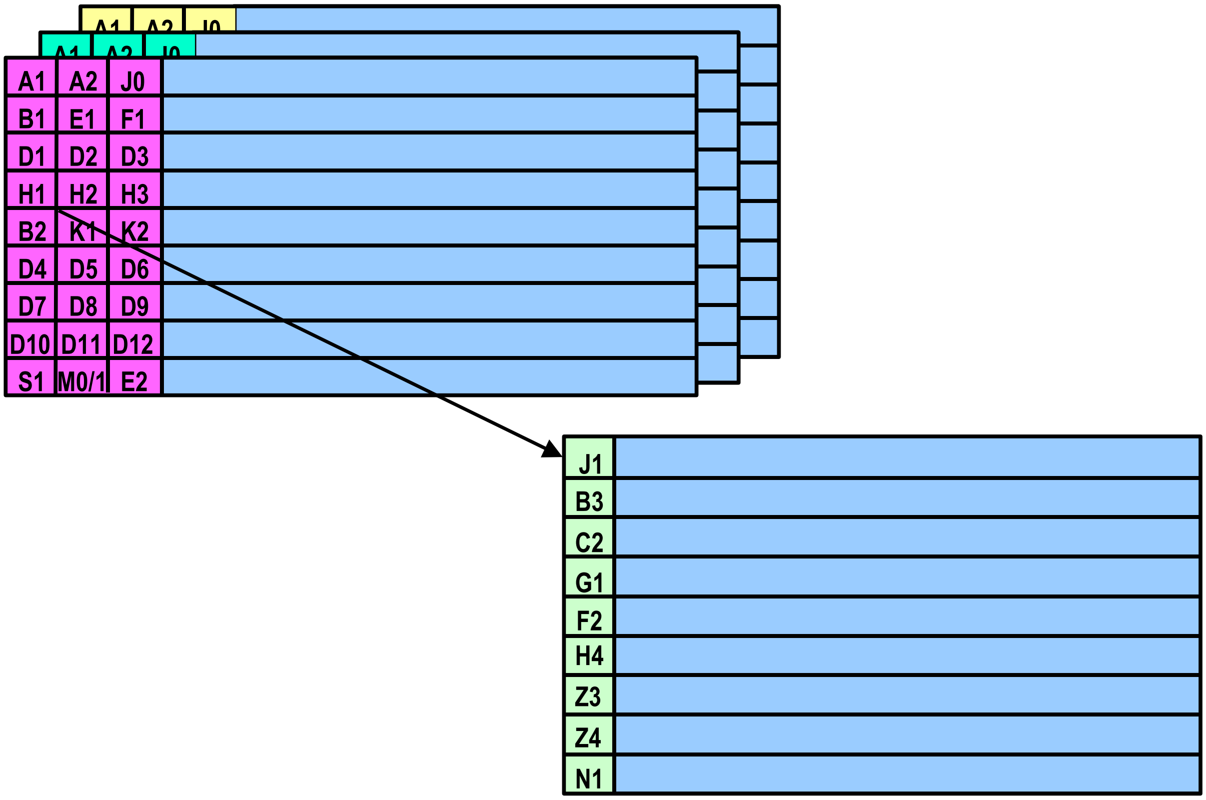

Figure 11: An STS-3c concatenated payload is pointed to by the H1, H2, H3 pointers in the first STS-1. Note that no stuff columns are shown because they are only required for N>3.

Detail of the Transport Overhead

Now, let us examine each of the overhead octets in the transport overhead (the first three columns of an STS-1 frame). We'll examine the octets in the payload overhead (POH) column later. Refer to Figure 12 as I describe each octet.

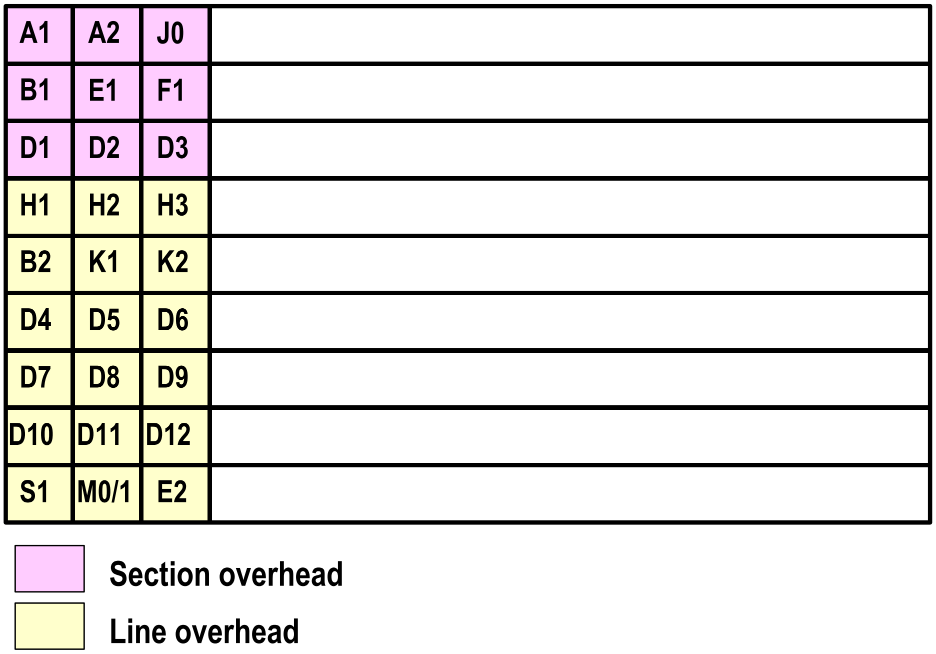

Figure 12: A SONET STS-1 frame showing detail of the transport overhead.

I've taken the descriptions of the overhead octets from the ANSI T1.105 and ITU G.707 documents. In each case, I attempted to use the latest version of the documents, including drafts which have not yet been approved[10]. So be careful – things could change!

There are several sets of overhead in SONET. I've tried to "distill" the meaning of the overhead but even distilling it as much as I could, the definitions are long, complex, and confusing. *To further reduce the definitions, I've put one sentence in italics to indicate the minimum definition*. On a first reading, just read the italicized sentence and skip the rest.

Framing octets (A1, A2) – *These octets allow the receiver to find the start of the SONET/SDH frame*. The A1 octet is 1111 0110 (hex 0xf6) while the A2 octet is 0010 1000 (hex 0x28). For SONET levels greater than STS-1 and less than or equal to STS-192, the A1 octet will be found in row one, columns 1 to N (where N is the SONET level). The A2 octet will be found in row one, columns N+1 to 2N. Framing for STS-768 uses the same A1, A2 values but limits the placement to columns 705 to 768 for A1 and columns 769 to 832 for A2[11]. SDH uses the same values for the framing octets.

Section trace (J0) – *This octet allows two connected sections to verify that the connection is still alive and still connected to the right terminations*. The J0 octet is used to repetitively transmit either a one octet fixed length string or a sixteen octet message so that a receiving terminal in a section can verify its continued connection to the intended transmitter. Only the first STS-1 carries the J0 value. This octet in

[10] Some of the text in the descriptions here is taken verbatim from the latest draft T1.105 standard.
[11] The reason for this limitation on the number of A1, A2 octets in STS-768 has to do with DC balance. The A1, A2 octets are DC balanced when taken together but are not DC balanced by themselves. If STS-768 followed the pattern of lower levels of SONET, there would be 768 A1s in a row, followed by 768 A2s (remember that the A1, A2 octets are not scrambled).

the other STS-1s in a higher level SONET signal are reserved for future standardization. These reserved octets are referred to as **Z0** octets. SDH uses this octet for the same purpose.

Parity (B1) – *The B1 octet is used by the receiver to estimate the bit error rate*. This octet is known as the Bit Interleaved Parity (BIP-8) octet. Since the octet has 8 bits, eight parities are computed, one for each bit of the octets of the frame. That is, you take the first bit of all of all of the octets in the frame and then set the first bit of the B1 octet so that the parity of these bits is even. Then you take the second bit of all of the octets in the frame, and set the second bit of the B1 octet so that it gives even parity, etc. The parity represented by this octet is the parity of the *previous* frame. It is used to estimate the bit error rate (BER) on the line. Note that the B1 parity is computed over all the octets in the frame, no matter how large the frame. Because of this, the B1 octet does not provide a good BER estimation for large frames (perhaps STS-48 and larger) under adverse error conditions. The B2 octet, described below, is computed over an STS-1 and provides better BER estimates. The B1 octet is only defined for the first STS-1 of an STS-*N* signal (there's only one B1 octet in a frame, while there are N B2 octets). SDH uses this octet for the same purpose.

Orderwire (E1) – *This octet is not important*. This octet was intended to be used for a voice channel between two technicians as they installed and tested an optical link. It is almost never used today. Technicians carry cellular phones and use these for communications when doing an installation. The E1 octet is only defined for the first STS-1 of an STS-*N* signal. SDH uses this octet for the same purpose.

Section user channel (F1) – *This octet is not important*. This octet is reserved for use by the network service provider. This octet is passed from Section to Section within a Line and is readable, writable, or both at each Section Terminating Equipment in that line. The use of this function is optional. The F1 octet is defined only for STS-1 number 1 of an STS-*N* signal. SDH uses this octet for the same purpose.

Section data communication channel (D1, D2, D3) – *These octets form a communication channel to send administrative messages*. These octets are allocated for Section data communication and should be considered one 192-kbit/s message-based channel for alarms, maintenance, control, monitor, administration, and other communication needs between Section Terminating Equipment. This channel is available for internally generated, externally generated, and manufacturer-specific messages. These octets are defined only for STS-1 number 1 of an STS-*N* signal. SDH uses these octets for the same purpose.

Pointers and pointer action (H1, H2, H3) – *These octets are very important and will be described in a later section*. These octets point to the payload (SPE), provide flags to indicate when the payload location changes, and provide a location for a data octet when a negative pointer adjustment is made. The operation of these pointers will be described in more detail in a later section. SDH handles pointers in the same way; however, the minimum SDH rate of STM-1 contains three H1 octets, three H2 octets, and three H3 octets.

Line parity (B2) – *The B2 octet is used by the receiver to estimate the bit error rate*. This octet operates in a fashion similar to the B1 octet, except that it excludes all of the section overhead octets and only applies to an STS-1. Since it only applies to an STS-1, there is a B2 octet in columns 1 to N of an STS-*N* signal. Note that this octet carries parity for the *previous* frame. SDH uses this octet for the same purpose.

Automatic Protection Switching (APS) channel (K1, K2) – *These octets will be described in a later section.* These octets are used for Automatic Protection Switching (APS) signaling between Line level entities. These octets are defined only for STS-1 number 1 of an STS-*N* signal. The operation and functionality of these octets will be described in more detail in a later section of this paper. SDH uses these octets for the same purpose.

Line data communications channel (D4 – D12) – *These octets form a communication channel to send administrative messages*. These octets are allocated for Line data communication and should be considered as one 576-kbit/s message-based channel for alarms, maintenance, control, monitor, administration, and other communication needs between Line-terminating entities. This is available for internally generated, externally generated, and manufacturer-specific messages. These octets are defined only for STS-1 number 1 of an STS-*N* signal. SDH uses this octet for the same purpose. SDH uses these octets for the same purpose but with additional codings.

Synchronization messaging (S1) – *This octet is not important to our discussion.* This octet is allocated for transporting synchronization status messages. S1 is defined only for STS-1 number 1 of an STS-*N* signal. Currently only bits 5-8 of S1 are used to transport synchronization status messages. These messages provide an indication of the quality level of the synchronization source of the SONET signal. SDH uses this octet for the same purpose.

STS-1 REI (M0) – *This octet sends the number of errors detected by the "B" octets back to the transmitter so it knows the line status as well as the receiver*. The octet in row 9, column N+1 (where N is the value of the STS-*N*) can have two meanings. When the signal is an STS-1, it has the meaning of "Remote Error Indicator" (REI). Currently only bits 5-8 of the M0 octet shall be used as a line REI function. These bits are used to convey the count of errors detected by the line BIP-8 (B2) octet back to the transmitter device. This count has 9 legal values, namely 0 to 8. The remaining possible 7 values shall be interpreted as zero errors. Bits 1-4 of M0 are reserved for future use. Since there is no rate in SDH equivalent to STS-1, SDH does not define an M0 value for this octet (but see M1, below).

STS-*N* REI (M1) – *This octet sends the number of errors detected by the "B" octets back to the transmitter so it knows the line status as well as the receiver.* In a SONET signal at rates from STS-3 to STS-192, one octet, the M1 octet, is allocated for a line REI function. The M1 octet is located in the third STS-1 in order of appearance in the octet interleaved STS-*N* frame. The entire M1 octet is used to convey the count of errors detected by the Line BIP-8 (B2) octet. This count has (8 times *N*) + 1 legal values, namely 0 to 8*N* errors. For rates below STS-48, the remaining possible 255-(8 times *N*) values are interpreted as zero errors. For the STS-48 and STS-192 rates, if the line BIP-8 detects greater than 255 errors, the line REI will relay a count of 255 errors. SDH uses this octet for the same purpose.

Orderwire (E2) – *This octet is not important*. This octet has the same purpose for line entities as the E1 octet has for section entities, and isn't used any more than the E1 octet. SDH uses this octet for the same purpose.

Payload Pointer Processing

So why do we have pointers in SONET/SDH? It all has to do with differences in clocks and accommodating those clocks. Let's look a bit closer at clocks, as used in communications systems.

Suppose we have some type of network box which takes traffic from one side of the box, perhaps processes the traffic, and then sends the traffic out the other side. See Figure 13. Here, we're only going to look at one direction of the line, assuming that bits are flowing into the box from the left and leaving the box on the right.

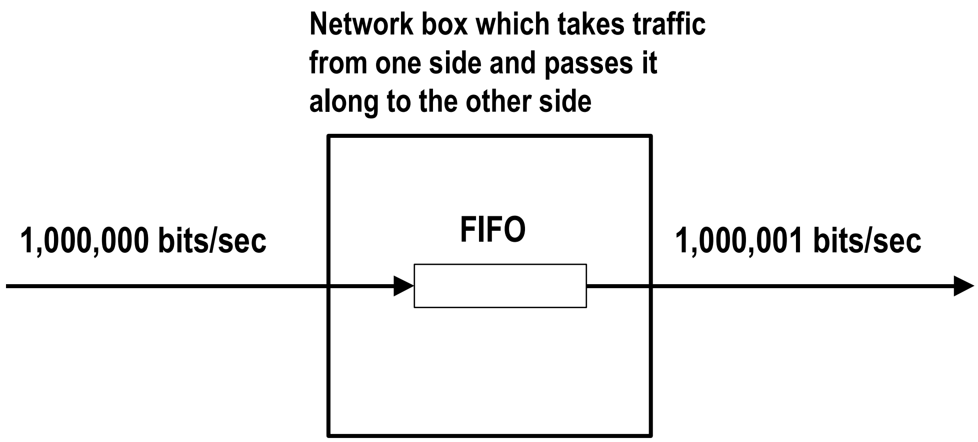

Figure 13: A hypothetical network box which takes traffic from the left side and sends it out the right side.

Note that the bit rate on the two sides is different. The bits are flowing in at one million bits per second but leaving at one million and one bits per second. Between the two lines is a FIFO or some kind of buffer which is used to minimize the problem we're going to discuss. Since the output line is faster, let's assume that we start the system off by allowing the line on the left to fill the FIFO before we start sending the bits out the right side.

Now, the right side line will send one bit more each second than the left side can put in the FIFO. Suppose we have an eight-bit FIFO. After eight seconds, the right side line will go to the FIFO to get a bit to transmit but there will not be any bits there. To fix this, the right side can send a "stuff " octet allowing the left side to fill the FIFO again. The important thing here is that the "stuff" octet be identifiable to the receiver connected to the right side line. That receiver can then throw the stuff octet away, leaving only valid data. Of course, throwing this stuff octet away may cause problems for that receiver, because it has no data to send for a short period of time. SONET/SDH actually uses this technique, and one to handle the case of a faster line feeding a slower line. The techniques for doing this are described in this section.

Technically, SONET/SDH are synchronous systems, meaning that all of the clocks are the same in the system (or so close as to be the same). However, even when all of the clocks are the same, there can be jitter which must be accommodated. And in reality, the clocks are not always the same in a SONET/SDH system.

Suppose that data is coming into a device slower (or faster) than it is being transmitted out the other side. While buffers can be used to mitigate the effect of different clocks, eventually something has to be done to adjust for the difference between the receive and transmit clocks. This is where the pointer and pointer action octets (H1, H2, H3) come in.

The H1, H2 octets are the pointer octets, comprising 16 bits. The first four bits are the New Data Flag (NDF) bits and are set to 0110 during normal operation. We'll see that one way to introduce a new pointer value is by setting the new data flag and including the new pointer. The next two bits have no meaning in SONET but are used in SDH[12].

[12] They were set to 10 to indicate the type of AU (AU-4, Au-4-Xc, AU-3, or TU-3). In the October 2000 revision of G.783 and G.806, these bits are no longer used to determine the type of AU.

The last 10 bits are the actual pointer and can vary from 0 to 782. A value of zero indicates that the payload (the SPE) starts at the first octet after the H3 octet. If the payload started at the second octet after the H3 octet, the pointer would have a value of one, etc. See Figure 14 which shows the layout of the H1, H2 pointers and Figure 15 which shows the location of the SPE for different values of the pointer. For the time being, ignore the "I" and "D" labels in Figure 14. The meaning of "I" and "D" will be explained a little later in this section.

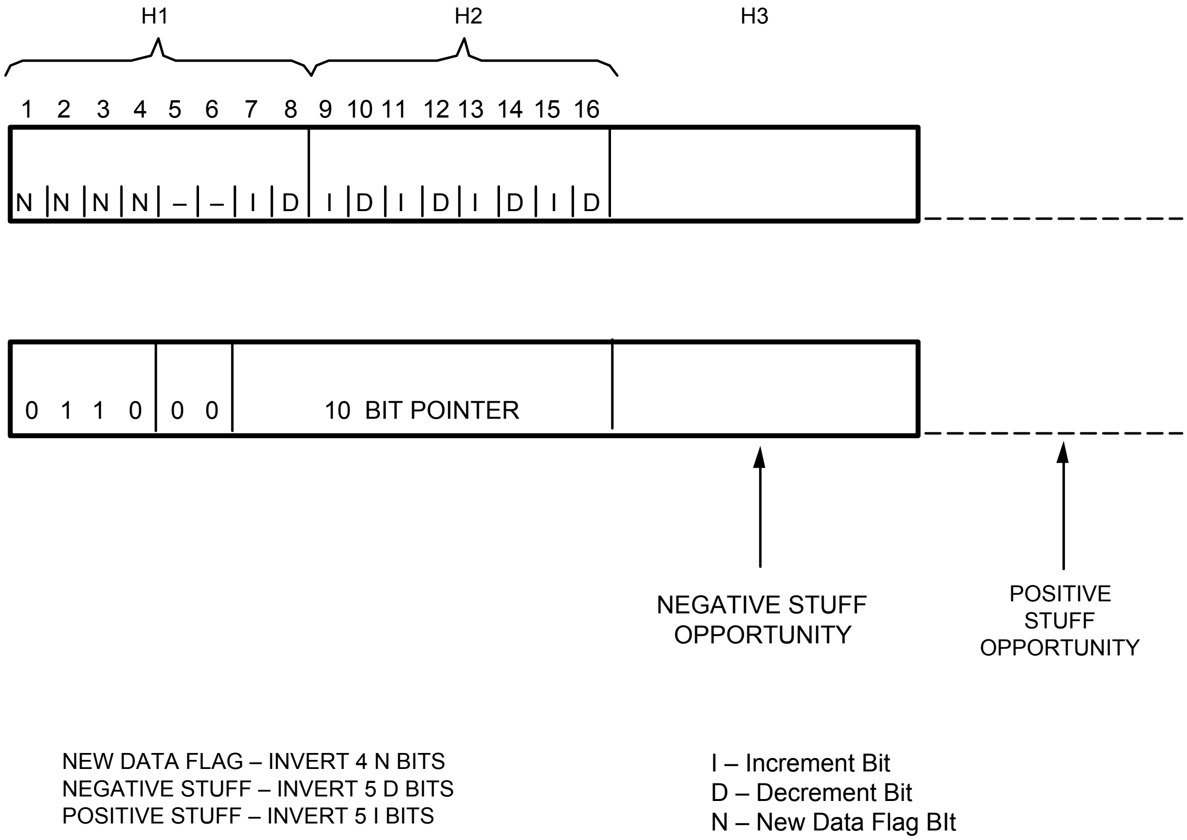

Figure 14: The usage of bits in the H1, H2 pointer octets. (source: Draft standard T1.105, Oct, 2000)

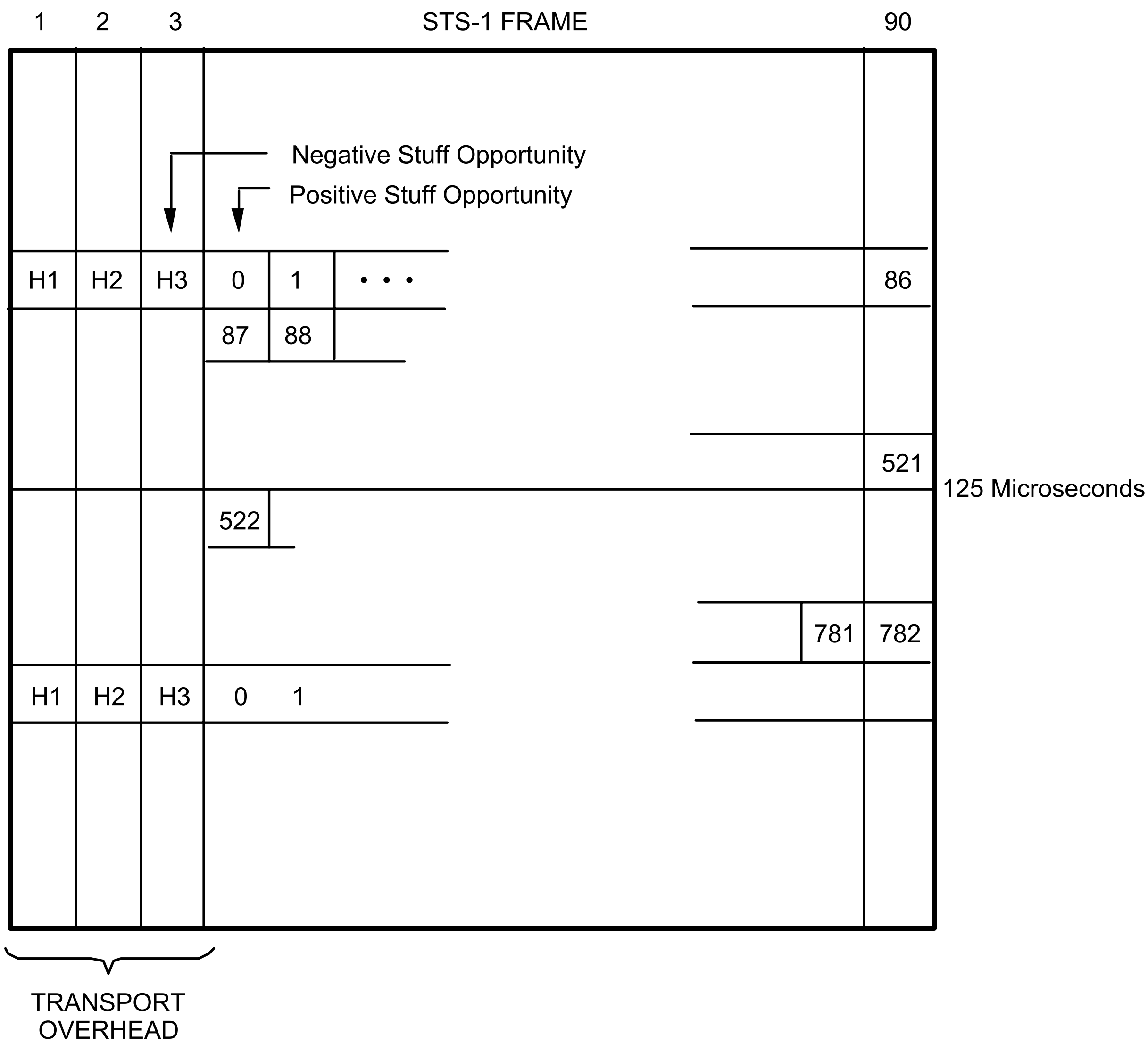

Figure 15: Pointer values for an STS-1 SPE. The numbers indicate the value that would be carried in the last 10 bits of the H1, H2 octets to point at that specific octet. For example, the H1, H2 pointer would contain zero to point to the octet after the H3 octet, and would contain 782 to point to the last octet in row 3 of the next frame. Another common pointer is to point to the first payload octet of the next frame, value 522. (source: Draft standard T1.105, Oct, 2000)

The H3 octet is used to carry a payload octet under certain conditions. Let's examine that now. Suppose the incoming clock was faster than the outgoing clock. Eventually, we'll accumulate an extra octet in our receive buffer, compared to what we can transmit. How are we going to "catch-up" this extra octet?

The way we do it is to put that extra octet in the H3 location. So when we transmit one SONET frame of 810 octets, we actually transmit 784 octets of payload (86 columns times 9 rows, plus one H3 octet), rather than 783 octets of payload. So now we've "caught-up" with that extra octet that accumulated on the receive side. See Figure 16.

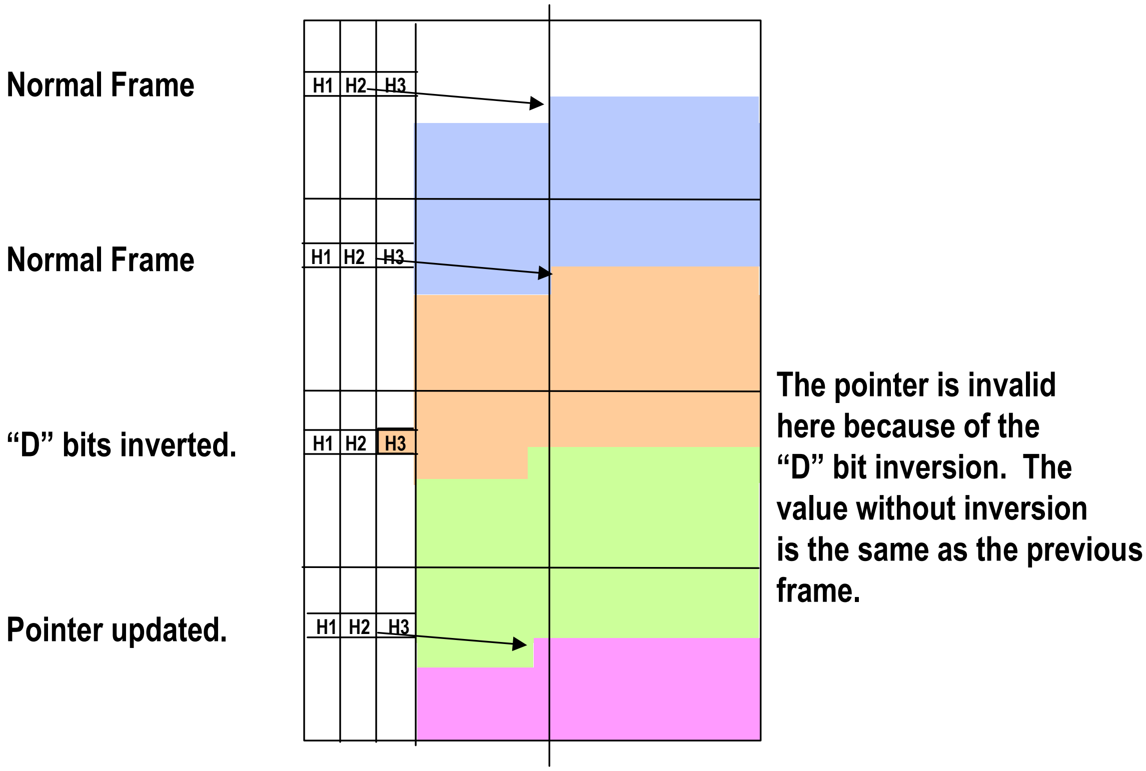

Figure 16: Negative STS-1 pointer adjustment. The "pointer" is the last ten bits of the H1, H2 octets. The meaning of the "D bits" is explained in the text, a few paragraphs down.

A similar problem occurs if the incoming clock is slow. Eventually, they'll be an octet "deficit" in the receive buffer which we'll have to accommodate. This is done by putting a "stuff" octet in the location after the H3 octet. So when we transmit a SONET frame of 810 octets, we transmit only 782 octets of payload, rather than the 783 octets of payload. See Figure 17 which show this stuff octet adjustment.

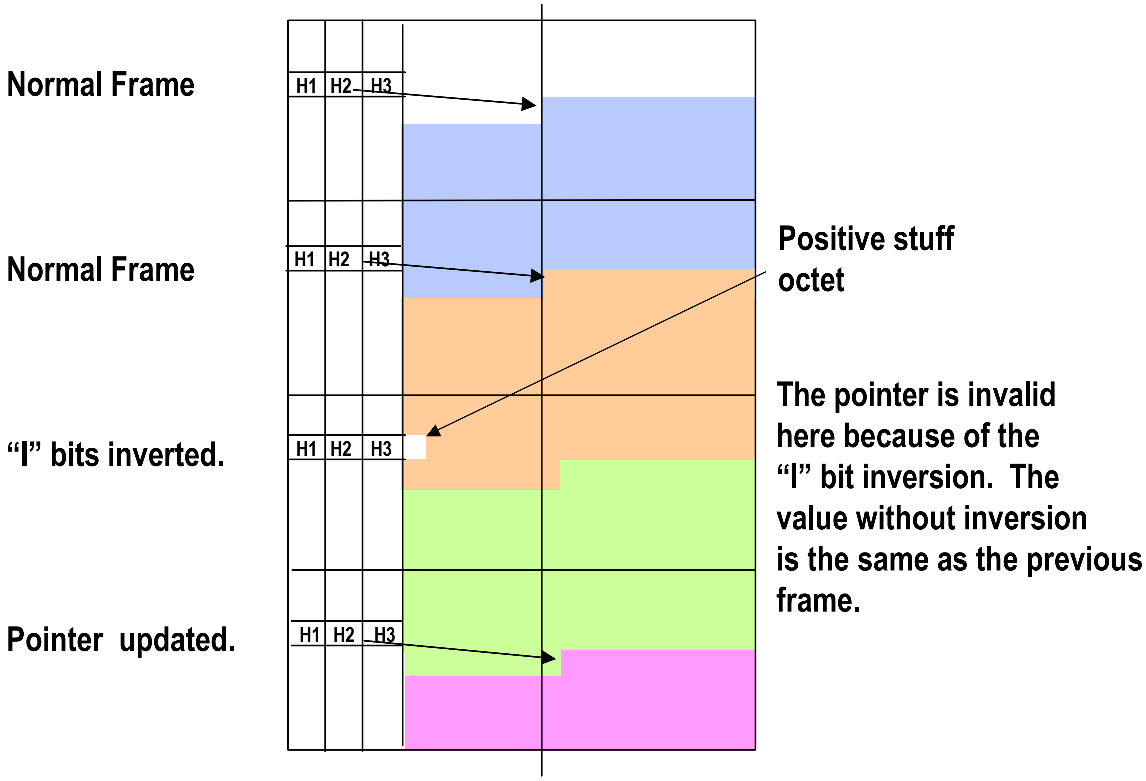

Figure 17: Positive STS-1 pointer adjustment. The "pointer" is the last ten bits of the H1, H2 octets. The meaning of the "I bits" is explained in the text, a few paragraphs down.

But this affects the pointers. The SPE is still just 783 octets long so the SPE actually moved "backwards" for a negative justification, or "forwards" for a positive justification, in the SONET frame. Since the H1, H2 pointers point to the start of the SPE, these pointers have to be adjusted to account for this movement. Let's now discuss how this occurs.

Note in Figure 14, the 10 bits of the pointer are labeled IDIDIDIDID. The "I" indicates "increment" and the "D" indicates decrement. When a new pointer value is to be introduced, the pointer (the last 10 bits) is modified to indicate whether the adjustment is a positive or negative adjustment. If the adjustment is to be positive, the "I" bits are then inverted; if the adjustment is to be negative, the "D" bits are inverted.

The SONET equipment has it's own register which it uses to track the location of the SPE, and it compares the value in the last 10 bits of H1, H2 to that register to detect whether the adjustment is positive or negative (which bits are inverted). Another way for the pointer to change is for the NDF to be set to indicate that a new pointer value is being introduced. We'll cover that in more detail later.

Under normal operation, when the pointer is to be changed, the pointer will be modified, with the I bits inverted if the pointer is to be incremented and with the D bits inverted if the pointer is to be decremented. The SONET equipment compares the pointer value with the value in its register to determine if the action is an increment or a decrement. Let's look at this a bit closer. We'll start with the positive pointer adjustment. It may help if you also look at Figure 17 during the discussion on positive justification and Figure 16 during the discussion on negative justification.

Positive Justification using Inverted "I" Bits

Frame status	New Data Flag				Unused		I	D	I	D	I	D	I	D	I	D
Normal frame	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	0
Invert "I" bits	0	1	1	0	X	X	1	0	1	1	0	1	0	1	0	0
New ptr value	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	1
New ptr value	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	1
Normal frame	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	1

Figure 18: The bit values for H1, H2 for a positive pointer adjustment using inverted "I" bits.

Let's assume that the pointer is initially pointing at the 127th octet in the payload (I chose this number because it looks about where the pointer is pointing in Figure 16 and Figure 17). Since the pointer value is one less (remember that the pointer value is zero to point at the first octet), the pointer will have a decimal value of 126, or 0x7e in hex. In terms of the ten bits of the pointer, it will be 00 0111 1110.

So follow along with me as I walk through a pointer change. The first row in Figure 18 indicates the bit values of H1, H2 for normal operational frames prior to the positive justification. On the frame where we're going to do the positive justification, the H1, H2 octets will have the values indicated in the second row of Figure 18. Since we're doing a positive justification, all of the "I" bits will be inverted. The SONET equipment has a register where it keeps the pointer value so it can easily compare the value in the last 10 bits of H1, H2 to its register to see what's changed (and that the "I" bits are inverted). It is in this SONET frame that the octet after the H3 octet is stuffed with a non-data octet – see Figure 17. When a positive adjustment is done the stuff octet after the H3 octet is ignored, so its content is meaningless – but it is usually all zeros.

Since the SONET equipment has been told that we're doing a positive justification, it can increment its register to point to the adjusted payload, even though the pointer in the H1, H2 octets has been "corrupted" by the "I" bit inversion.

On the next frame, the pointer has the new, incremented value. This is repeated for the next frame, and the fourth frame is considered a "normal" frame, available for another pointer adjustment.

Now, let's look at a negative adjustment. Assume the same starting conditions as above, payload at location 127, with a pointer value of 126.

Negative Justification using Inverted "D" Bits

Frame status	New Data Flag				Unused		I	D	I	D	I	D	I	D	I	D
Normal frame	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	0
Invert "D" bits	0	1	1	0	X	X	0	1	0	0	1	0	1	0	1	1
New ptr value	0	1	1	0	X	X	0	0	0	1	1	1	1	1	0	1
New ptr value	0	1	1	0	X	X	0	0	0	1	1	1	1	1	0	1
Normal frame	0	1	1	0	X	X	0	0	0	1	1	1	1	1	0	1

Figure 19: The bit values for H1, H2 for a negative pointer adjustment using inverted "D" bits.

The first row in Figure 19 indicates the bit values of H1, H2 for normal operational frames prior to the negative justification. On the frame where we're going to do the negative justification, the H1, H2 octets will have the values indicated in the second row of Figure 19. Since we're doing a negative justification, all of the "D" bits will be inverted. The SONET equipment has a register where it keeps the pointer value so it can easily compare the value in the last 10 bits of H1, H2 to its register to see what's changed (and that the "D" bits are inverted). It is in this SONET frame that the H3 octet is used to carry a data octet.

Since the SONET equipment has been told that we're doing a negative justification, it can decrement its register to point to the adjusted payload, even though the pointer in the H1, H2 octets has been "corrupted" by the "D" bit inversion.

On the next frame, the pointer has the new, decremented value. This is repeated for the next frame, and the fourth frame is considered a "normal" frame, available for another pointer adjustment.

When the H3 location is used for data, the actual payload octet placed in the H3 location is the octet that would normally have gone in the location just after the H3 octet. That is, the payload octets of row four slide left by one position. This leaves an "empty" octet in location 90 of row four which is filled by the octet that would have been in location 4 of row 5 (meaning that the payload of row four is 88 octets instead of 87 octets). So all of the payload octets from row four on are advanced one octet, including the next SPE.

In the next SONET frame, the 10 bits of pointer will point at the SPE (either one greater for a positive adjustment or one less for a negative adjustment). This adjustment can only occur every fourth SONET frame.

Pointer adjustments can also be signaled by the sender setting the new data flag (NDF). Let's examine only a positive pointer adjustment and assume that the pointer is being adjusted by one octet for some administrative reason. Follow along in Figure 20 as I discuss this.

One Octet Positive Adjustment using the NDF

Frame status	New Data Flag				Unused		I	D	I	D	I	D	I	D	I	D
Normal frame	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	0
NDF indicator	1	0	0	1	X	X	0	0	0	1	1	1	1	1	1	1
New ptr value	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	1
New ptr value	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	1
Normal frame	0	1	1	0	X	X	0	0	0	1	1	1	1	1	1	1

Figure 20: The bit values for H1, H2 for a one octet positive pointer adjustment using the new data flag.

The first line indicates a normal frame. In the next frame (the second row), the sending equipment has set the NDF and put a new pointer value in the last 10 bits of H1, H2. Note that the value is greater by one, indicating that the SPE has moved forward by one octet. Pointer values can change by much more than one octet with the use of the NDF. Suppose the sending equipment needed to make a large change in the location of the SPE for some reason. It could signal this change by the use of the NDF (this would likely cause errors to the customer, however).

Just like the pointer change indicated with the inverted "I" or "D" bits, pointer adjustments utilizing the NDF cannot occur more often than every fourth SONET frame.

One reason for the limit of every fourth SONET frame is that another way a new pointer value can be set is if a new value appears for three consecutive frames, even if the "I" or "D" bits or the NDF are not set. After three consecutive frames with the same new pointer value, the receiver will accept the new value.

Another reason for the limit of every fourth SONET frame is the positive pointer adjustment case where the pointer rolls over from 782 to zero. If you'll work out the situation in this case, you'll see that the pointer in the frame following the frame with the NDF inverted will point to the same SPE as the (implied) pointer of the frame with the inverted NDF. This means that the system has to ignore the pointer for one SONET frame and the first valid pointer will be the third SONET frame (the frame with the NDF inverted is one, the next frame is two, and the third frame has a valid pointer).

A similar special case occurs when the pointer rolls over backwards from zero to 782. To begin, the zero value pointer is pointing at the octet after the H3 octet. The next frame has the NDF set and the H3 octet is used for data. The implied pointer is pointing to the H3 octet. The beginning of the next SPE is the last octet in row 3 of the SONET frame after the frame with the NDF inverted. However, the pointer in that frame cannot point backwards in the frame, so the SONET equipment has to handle this situation as a special case. The pointer in that SONET frame points to the last octet in row 3 of the next (third) SONET frame. This can be confusing but if you draw pictures similar to Figure 16 and Figure 17 you'll see how it works.

This all works fine for an STS-1 which has only 783 octets in the payload. But what do we do when we have a concatenated payload, such as an STS-3c. Now we have 2,349 payload octets but the pointer is only 10 bits in length, giving the ability to point at only 1024 locations (0 to 1023). And much higher levels of concatenation are possible, such as an STS-192c. How do we get the pointers to work with these payloads? The answer is that we still use a value of 0 to 782 in the pointer but multiply it by N where N is the value in the STS-*N*. So for an STS-3c, we multiply the pointer value by 3 and adjustments take place three octets at a time. For STS-192c, we multiply the value in the pointer by 192 and adjustments take place 192 octets at a time.

Payload Overhead

Up to this point, we've talked about the overall structure of the SONET/SDH frame, and how we point to, and track, the first octet of the SPE, which is also the first octet of the payload overhead (POH), but we haven't talked about the POH itself and what the POH overhead octets mean. This is the subject of this section.

The POH is the first column of the synchronous payload envelope (SPE). See Figure 21. It consists of nine octets, the function of which will be explained below.

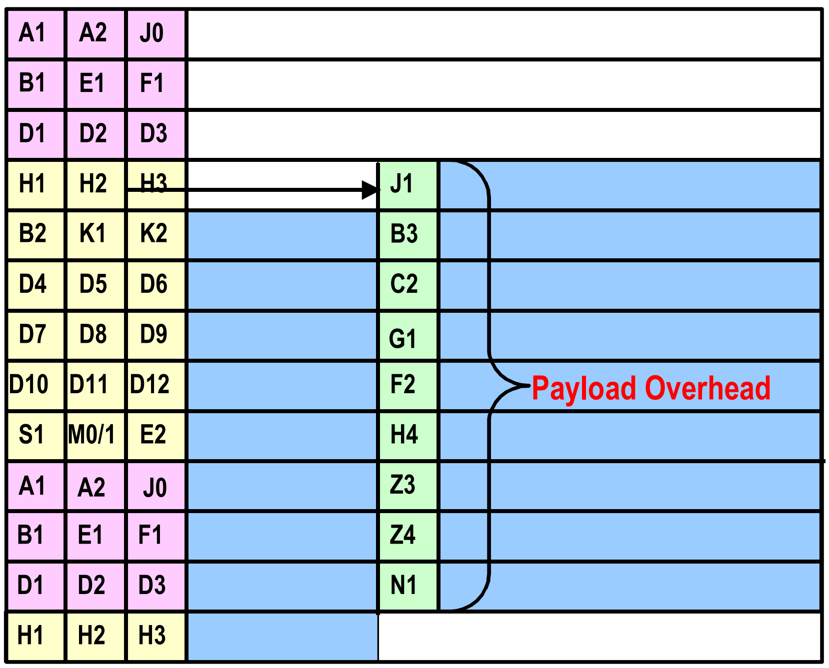

Figure 21: Payload overhead (POH) is the first column of the synchronous payload envelope (SPE).

Path trace (J1) – *This octet allows the two ends to verify that the connection is still alive and still connected to the right terminations. It's just like J0 but on a path basis.* This octet is used to transmit repetitively an STS Path Access Point Identifier so that a path receiving terminal can verify its continued connection to the intended transmitter. A 64-byte frame is used for the transmission of Path Access Point Identifiers. SDH uses this octet for the same purpose but uses a different message format, defined in G.707/G.831.

Path BIP-8 (B3) – *The B3 octet is used by the receiver to estimate the bit error rate.* This octet is used to provide parity over the payload. The path BIP-8 is calculated over all bits of the previous STS SPE before scrambling. Conceptually, this octet is similar to the B1 and B2 octets. SDH uses this octet for the same purpose but excludes the fixed stuff octets (which will be described when we talk about payload mappings) when calculating the parity. Because of this, the SONET equipment should use stuff octets which will produce the same parity as would be obtained by calculating the parity without the stuff octets[13].

STS path signal label (C2) – *This octet indicates the type of traffic carried in the payload. For our discussion, this octet may have a value of 0x02 to indicate floating VT mode, 0x04 for asynchronous mapping of a DS-3, 0x13 to indicate mapping of ATM, 0x16 for packet over SONET (POS), and 0x1b for generic framing procedure.* There are many other values – see T1.105 for the complete list. This octet is used to identify the construction and content of the STS-level SPE, and for STS path payload defect indication (PDI-P). PDI-P is an application specific code that indicates to downstream equipment that there is a defect in one or more directly mapped embedded payloads in the STS SPE. SDH does not define codes for PDI-P.

Path status (G1) – *This octet is mainly used to send status back to the transmitter about the path status*. This octet is used to convey back to an originating STS path terminating equipment (PTE) the path-terminating status and performance. This feature permits the status and performance of the complete duplex path to be monitored at either end or at any point along that path. As illustrated in Figure 22, Bits 1 through 4 convey the count of interleaved bit blocks that have been detected in error by the Path BIP-8 code (B3). This count has nine legal values, namely 0 to 8 errors. The remaining seven possible values represented by these four bits can only result from some condition unrelated to the forward path and shall be interpreted as zero errors. Bits 5, 6 and 7 provide codes to indicate both an old version (compliant with earlier versions of this standard) and an enhanced version of the STS Path remote defect indication (RDI-P). The enhanced version of RDI-P allows differentiation between payload, connectivity, and server defects. Bit 7 is set to the inverse of bit 6 to distinguish the enhanced version of RDI-P from the old version. Overhead bit codes for RDI-P defects are described in T1.105. Bit 8 is unassigned at this time. SDH uses this byte for the same purpose. STS Path REI coding is not suppressed when RDI-P is active. If RDI-P is triggered by a server defect, the REI will be undefined. If RDI-P is triggered by a connectivity defect or payload defect, the REI coding will reflect the incoming bit errors. The REI signal is undefined in SDH when RDI is active.

[13] Since there are an even number of stuff octets, one way to achieve this is to put the same value in every stuff octet.

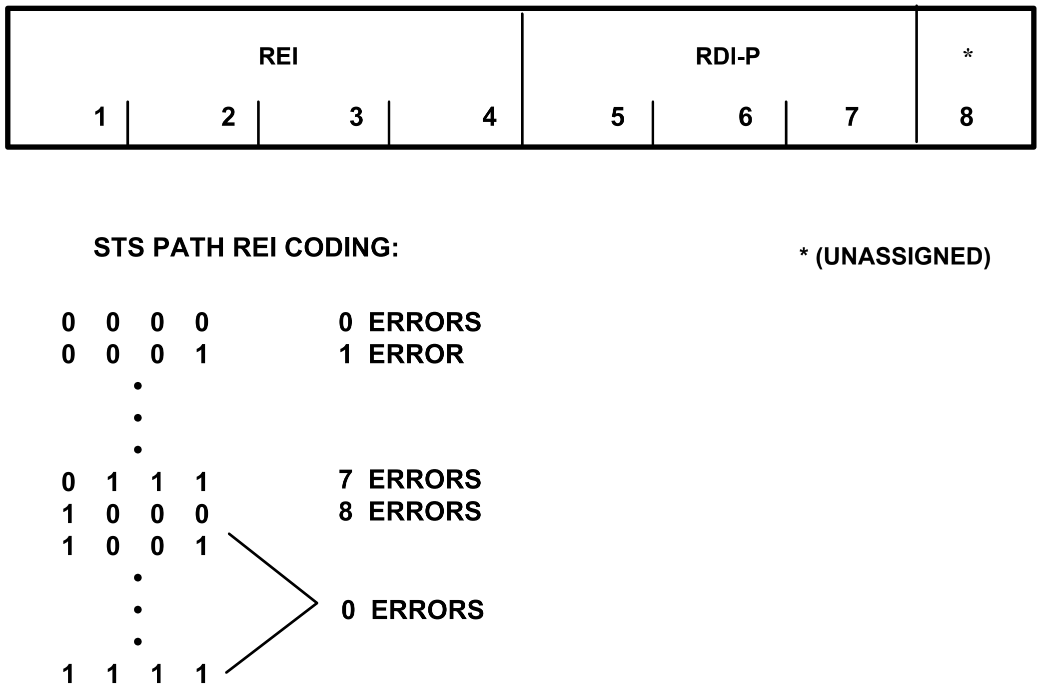

Figure 22: The Path status octet, showing the two fields, the REI and the RDI-P. (source: Draft standard T1.105, Oct, 2000)

Path user channel (F2) – *This octet is not important.* This octet is allocated for user communications, similar to the F1 octet in the transport overhead. SDH uses this octet for the same purpose.

Multiframe indicator (H4) – *The uses of this octet will be described in later sections.* This octet provides a generalized multiframe indicator for payloads. This indicator is used for two purposes. The first is for VT-structured payloads, which will be described in a later section. The second use is for the support of virtual concatenation of STS-1 SPEs. SDH uses this octet for the same purpose.

Growth octets (Z3, Z4) – *Ignore these two octets.* These two octets are reserved for future standardization. SDH defines Z3 as growth. SDH refers to Z4 as K3. Bits 1-4 of K3 are defined as a STS Path APS overhead channel. Bits 5-8 of K3 are reserved for growth.

Tandem connection maintenance/tandem connection data link (N1) – *Absolutely ignore this octet!* This octet is allocated to support Tandem Connection Maintenance and the Tandem Connection Data Link. Bits 1-4 of N1 are used to provide the Tandem Connection Incoming Error Count (IEC). In option 1, bits 5-8 of N1 are used to provide the Tandem Connection Data Link. In option 2, bits 5-8 of N1 are used to provide maintenance information including remote error indication, outgoing error indication, remote defect information, outgoing defect information, and TC access point identifier. For more information regarding Tandem Connection Maintenance, refer to T1.105.05. SDH uses this byte for a similar purpose.

The Tandem Connection Data Link of option 1 is an optional 32 Kb/s data channel available to applications or services (e.g., Tandem Connection, Security, etc.) that span more than one LTE-LTE (LTE = line terminating equipment) connection, but may be shorter than a PTE-PTE (PTE = path terminating equipment) connection. As a result, the B3 byte must be recalculated at each point where this channel is altered. Any errors received at this point must be included in the resultant newly calculated B3 byte. It should be noted that the Tandem Connection Data Link can exist independently of Tandem Connection Maintenance. The Tandem Connection Data Link will use the LAPD protocol for passing information. Due to their timing requirements, Tandem Connection Maintenance messages get precedence of use for the Tandem Connection Data Link, and may preempt other messages on that channel. Since Tandem Connection Terminating Equipments are not required to perform store-and-forward or Layer 2 termination functions on non-Tandem Connection Maintenance messages, some or all of the preempted messages may be lost and require retransmission. Note that the Tandem Connection Data Link is not supported in SDH, which uses only option 2.

Concatenated Payloads

Up to this point, we've been talking about STS-1 (OC-1) streams (at 51.84 Mbps) and a little about SDH STM-1 streams (at 155.52 Mbps). For simplicity, let's continue to concentrate on the SONET situation.

We also talked about interleaving multiple STS-1 streams into a higher stream on the line, e.g. three STS-1 streams into an STS-3. But with interleaved streams, each STS-1 stream maintains its own identity and is limited to 51.84 Mbps. Suppose, however, that you need a data stream faster than 51.84 Mbps – how can you get a faster rate from SONET?

The answer is with concatenated payloads. With concatenated payloads, multiple STS-1 payloads are joined together and treated as a single payload. For example, a common concatenated payload is to join three STS-1 payloads together. When this is done, the resulting stream is known as an STS-3c, with the lower case "c" indicating concatenated. An STS-3c would look like Figure 8 – it would consist of 270 columns with nine rows and 27 columns of transport overhead. An important difference between an interleaved STS-3 and a concatenated STS-3c is that there are three payload overhead (POH) columns in the interleaved structure, one for each STS-1, while there is only one POH in the concatenated structure. See Figure 10.

Thus, an STS-3c provides a payload that's a little more than the payload of three STS-1s (since only one POH is required the other two can be used to carry data). While not precisely accurate, and STS-3c is essentially the same as an SDH STM-1.

A reasonable question at this point is how does the system know that the data stream is concatenated? Look back at Figure 8, row 4, and you'll see three sets of H1, H2, H3 octets. The pointer is formed by taking the first H1 and the first H2 octets and interpreting the bits as described earlier, but multiplying the value in the last 10 bits by three (because it's an STS-3). The remaining H1 and H2 octets are used to indicate that the payload is concatenated.

The second H1 octet is paired with the second H2 octet to produce a 16-bit string similar to that produced by the first H1, H2 octets. Let's call this 16-bit word the "second pointer". The third H1 is paired in the same manner with the third H2 octet, and we'll call this 16-bit word the "third pointer". To indicate concatenation, the second and third pointers are set up as follows: the first four bits are set to 1001 (the new data flag value), the next two bits can be anything, and the last 10 bits are set to all ones. Since 10 ones form a value greater than 782, which would normally be invalid, it (together with the NDF in the first four bits) provides the indicator mechanism that the SPE which would normally be pointed to by this pointer is joined to the previous SPE. See Figure 11. This "joining" is a chain which is terminated only by a valid pointer value so any level of concatenation can be done, from 2 SPEs to 768 SPEs.

SDH indicates concatenation in the same fashion. However, if the payload of an STM-1 consists of three virtual container level 3s (VC-3), all three sets of the H1, H2, H3 pointers will contain valid pointers, each to a different VC-3. If the SDH STM-1 payload consist of a single payload, the pointers will be as described above for the STS-3c.

Mapping of SONET/SDH Payloads

One way to look at the payload of SONET/SDH is simply as a bit stream. We could look at an STS-1 stream as providing 86 columns of octets, by nine rows, 8,000 times per second, for a payload rate of 49.536 Mbps. An STS-3c could be thought of as 260 columns of octets (270 total columns minus 9 columns of transport overhead, minus one column of payload overhead), by nine rows, 8,000 times per second for a payload rate of 149.76 Mbps. Within that payload, we could put any traffic we wanted.

And actually we wouldn't be that far off if we looked at it that way. However, there are some additional complexities in the payload area. Remember back at the beginning of this paper I said that the designers of SONET/SDH were concerned about carrying their plesiochronous traffic. So we need to look at what facilities were built into the payload area to handle the different DS-N and E rates and how the differing clocks of the plesiochronous traffic are accommodated. Handling the clocks is especially important – SONET/SDH is specified as a synchronous system which means we should only have to accommodate clock jitter. But plesiochronous networks do not use the same clock – there are definitely differences between clocks on different plesiochronous circuits and also between the plesiochronous traffic and the SONET/SDH clocks.

So the first thing we're going to look at is how plesiochronous traffic is mapped into SONET. Later, we'll examine how non-plesiochronous traffic, such as asynchronous transfer mode (ATM), packet over SONET (POS), and generic framing procedure (GFP) is mapped.

Virtual Tributaries

This section and the following sections cause "glaze over" in all but the most ardent SONET/SDH people. If you're just looking for an introduction, you can stop reading here.

The specific traffic which the designers were interested in carrying is shown in Table 2. Although SONET was designed in the American National Standards Institute (ANSI), note the specification of a European rate, E1. This was done to reduce problems of cross border traffic. Let's say that a company in Europe wanted to provide a high-speed connection to an office in the US. They might purchase an E1 leased line (2.048 Mbps). But when the circuit got to the US, it would either have to be reduced to DS-1 (1.544 Mbps) or two DS-1's would have to be used and the traffic divided between them. Supporting E1 rates in SONET allowed E1 rates to be delivered in the US. And at the time SONET was designed, higher rates were too expensive to use for this type of application.

Type of digital circuit	Bit rate (Mbps)
DS-1 (T1)	1.544
E1	2.048
DS-1C	3.152
DS-2	6.312
DS-3 (T3)	44.736

Table 2: Plesiochronous traffic mapped into a SONET STS-1 frame.

Unfortunately, this area of mapping plesiochronous traffic into SONET is not simple. As I did earlier, I'm going to approach this in "onion" fashion, taking one layer at a time. Hopefully, the reader who decides not to finish reading this section will get some value by reading a portion of it.

To begin, the payload area of an STS-1 SONET frame consists of 87 columns by 9 rows. See Figure 23.

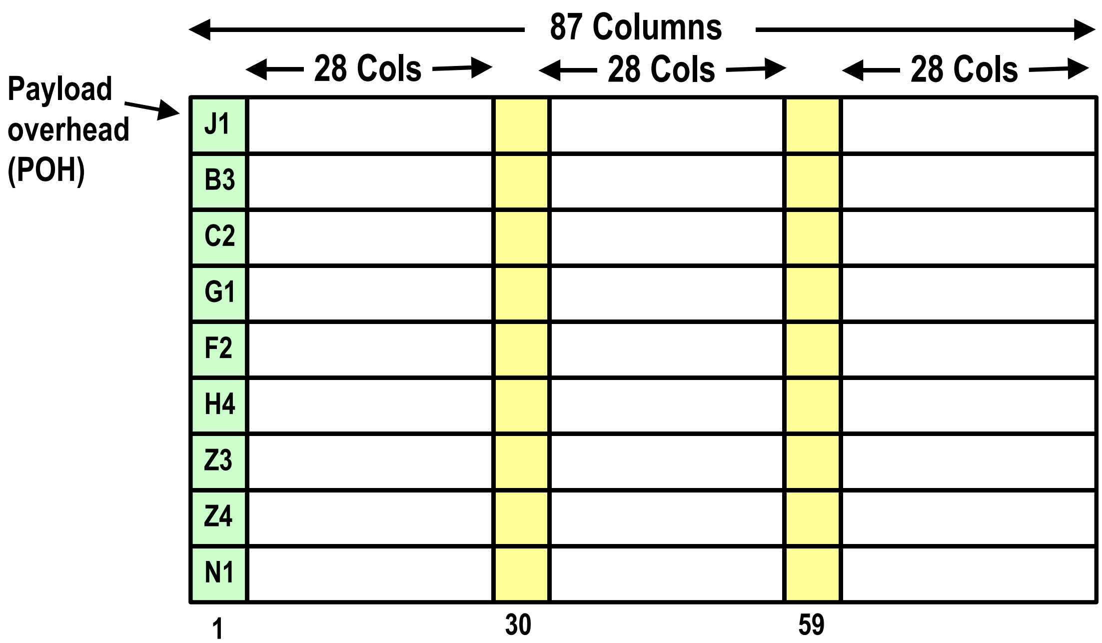

Figure 23: The payload (SPE) of an STS-1 SONET frame.

One column is taken by the payload overhead (POH) leaving 86 columns. Next, we break the 86 columns into seven groups of 12 columns. Now seven groups of 12 columns is only 84 columns, leaving two extra columns. These two columns are columns 30 and 59[14], where the POH is counted as column 1. All mappings of payloads into an STS-1 frame have these two columns "blocked out" meaning that the real payload of an STS-1 SPE is really only 84 columns by 9 rows, by 8 bits, eight thousand times per second or 48.384 Mbps.

Each of these seven groups is called a Virtual Tributary Group (VTG). The seven VTGs are interleaved into the 84 columns in the same manner as was discussed earlier for interleaving STS-1s into higher levels of SONET, e.g., into an STS-3. That is, the first column of the first VTG goes into the column after the POH. Then the first column of the second VTG, then the first column of the third VTG, etc.

[14] This breaks the SPE into three equal size groups of 28 columns of payload.

After you finish with first column of all seven VTG, you do the same thing with the second columns of all the VTGs, etc. This interleaves the VTGs. See Figure 24.

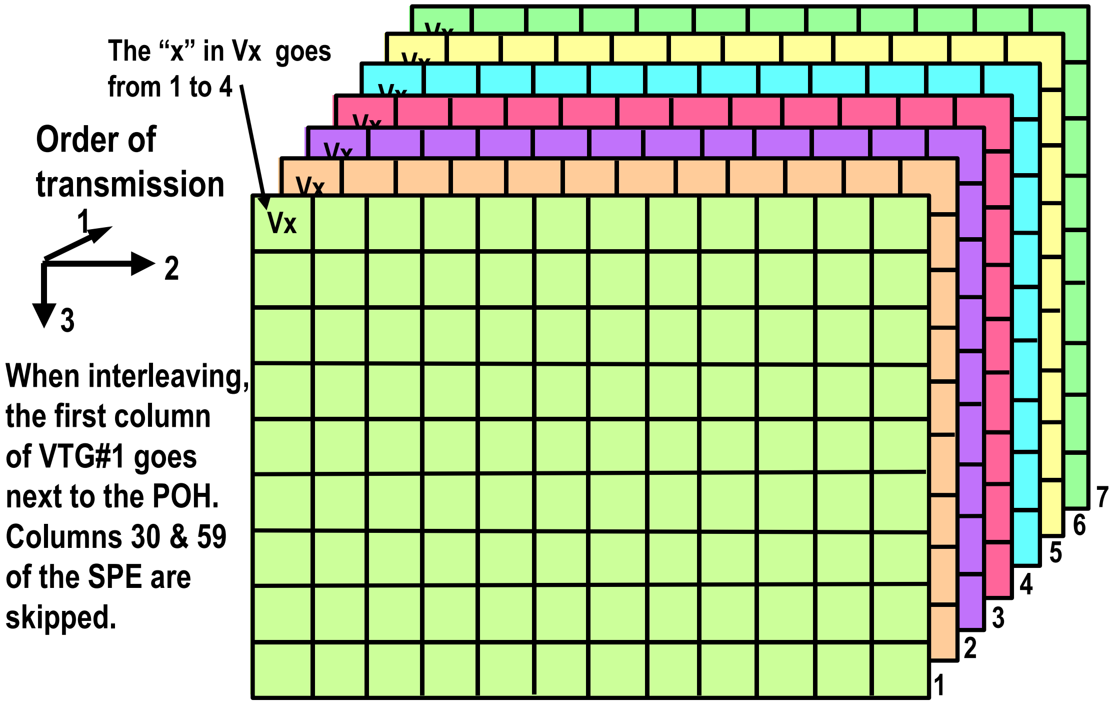

Figure 24: The 84 usable columns of the STS-1 SPE are divided into seven groups of 12 columns. Each group of twelve columns is known as a virtual tributary group (VTG). The VTGs are interleaved into the SPE in the same fashion as STS-1 are interleaved into an STS-N.

Now let's look closer at one of the VTGs. It consists of 12 columns, equaling a gross bit rate of 6.912 Mbps. Remember that we want to carry a number of different plesiochronous rates, from 1.544 Mbps to 6.312 Mbps. The gross bit rate is good for the DS-2 (at 6.312 Mbps) but it would be wasteful for the lower rates.

We can accommodate all of the rates specified in Table 2, however, simply be subdividing the 12 columns. For example, three columns, which are called a VT-1.5, give a gross bit rate of 1.728 Mbps. Four columns, which are called a VT-2, give a gross bit rate of 2.304 Mbps. Six columns, called a VT-3, give a gross bit rate of 3.456 Mbps. And twelve columns, called a VT-6, give a gross bit rate of 6.912 Mbps. So a DS-1 (T1) fits into three columns (a VT-1.5), an E1 fits into four columns (a VT-2), a DS-1C fits into six columns (a VT-3), and a DS-2 fits into 12 columns (a VT-6). One restriction is that a VTG can only contain one type of mapping, i.e., four VT-1.5s for four DS-1s, three VT-2s for three E-1s, two VT-3s for two DS-1Cs, or one VT-6 for one DS-2. An SPE payload can contain different VTGs with different types of VTs but within a VTG you can only have one type of traffic.

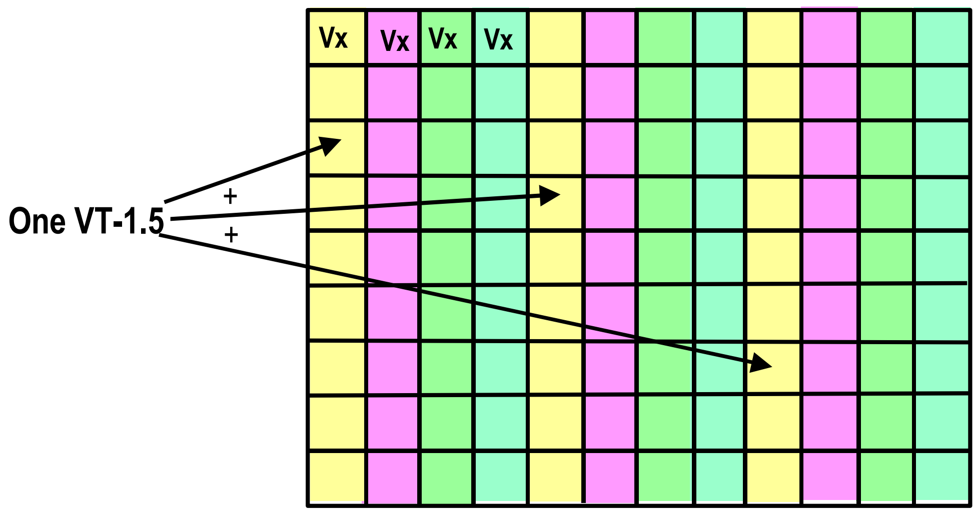

Figure 25: A VTG with four VT-1.5s interleaved within it. Each color represents one VT-1.5.

Next, we're going to look at the overhead octets which are required to make all this work. I'm going to only describe the mapping of DS-1 (T1) circuits. The mappings of the other rates is similar – if you understand the DS-1 case, you'll be able to read the specification to learn the specifics of how the other rates are handled.

To begin, the C2 octet in the POH signals that the payload contents are virtual tributaries (VT) by containing the hex code 0x02 (0000 0010). Once this is known the SONET equipment knows the location of each of the VTGs because they are in fixed positions in the payload.

But we immediately run into a problem. The three columns used to transport a DS-1 have only 27 octets per frame (which occurs 8,000 times per second). A DS-1 has 24 octets per frame (which occurs 8,000 times per second) plus an extra framing bit. If we have to use more than three of our VT octets for overhead, we can't carry the DS-1. This is a problem because we have to implement a pointer system, similar to the H1, H2, H3 octets used in the SONET transport overhead, to handle the differences in clocks between the plesiochronous traffic and the SONET network.

To get around this problem, the designers created the concept of a "superframe." A superframe is simply four VT frames. If we take one octet from each of the four frames, we'll have enough octets to implement a pointer like the H1, H2, H3 pointer system. And, it turns out, that's exactly what we do.

But how are we going to indicate the superframe? We use the last two bits in the H4 octet in the POH. These two bits count from 00 to 11 and then roll over to 00 again (00, 01,10, 11, 00). There is one special thing about the count in the H4 octet. The value in the H4 octet indicates the superframe number for the *next* payload, not for the payload associated with this H4. So an H4 with the value 00 in the last two bits indicates that the first frame of a superframe will occur in the next SONET payload.

So now that we can identify the frames of the superframe, we take the first octet in the VTG (which is also the first octet of the first VT in the VTG) and use it for overhead. Since we have a superframe of four frames, we have four overhead octets, known as V1, V2, V3, and V4. The SONET equipment takes the V1 and V2 octets and creates a 16 bit word. This word is interpreted in a very similar fashion to the way the word created by combining the H1, H2 octets in the transport overhead. See Figure 26.

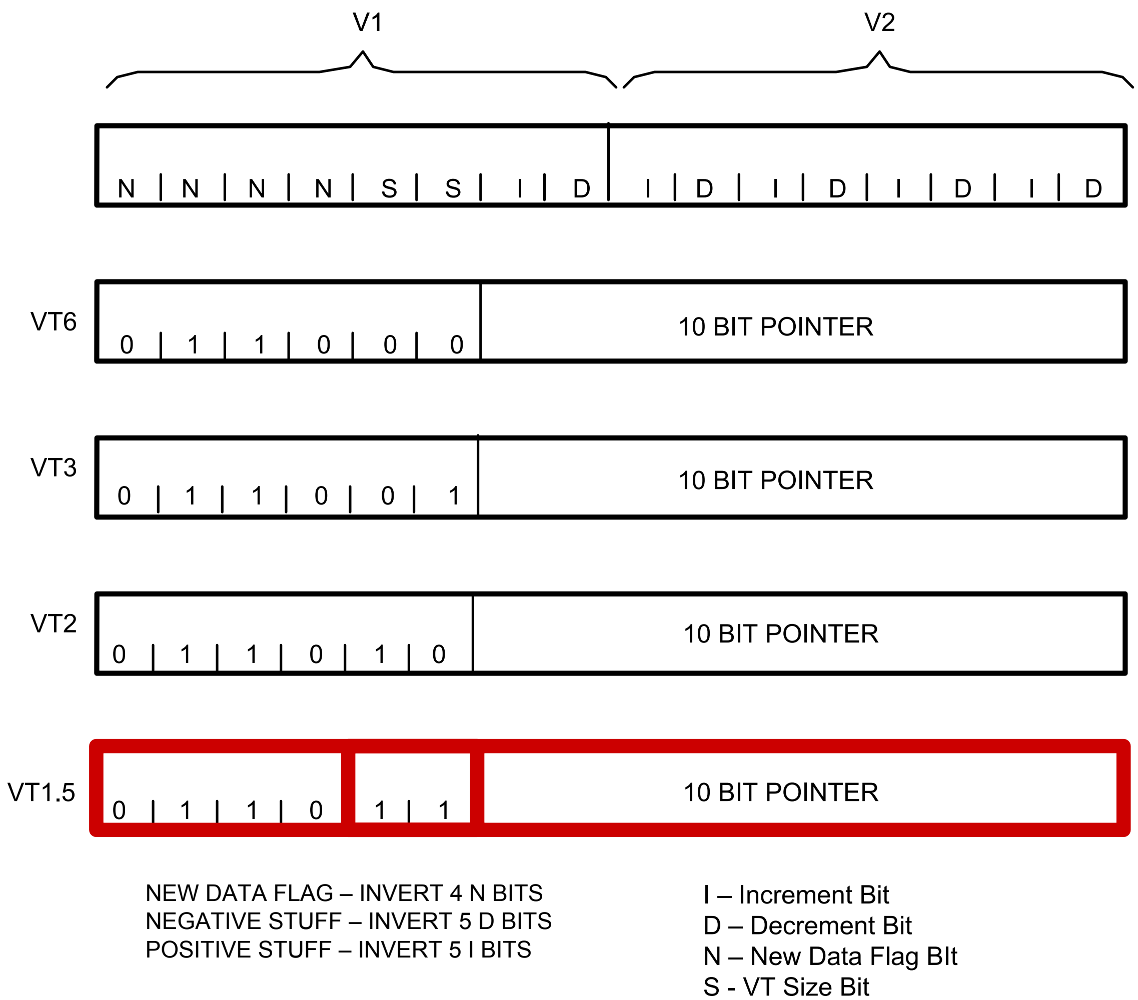

Figure 26: The bit definition of the V1, V2 octets in a virtual tributary. The VT-1.5 is highlighted since that's the one we're going to discuss. (source: Draft standard T1.105, Oct, 2000)

Note that the layout of the 16-bit word is the same as the H1, H2 octets, except that bits 5 and 6 are now used to indicate the type of traffic within the VTG. Each virtual tributary has the first octet taken as overhead (the Vx octets). Since the first VT will occupy the first column of the VTG, and since the SONET equipment knows how to find the VTGs, it can easily find the Vx octet of the first VT. Once it has processed one superframe it will have the V1, V2 octets for that VT. It can look at bits 5 and 6 of the V1, V2 word and find out what kind of VTs are in the VTG (in our case, it will contain VT-1.5). Since all the VTs in a VTG must be the same, the SONET equipment will know how many columns are contained in the VT and from that, it can find the other VTs in the VTG[15]. Finding the start of the VTs is easy, of course, because of the interleaving. They are together at the beginning of the VTG. But until the SONET equipment knows what kind of VT, it doesn't know how many VTs are in the VTG.

The V3 octet is used in the same fashion as the H3 octet. It normally does not contain data but will when a negative justification is done. The octet after the V3 octet is used for a positive justification, just like the octet after the H3 octet. The V4 octet has no meaning and is reserved for future standardization.

[15] The VTs are interleaved in a VTG but the SONET equipment can easily account for this.

Okay, I mentioned that the V1, V2 octets form a pointer, but what does this pointer point to? Well, just like the H1, H2 pointer points to the SONET payload (SPE), the V1, V2 pointer points to the VT payload. And just what is this VT payload and where is it?

Well, we started with 27 octets in each VT-1.5 and we used one octet for the V1, V2, V3, and V4 overheads. We now have 26 octets remaining in each payload frame of the superframe. So what we have left is four frames of 26 octets each, or a total of 104 octets. Since there are 104 payload octets, the pointer value for a VT-1.5 goes from zero to 103 (decimal), where zero points to the first location and 103 points to the last location, exactly like the H1, H2 pointer. The operation of the V1, V2 octets is exactly the same as the H1, H2 octets. To increment (decrement) the pointer, the sender can invert the "I" ("D") bits to signal the receiver. Likewise, the new data flag (NDF) can be used in exactly the same fashion as was described for the H1, H2 octets. Now, let's see what those values mean and where they point. See Figure 27.

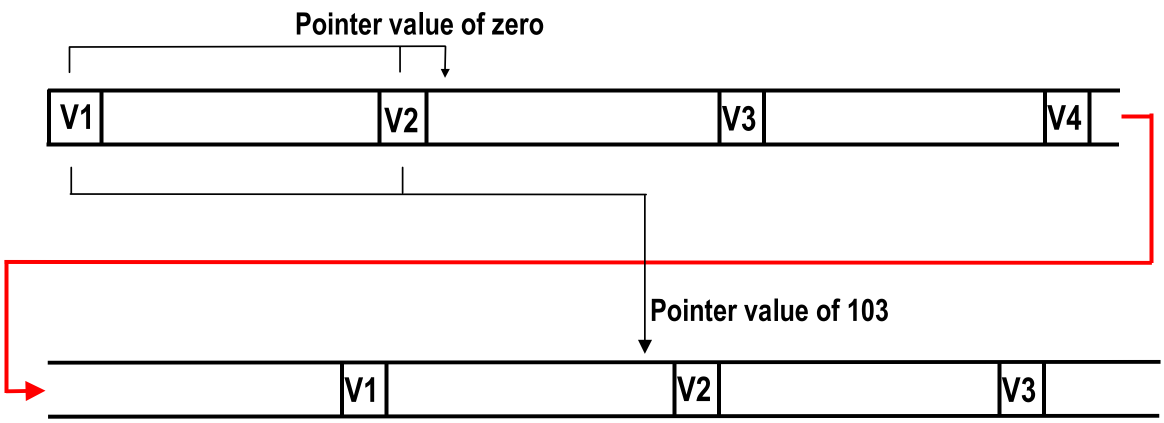

Figure 27: The value of the pointer for different locations of the VT payload.

The V1, V2 pointer points to the first octet of the 104-octet VT superframe. If this octet is located immediately after the V2 octet, the pointer value is zero. Since there are only 104 octets in the VT superframe, the highest value of the pointer is 103, which means that the first octet of this payload immediately precedes the V2 octet of the next superframe.

This 104-octet payload "floats" just like the SONET payload. The only sort of complexity is that the V1, V2, V3, and V4 octets are mixed in with these 104 octets (the total of the VT payload and the V1, V2, V3, V4 octets gives 108 octets) but let's ignore this right now.

The SONET equipment keeps track of where the V1, V2, V3, and V4 octets are so we can think of the VT payload as existing in isolation without the V1, V2, V3, V4 octets. And since we have a four-frame multiframe, the VT payload is divided into four sections, or frames, of 26 octets each. See Figure 28.

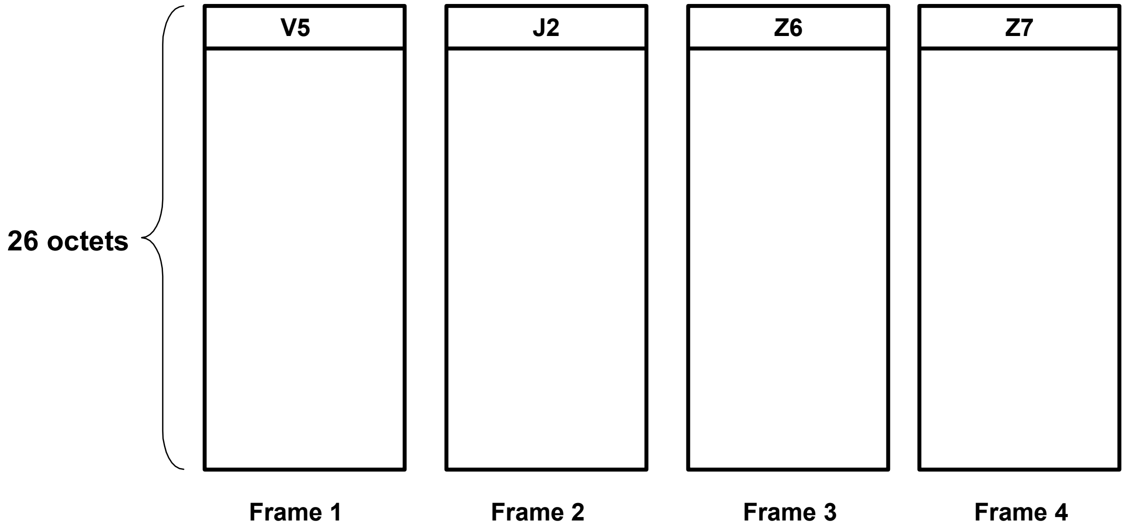

Figure 28: The four VT payload frames of a superframe.

Each payload frame of the VT superframe contains one overhead octet at its beginning, leaving 25 octets in each frame. These overhead octets are the V5, J2, Z6 and Z7 octets and are defined below. Much of the text for the definition of each octet (below) is taken from the T1.105 draft.

VT Header (V5) – *This octet serves a number of purposes but for our discussion, below, we are only concerned with certain values in bits 5, 6, and 7, specifically in the value of 010, which indicates "asynchronous mapping" of a DS-1, and in a value of 100 which indicates "byte-synchronous mapping" of a DS-1. Remember that a value of 11 for bits 5 and 6 of the V1, V2 word indicates that we are mapping a DS-1.* This octet is the first octet of the VT payload, is pointed to by the VT pointers (V1, V2), and provides for VT Paths the same functions that B3, C2, and G1 provide for STS Paths, namely Error checking, Signal Label, and Path Status. The bit assignments of the V5 byte are specified in the following paragraphs and are illustrated in Figure 29. SDH uses this byte for the same purpose.

Bits 1 and 2 of V5 are used for error performance monitoring. A bit-interleaved parity (BIP) scheme is specified. Bit 1 is even-parity calculated over all odd-numbered bits (1, 3, 5, and 7) in all bytes in the previous VT SPE. Similarly, bit 2 is even-parity calculated over the even-numbered bits. Note that the calculation of the BIP-2 includes the VT Path Overhead bytes but excludes the VT Pointers.

Bit 3 of V5 is a VT Path remote error indication (REI) that is sent back towards an originating VT PTE if one or more errors were detected by the BIP-2. REI coding is not suppressed when RDI-V is active. If RDI-V is triggered by a server defect, the REI will be undefined. If RDI-V is triggered by a connectivity defect or payload defect, the REI coding will reflect the incoming BIP errors.

Bit 4 of V5 is reserved for mapping-specific functions. Currently, it is only defined for the byte-synchronous DS1 mapping, in which case it is used for a VT Remote Failure Indication (RFI-V). RFI-V is generated by setting bit 4 of V5 to one. RFI-V is cleared by setting bit 4 of V5 to zero.

Bits 5 through 7 of V5 provide a VT Signal Label. Eight binary values are possible in these three bits. See the T1.105 standard for a complete list of meanings for bits 5 through 7 of V5.

Bit 8 of V5, in combination with bits 5, 6 and 7 of Z7, provide codes to indicate both an old version and an enhanced version of the VT Path remote defect indications (RDI-V). The enhanced version of RDI-V allows differentiation between payload, connectivity, and server defects. Bit 8 of V5 is set equal to bit 5 of Z7. This allows old equipment to receive and interpret an RDI-V indication. Bit 7 of Z7 is set to the inverse of bit 6 of Z7, to distinguish the enhanced version of RDI-V from the old version.

BIP-2		REI	RFI-V	SIGNAL LABEL			RDI-V *
1	2	3	4	5	6	7	8

* RDI-V includes bits 5-7 of byte Z7

Figure 29: The bit assignments of the V5 octet. (source: Draft standard T1.105, Oct, 2000)

VT Path Trace (J2) – *This octet allows the two ends to verify that the connection is still alive and still connected to the right terminations.* This octet is used to transmit repetitively a VT Path Access Point Identifier so that a path receiving terminal can verify its continued connection to the intended transmitter. A 16-byte frame is used for the transmission of Path Access Point Identifiers. SDH uses this byte for the same purpose, but uses the format specified in G.707/G.831.

Growth (Z6, Z7) – *Ignore these two octets.* These octets actually have some complex meanings that will not be described here. See T1.105 for additional details.

Note that we have 25 octets left for each frame (which occurs each 125 µsecond). A DS-1 has 24 octets in a 125 µsecond period so it looks like we'll be okay. But wait! There's a couple of additional problems.

A DS-1 can be used for carrying 24 voice calls, or it can be used to carry data, such as Internet traffic. Also, a DS-1 is not just a stream of octets. Every 24 octets (or voice samples when the DS-1 is used for voice) there's an extra bit used for framing. Additionally, the voice samples may have associated signaling to indicate the status of the voice call.

It turns out that we can communicate all of this by using one octet of each VT frame, leaving 24 octets to carry the DS-1 payload. Let's see how this is done.

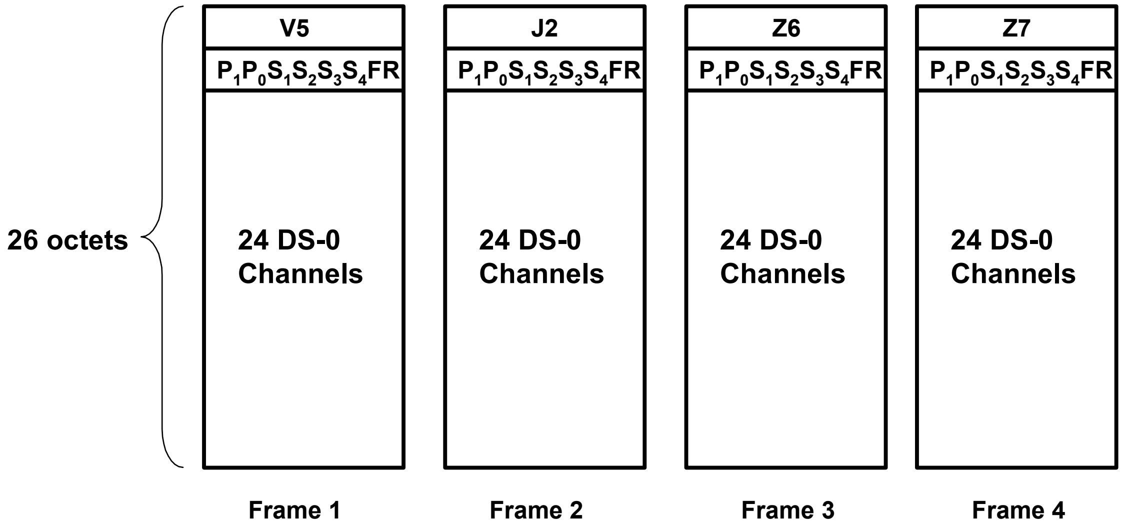

Figure 30: Byte synchronous mapping of a DS-1 signal. The octet after the overhead octets is used to transport the superframe (SF) or extended superframe (ESF) bits, as well as the A, B, C, D signaling bits for each DS-0.

(Note: Understanding the following couple of paragraphs requires that the reader have a good knowledge of DS-1 framing and signaling. You can skip them without great loss.) The "P" bits are used to indicate the phase of the DS-1 signal. For example, a DS-1 using superframe (SF) framing has a superframe of twelve 193-bit frames. A DS-1 using extended superframe (ESF) framing has a superframe of twenty-four 193-bit frames. The two "P" bits allow the framing to be divided into groups of six. For example, a value of 00 is used in the "P" bits to indicate the first six frames of either SF or ESF framing.

The "S" bits are used to communicate the A, B, C, and D channel signaling bits[16]. The status of four channels can be sent each frame. Since each "P" period is six frames, the status of a single indicator (A, B, C, or D) can be sent each "P" period. Thus, for ESF, all channel status bits (A, B, C, and D) can be sent for all 24 channels through one rotation of the "P" bits (from 00, to 01, to 10, to 11). Table 3 gives the meaning of the framing and channel signaling bits for various values of the "P" bits.

[16] These bits are defined as part of a DS-1 and are used to send call status, such as "on-hook," "off-hook," "answer," and a number of other signaling conditions. See [T.403.02] for more information. For our discussion, just accept that they need to be transported as part of the DS-1.

Signalling			Format		
2 State	4 State	16 State	SF	ESF	
S_1 S_2 S_3 S_4	S_1 S_2 S_3 S_4	S_1 S_2 S_3 S_4	F	F	P_1P_0
A_1 A_2 A_3 A_4	A_1 A_2 A_3 A_4	A_1 A_2 A_3 A_4	F_1	M_1	00
A_5 A_6 A_7 A_8	A_5 A_6 A_7 A_8	A_5 A_6 A_7 A_8	S_1	C_1	00
A_9 A_{10} A_{11} A_{12}	A_9 A_{10} A_{11} A_{12}	A_9 A_{10} A_{11} A_{12}	F_2	M_2	00
A_{13} A_{14} A_{15} A_{16}	A_{13} A_{14} A_{15} A_{16}	A_{13} A_{14} A_{15} A_{16}	S_2	F_1	00
A_{17} A_{18} A_{19} A_{20}	A_{17} A_{18} A_{19} A_{20}	A_{17} A_{18} A_{19} A_{20}	F_3	M_3	00
A_{21} A_{22} A_{23} A_{24}	A_{21} A_{22} A_{23} A_{24}	A_{21} A_{22} A_{23} A_{24}	S_3	C_2	00
A_1 A_2 A_3 A_4	B_1 B_2 B_3 B_4	B_1 B_2 B_3 B_4	F_4	M_4	01
A_5 A_6 A_7 A_8	B_5 B_6 B_7 B_8	B_5 B_6 B_7 B_8	S_4	F_2	01
A_9 A_{10} A_{11} A_{12}	B_9 B_{10} B_{11} B_{12}	B_9 B_{10} B_{11} B_{12}	F_5	M_5	01
A_{13} A_{14} A_{15} A_{16}	B_{13} B_{14} B_{15} B_{16}	B_{13} B_{14} B_{15} B_{16}	S_5	C_3	01
A_{17} A_{18} A_{19} A_{20}	B_{17} B_{18} B_{19} B_{20}	B_{17} B_{18} B_{19} B_{20}	F_6	M_6	01
A_{21} A_{22} A_{23} A_{24}	B_{21} B_{22} B_{23} B_{24}	B_{21} B_{22} B_{23} B_{24}	S_6	F_3	01
A_1 A_2 A_3 A_4	A_1 A_2 A_3 A_4	C_1 C_2 C_3 C_4	F_1	M_7	10
A_5 A_6 A_7 A_8	A_5 A_6 A_7 A_8	C_5 C_6 C_7 C_8	S_1	C_4	10
A_9 A_{10} A_{11} A_{12}	A_9 A_{10} A_{11} A_{12}	C_9 C_{10} C_{11} C_{12}	F_2	M_8	10
A_{13} A_{14} A_{15} A_{16}	A_{13} A_{14} A_{15} A_{16}	C_{13} C_{14} C_{15} C_{16}	S_2	F_4	10
A_{17} A_{18} A_{19} A_{20}	A_{17} A_{18} A_{19} A_{20}	C_{17} C_{18} C_{19} C_{20}	F_3	M_9	10
A_{21} A_{22} A_{23} A_{24}	A_{21} A_{22} A_{23} A_{24}	C_{21} C_{22} C_{23} C_{24}	S_3	C_5	10
A_1 A_2 A_3 A_4	B_1 B_2 B_3 B_4	D_1 D_2 D_3 D_4	F_4	M_{10}	11
A_5 A_6 A_7 A_8	B_5 B_6 B_7 B_8	D_5 D_6 D_7 D_8	S_4	F_5	11
A_9 A_{10} A_{11} A_{12}	B_9 B_{10} B_{11} B_{12}	D_9 D_{10} D_{11} D_{12}	F_5	M_{11}	11
A_{13} A_{14} A_{15} A_{16}	B_{13} B_{14} B_{15} B_{16}	D_{13} D_{14} D_{15} D_{16}	S_5	C_6	11
A_{17} A_{18} A_{19} A_{20}	B_{17} B_{18} B_{19} B_{20}	D_{17} D_{18} D_{19} D_{20}	F_6	M_{12}	11
A_{21} A_{22} A_{23} A_{24}	B_{21} B_{22} B_{23} B_{24}	D_{21} D_{22} D_{23} D_{24}	S_6	F_6	11

Notes		
SF Format:	F_1 - F_6	= Frame Alignment Bits
	S_1 - S_6	= Signalling Framing Bits
ESF Format:	F_1 - F_6	= Frame Alignment Bits
	C_1 - C_6	= Cyclic Redundancy Check-6 Bits
	M_1 - M_6	= Data Link Bits
	A_1 - A_{24}	= Signalling Bits
	B_1 - B_{24}	= Signalling Bits
	C_1 - C_{24}	= Signalling Bits
	D_1 - D_{24}	= Signalling Bits

Table 3: Interpretation of the S_n and SF and ESF framing bits for different values of the “P” bits. (source: ANSI standard T1.105.02-1995)

(Okay, start reading again.) This type of mapping is known as byte-synchronous mapping. Byte-synchronous mapping preserves the format of the DS-1 signal, allowing any DS-0 to be extracted anywhere along the communication chain.

Asynchronous mapping, which I describe next, does not maintain the DS-1 payload octet identity – it just takes a group of bits, usually 193 bits and puts them in a frame of the VT superframe. Asynchronous mapping is very common. A great many DS-1 circuits do not carry channelized voice (i.e., telephone calls) – they carry data, usually IP traffic. Because of this, there’s no need to maintain the identity of the DS-1 payload octets; simply carrying the stream of bits is sufficient. The terminating equipment at the ends of the circuit will be able to obtain frame synchronization and extract the payload.

Asynchronous mapping is simpler than byte synchronous mapping. Because of this, the bits in the second octet of each VT frame have a different meaning from byte synchronous mapping. See Figure 31.

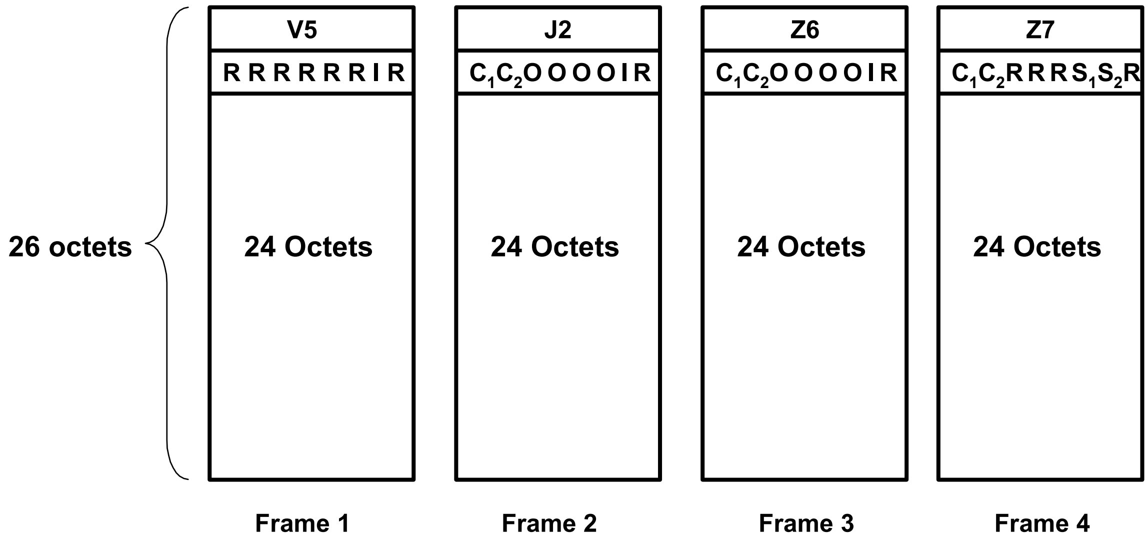

Figure 31: Asynchronous mapping of a DS-1 signal. The octet after the overhead octets is used to transport the framing bits and to accommodate jitter.

The R bits are unused (fixed stuff). The O bits are reserved for future standardization and should be considered the same as the R bits. The I bits are simply information bits, needed because we normally carry 193 bits in a frame.

Note that the last frame of the superframe has different bit assignments in the second octet. The purpose of these signaling bits is to allow the last frame to carry either 192, 193, or 194 bits to adjust for clock differences. Let's see how this is done.

The three C_1 bits are used to control the function of the S_1 bit while the three C_2 bits are used to control the function of the S_2 bit. If all of the C_1 bits are zero, it indicates that the S_1 bit should be treated as data and its content included in the output data stream. If all of the C_1 bits are one, it indicates that the S_1 bit should be treated as a stuff bit and its content *should not* be included in the output data stream. The C_2 bits are used in the same way to control the use of the S_2 bit. Majority voting applies (two out of three in case of a bit error).

So why do we do this. Normally, the last frame of the VT superframe should contain 193 bits. By convention, bit S_2 is used to carry an information bit under this normal situation. If the DS-1 clock is running fast, however, we use the S_1 bit to carry an extra information bit for this one frame of the superframe. The last frame of the VT superframe will carry 194 bits in this case, or 773 bits in the VT superframe.

If the DS-1 clock is running slow, we can signal that the S_2 bit is a stuff bit and only carry 192 bits in the last frame of the VT superframe, or 771 bits in the VT superframe. This allows us to make one clock adjustment every VT superframe.

This completes our discussion of how DS-1 signals are mapped into SONET. The mapping of E1 (2.048 Mbps) is very similar to the mapping of DS-1 signals – E1 signals can be transported with byte-synchronous mapping or asynchronous mapping. Higher levels of signals, e.g., DS-1C and DS-2 signals can only be transported via asynchronous mapping.

Now, we move to examine the mapping of some signals which do not require virtual tributaries – these signals fill the STS-1 SPE. The specific signals we will examine are DS-3 (T3), ATM, packet over SONET (POS), and generic framing procedure (GFP).

Support of DS-3 signals

Carriage of a DS-3 signal is indicated by a value of hex 04 (0x04) in the C2 octet of the POH. When a DS-3 is mapped into an STS-1 SONET payload, columns 30 and 59 of the SPE are stuffed with a fixed value and cannot be used, the same as described above when the virtual tributary groups are used. This leaves 84 columns by nine rows for usable payload. The SPE is broken into nine frames, one per row – see Figure 32.

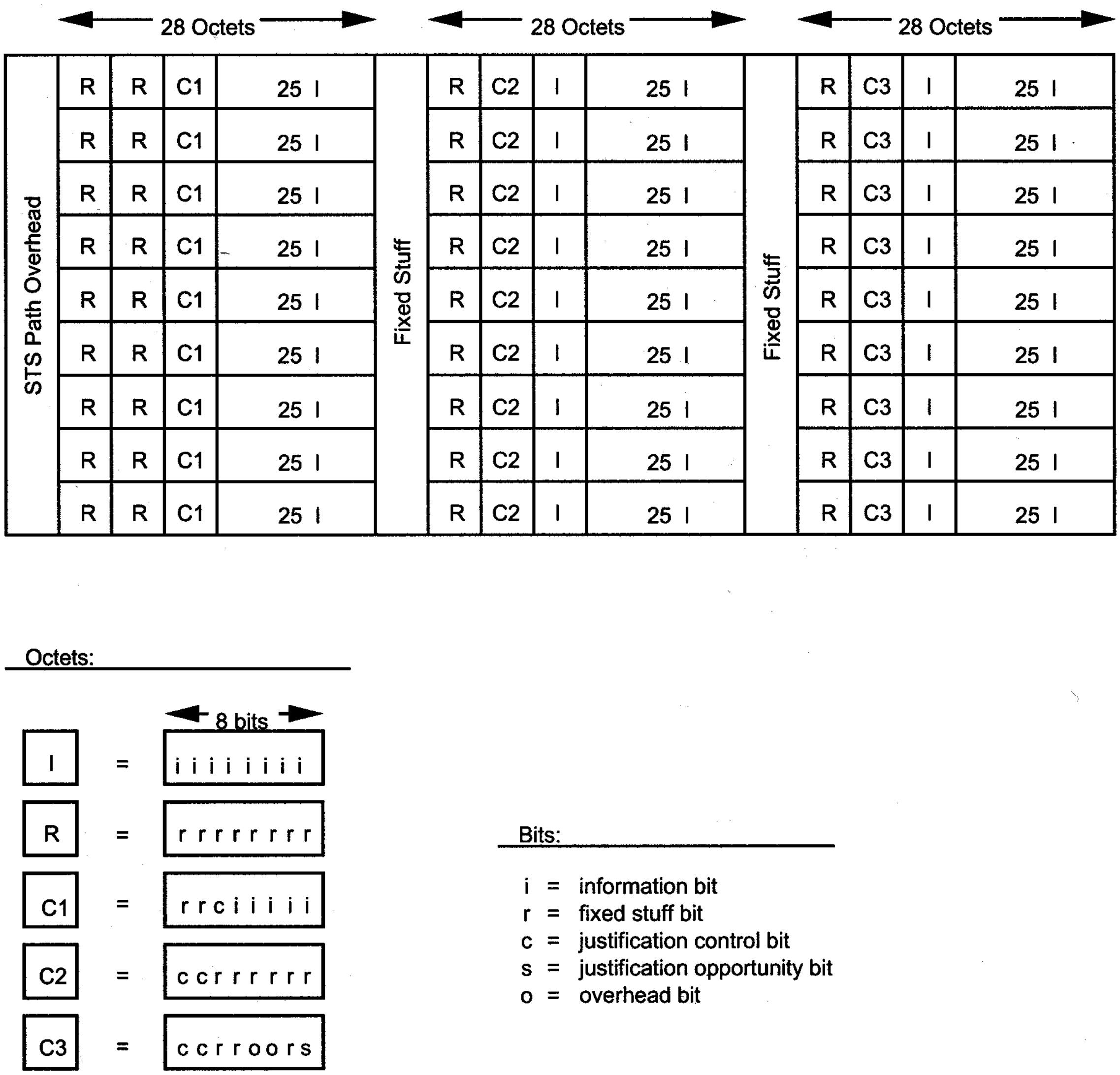

Figure 32: Mapping of a DS-3 signal into an STS-1 SPE (payload). Note that columns 30 and 59 of the SPE cannot be used, just as when VT transport is done. Each row is identical, providing nine frames every 125 µseconds. (source: ANSI standard T1.105.02-1995)

Note that there are 77 full information payload octets in each row. Additionally, the C_1 octet carries five information bits, providing a total of 621 information bits per row. Since there are nine of these frames per SPE and the SPE repeats 8,000 times per second, these 621 bits would provide a bit rate of 44.712 Mbps.

But wait! There's a problem here. A DS-3 operates at 44.736 Mbps. How are we going to carry that rate when we only have a capacity of 44.712 Mbps?

The answer lies in the stuff bit in C_3 and the "c" bits in C_1, C_2, and C_3. When the "c" bits in C_1, C_2, and C_3 are zero, the "s" bit is interpreted to be a data bit and its contents are inserted into the output data stream. When the "c" bits in C_1, C_2, and C_3 are one, the "s" bit is interpreted to be a stuff bit and its value is ignored.

If every frame had the stuff bit set as a data bit, each frame would carry 622 bits, which would give a data rate of 44.784 Mbps. So the use of this stuff bit allows us to achieve the proper bit rate for a DS-3 and also allows us to accommodate clock differences between the DS-3 signal and the SONET signal.

Support of ATM, POS, and GFP

Compared to carrying plesiochronous traffic, carrying asynchronous transfer mode[17] (ATM), packet over SONET[18] (POS), or generic framing procedure[19] (GFP) traffic is a piece of cake. For an STS-1, the payload consists of 9 rows and 87 columns, one of which is the POH, and two are fixed stuff (columns 30 and 50 numbered from the POH). This leaves 84 columns by nine rows for payload.

All three types of traffic require that the octet boundary of the traffic be available. That is, the traffic is placed in the SPE with the traffic octet aligned with the SONET payload octets. This is one reason that POS defines a shielding character instead of using zero bit insertion, as is done in ordinary HDLC. If zero bit insertion were done, octet alignment would be lost very quickly[20].

Beyond that one requirement, the payload of the SONET frame is simply viewed as an octet transport mechanism. The traffic is not examined in any way, nor is there any requirement for any kind of alignment on SPE boundaries. As an example, ATM cells are taken one octet at a time with each octet placed in the next available octet in the SPE without regard for any boundaries in the cell or the SPE, other than maintaining octet alignment. POS and GFP are handled in exactly the same way.

As an aside, note that the SPE of an STS-1 always has columns 30 and 59 of the SPE stuffed and unavailable for payload traffic. If a customer had the option of putting traffic into three STS-1s or one STS-3c, it would be better to choose the STS-3c. Let's see why. The SPE of an STS-3c consists of 261 columns (270 columns minus 9 columns for transport overhead). The POH will take one column of the SPE leaving 260 columns for user traffic.

If the customer used three STS-1, he/she would receive three times 84 columns of payload, or only 252 columns compared to 260 columns for the STS-3c. Eight columns of payload is equal to a little more than 4.6 Mbps, or the equivalent of about three DS-1s. It's one of the oddities of SONET/SDH that part of this extra bandwidth is only available at STS-3c and not at higher levels of SONET/SDH. For higher levels of SONET there are (N/3) - 1 columns of fixed stuff after the POH. This is true for SDH, also. For all levels of SDH greater than STM-1, there are N-1 columns of fixed stuff after the POH (where N indicates the STM level greater than 1).

Automatic Protection Switching

Automatic protection switching (APS) is the function in SONET/SDH which provides the ability to restore service in the case of a failure of a optical fiber line or a network node. Like many things in SONET/SDH, this area is very complex. To simplify things, I'm only going to describe how APS works in two situations, both of which involve ring topology.

[17] Indicated by a value of 0x13 in octet C2 of the POH.
[18] Indicated by a value of 0x16 in octet C2 of the POH.
[19] Indicated by a value of 0x1b in octet C2 of the POH. This value was specified in the latest version of G.707.
[20] The other reason is that handling bits at the highest SONET rates would require very expensive circuitry. It's less expensive to handle the traffic on octet boundaries. And most equipment actually handles traffic in 16 bit words.

SONET/SDH networks can be configured as linear networks, where the SONET/SDH nodes (known as add/drop multiplexers, or ADMs) are just hooked together in a line, as shown in Figure 33. There may be only two fiber connections between the ADMs, as shown in the figure, or four fiber connections, with one set serving as a "protection," or backup, pair.

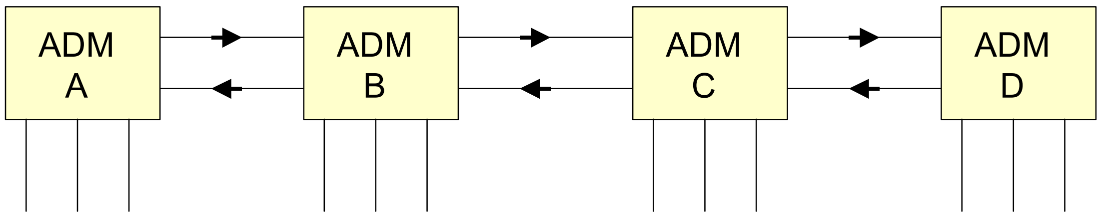

Figure 33: A linear SONET/SDH network.

While linear networks have some applicability, by far the most common topology in the network is the ring, as shown in Figure 34 and Figure 35. Rings are used because they provide an alternate path to communicate between any two nodes. For example, in the linear network, even if two sets of fiber were used between the nodes, it is possible for all of the fibers to be cut at the same time, unless great pains are taken with routing the fiber. And in most cases, the limitations on rights-of-way do not permit this kind of route diversification.

A two-fiber ring can be operated in either of two ways: (1) as a unidirectional ring, or (2) as a bi-directional ring. Let's examine the unidirectional case first. With a unidirectional ring, traffic could be limited to one fiber and always flows the same way around the ring. The second fiber is simply the protection fiber and is used in a special way to provide backup (explained later).

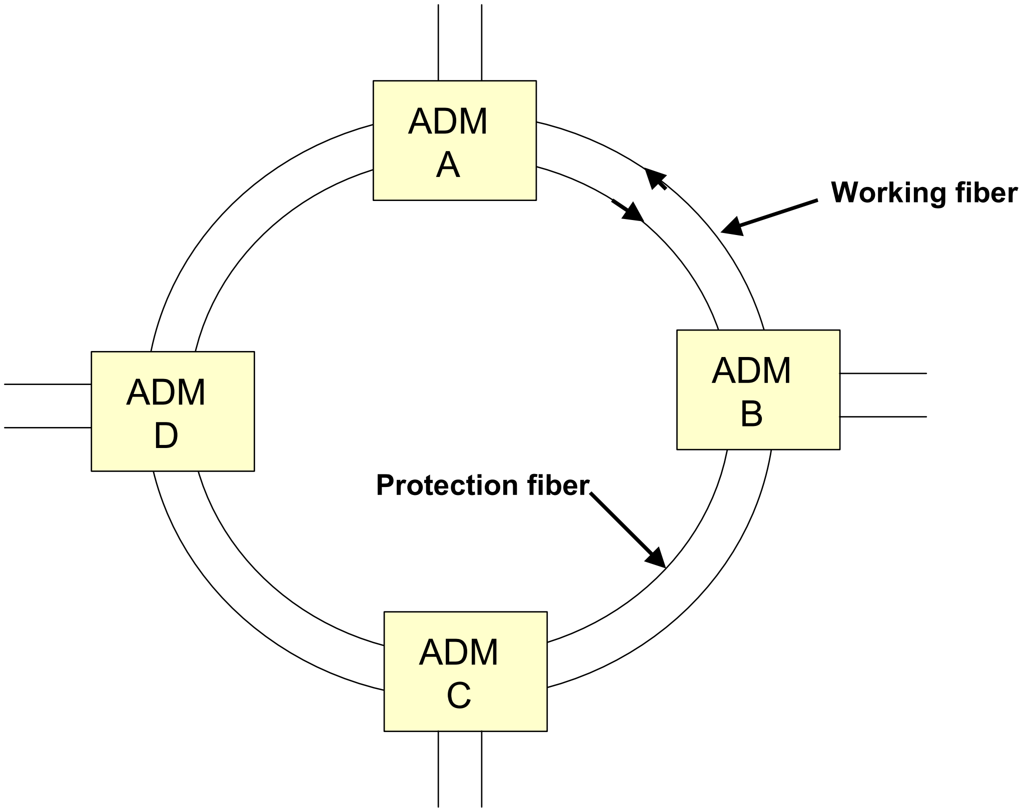

Figure 34: A two fiber SONET/SDH ring. The diagram shown here is for a unidirectional fiber ring. In this type of ring, all the traffic could be carried on one fiber. A two-fiber ring can also operate as a bi-directional ring.

With unidirectional traffic there can be a difference in the transmit and receive propagation delay between two nodes. For example, in Figure 34, if node B sends traffic to node A, the propagation delay is one link. However, when node A sends traffic to node B, the traffic has to go through nodes D and C to reach node B, leading to a longer propagation delay.

With bi-directional traffic, data is sent on both fibers so the concept of "working fiber" and "protection fiber" does not have any meaning – both fibers are working fibers. When data is sent between nodes A and B, it simply flows over the two fibers connecting the two nodes. Bi-directional rings do not buy us any additional capacity, however. In order to be able to provide backup, each fiber in a bi-directional ring can only be used to half its capacity – the second half of the capacity is reserved for backup.

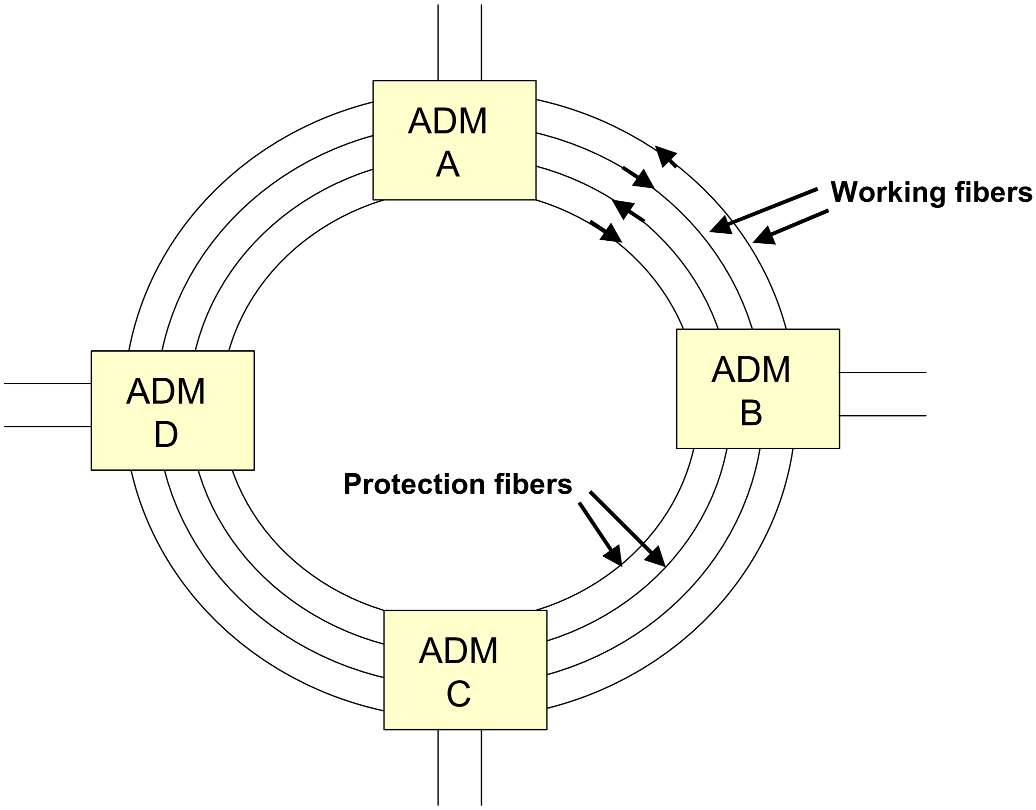

Figure 35: A four fiber (two pair) SONET/SDH ring. This type of ring is always bi-directional.

Four fiber rings are always operated as bi-directional rings. In this case, we get the full rate that can be put on the working fibers, but the protection fibers are not used under normal conditions. No matter how we slice it, rings always require twice as much bandwidth as the amount of traffic carried in the ring, if we want to provide full backup[21].

With four fibers, it is possible to do a link recovery if one, and sometimes two, of the fibers fails between two nodes. The more common case is when all the fibers between two nodes are cut. In this situation, the bi-directional ring provides restoration by routing the traffic over the protection fibers, in the opposite direction around the ring.

There are two backup systems used on fiber – path and line. I'll start by describing a path switched system. Path switching can be implemented on either a unidirectional or a bi-directional ring but, today, path switching is always implemented on a unidirectional ring, and is known as a "unidirectional path switched ring" (UPSR).

In a UPSR system, all of the traffic is transmitted in both directions around the ring, in one direction on the working fiber, and in the other direction on the protection fiber.

The add/drop multiplexers (ADMs) are the places where traffic enters or leaves the ring. This traffic can be at various levels in the SONET/SDH hierarchy. For example, a DS-1 could be feeding into an ADM.

[21] There are configurations where one fiber, or one pair of fibers, backup more than one working fiber, or pair of working fibers. This is known as 1:n backup. These types of systems cannot fully backup a ring, however. If all of the fibers are cut, some traffic must be discarded.

The ADM would put the DS-1 traffic into a virtual tributary, which would then be put into a virtual tributary group, which would then be put into an STS-1 SPE (payload) and this STS-1 might be multiplexed into a higher-level signal, such as an STS-48. The DS-1 would be transported to another ADM, where the traffic would be removed. In this example, the VT carrying the DS-1 traffic is a "path".

It's also possible that an external piece of equipment has placed certain traffic into a concatenated STS-3c and is presenting this STS-3c traffic to the ADM, which then multiplexes it into an STS-48. In this case, the STS-3c is considered the "path" for purposes of the SONET/SDH ring.

The only restriction in path switching is that both the entry and exit nodes for a path are operating at the same level.

In UPSR, the SONET/SDH equipment monitors the path traffic on both fibers, and selects the traffic which is the "best". This monitoring is based on a number of things. The BIP octets, available at every level of the multiplexing hierarchy, provide insight into the number of bit errors on the path. More serious errors can occur, such as the failure of one fiber or detection of an alarm indication signal (AIS) on a path.

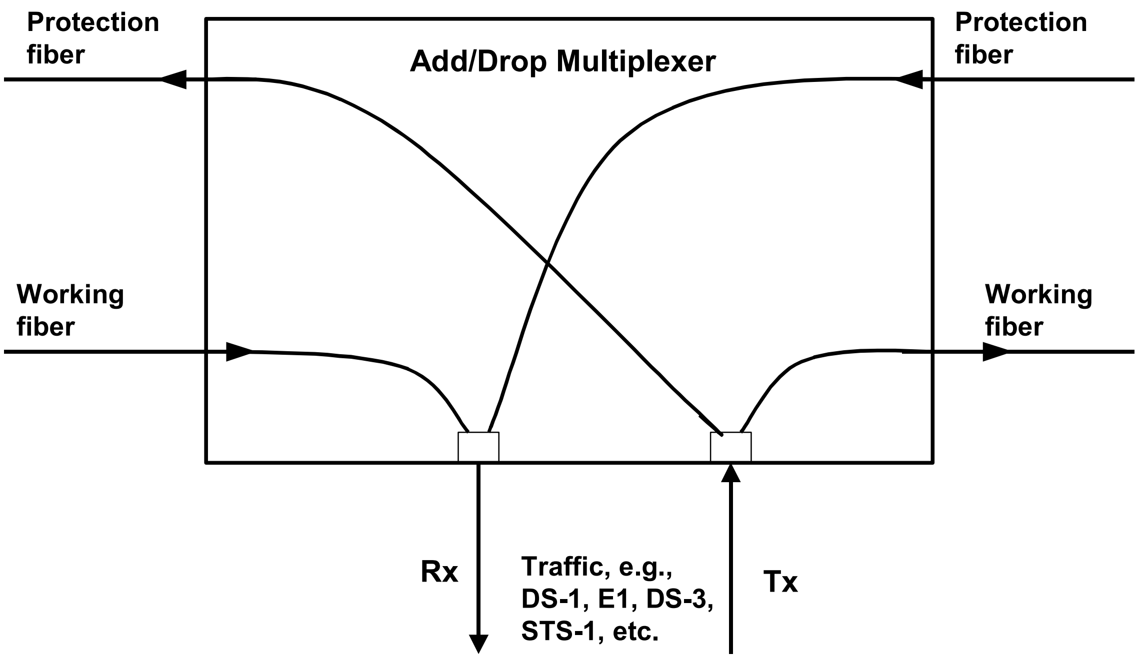

Figure 36: Path switching on a unidirectional ring. On the Tx side of the traffic, the path is sent over both counter rotating rings. On the Rx side, both fibers are monitored and the "best" traffic selected.

Since the receiver of each channel is monitoring both paths on both fibers, switching between fibers is immediate, with no loss of data. Additionally, restoral is accomplished by the receiver, without any coordination with the transmitter – no APS communication channel is needed in UPSR.

The description given above is valid for a fiber cut, which is one of the most common failures in SONET/SDH rings. If an ADM fails, of if multiple fiber cuts occur which cut an ADM out of the ring, additional actions are required. The loss of an ADM will be detected by the two ADMs which were connected to it. These ADMs must put an alarm indication signal (AIS) on each path that originated and terminated on the lost ADM. This is an immediate indication to the device on the other end of the path that the data in that path is bad. Another reason for doing this is that paths may be re-used in a unidirectional ring. Certain types of failures could occur which could cause data received in a path to be coming from the wrong source. The receiving device may not recognize the misconnection and deliver incorrect data. If AIS is put on the path, there is no question the data is invalid.

Path switching on a unidirectional ring has a number of advantages. It is simple for fiber cuts. No coordination is needed between the receiver and transmitter – it is fully implemented at the receiver (for loss of ADMs, additional processing is required). It provides hitless restoral. No data is lost on a restoral unless an ADM fails or is separated from the ring by multiple fiber failures.

On the negative side, it is expensive because two sets of path equipment are needed wherever a circuit is dropped at an ADM. Also, unidirectional rings have the disadvantage of asymmetric delay – the time it takes to go one way around the ring is usually different than the time it takes to go the other way around the ring. Because of this, buffering must be done at the path-terminating site. And as rings get larger and faster, more buffering must be done. This is the primary reason why unidirectional rings are primarily used in metropolitan networks and with lower line rates. In the backbone, especially in large rings, bi-directional rings are much more common.

And that's what we'll discuss now – bi-directional rings. The defining characteristic of bi-directional rings is that the traffic between two nodes flows in two directions, rather than in only a single direction, as occurs in unidirectional rings. For the simplest case of two nodes, e.g., A and B, adjacent to each other, traffic from node A to node B will flow in one direction around the ring, perhaps clockwise, while traffic from node B to node A will flow in the opposite direction.

Bi-directional rings can be either two-fiber or four fiber. In a two fiber bi-directional ring, each fiber can only carry half its capacity, reserving the other half for backup (also known as protection). In a four fiber bi-directional ring, two of the fibers are reserved for protection.

When a fiber failure occurs in a two fiber bi-directional ring, the only recovery possible is a ring switch (or as the standards call it, a line switch), sending data in the opposite direction over the two fibers. This is why the recovery mechanism is called "Bi-directional Line-switched Ring" or BLSR.

In a four fiber bi-directional ring, a single fiber failure can usually be recovered from by doing a span switch, simply switching to the protection fiber over that one link. Failure of multiple fibers will usually require a ring switch.

Look at Figure 37 as we discuss what happens in a bi-directional ring in order to accomplish a ring switch.

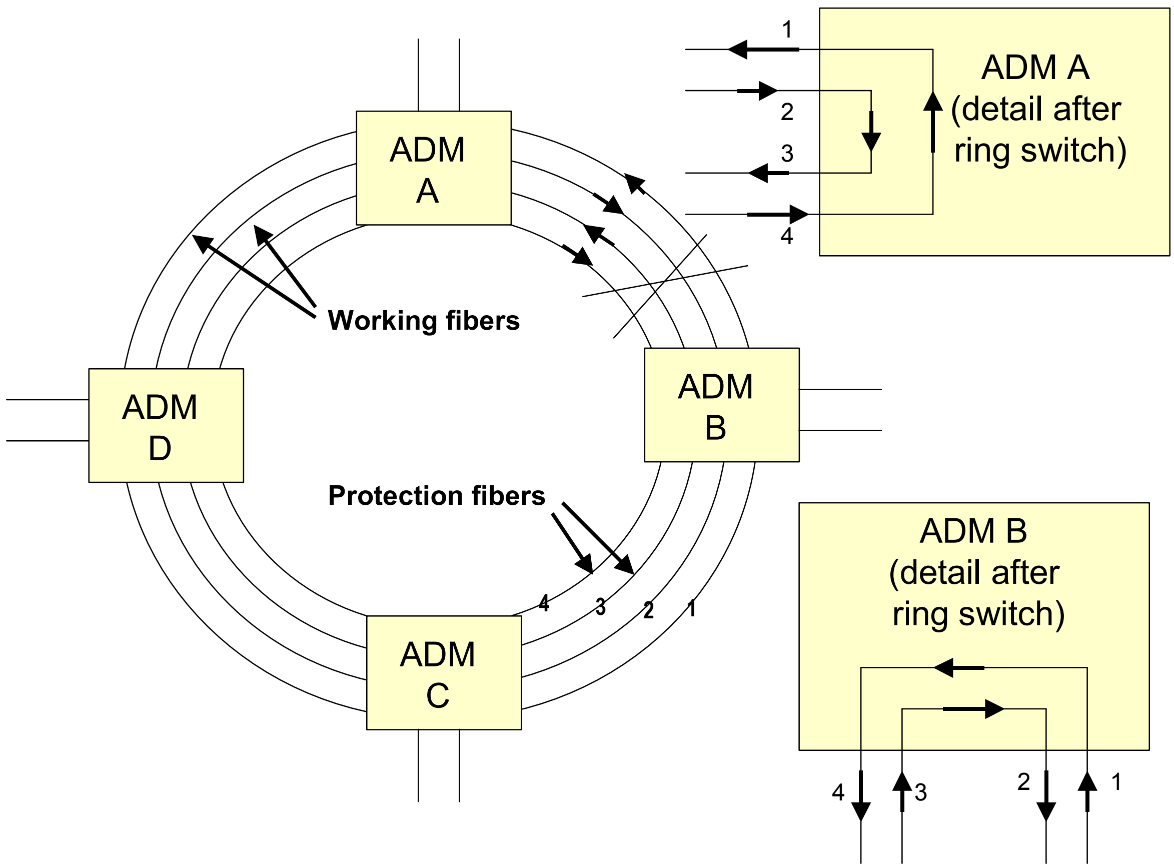

Figure 37: A four fiber bi-directional ring with a complete fiber failure between nodes A and B. The detail diagrams show how the nodes reroute the traffic.

Let's start with a four fiber bi-directional ring as shown in the figure above. Assume that a complete break occurs between ADM A and ADM B. Loss of a single fiber is of course possible but the recovery is a bit more complex. We'll use this example to keep things simple.

When the break occurs, both ADM A and ADM B will detect the loss because both will see loss of signal on the fibers, including the protection fibers. The two ADMs will then send a signal in the K1, K2 octets "backwards" over the ring, with the other ADM's address in the K1 octet. ADMs C and D will simply pass the K1, K2 octets on since the octets are not addressed to them. When ADM A and ADM B receive the failure messages from each other, they bridge the signals as shown in the detail diagrams of Figure 37.

Note what happens once the bridge occurs. Look at ADM B first. The signals arriving on fiber 1 would ordinarily be transmitted to ADM A on fiber 1. Now, however, the signal is put on the protection fiber number 4, which carries it in the reverse direction around the ring to ADM A. ADM A then takes the signal on fiber 4 and bridges it to fiber 1. So the signal on fiber 1 on ADM B still gets to fiber 1 on ADM A, just the long way around.

Signals on fiber 2 on ADM A are transported to fiber 2 on ADM B in the same fashion by fiber 3.

This type of bridging means that none of the other ADMs need to be concerned about the fiber cut between ADM A and ADM B. Those two ADMs handle the fault and the rest of the ring keeps doing the same thing.

The specifications require that the ring switch and restoration of service occur within 50 ms.

When all the fibers are cut, both ADMs see the failure at the same time and take the same actions. If only one fiber is lost, only one ADM will see the failure. In that case, the ADM detecting the failure notifies the other ADM through the K1, K2 octets and the other ADM then responds. In the case of a single fiber loss, the two ADMs will attempt a link switch first by attempting to switch to the protection fiber(s) across the link between them. If that is not possible, they will do a ring switch as described above.

Recovery on a two fiber bi-directional ring is essentially the same as on a four fiber bi-directional ring except that a line switch is not possible – only a ring switch is possible. Signaling with the K1, K2 octets is exactly the same. Traffic is routed back around the ring, not on a separate fiber, but in the unused capacity of the other fiber (which carries traffic in the opposite direction). Just as in the four fiber bi-directional ring, the protection channels simply bring the traffic the long way around the ring until it gets to the node that it would have arrived at if the fiber failure had not occurred.

I will not detail the meaning of the bits in the K1, K2 octets. The purpose of these octets has been described in the text and the interested reader can obtain the details in ANSI or ITU standards or the Telcordia documents.

Closing

Many readers may have found this paper "tough going" with lots of detail and complexities. Unfortunately, SONET/SDH is quite a bit more complex than what was presented in this paper. Many issues were glossed over and many boundary conditions were ignored. My hope, however, is that this paper gave you enough understanding to be able to tackle more advanced descriptions of SONET/SDH, including the standards themselves (should you need a more detailed understanding of SONET/SDH).

High-speed communications is the future. While SONET/SDH is an interesting topic, there are many other fascinating topics in high-speed communications, especially optical communications. I hope your investigation of SONET/SDH is just one stop on your journey in this important and fascinating area.

Appendix A – Synchronous Digital Hierarchy (SDH)

When approached for the first time, most people find SDH terminology difficult and convoluted. Perhaps it's just the way the G.707 standard is written. Once you understand the naming conventions, however, you can read the standard and make perfect sense of it[22]. But if you approach it without any prior knowledge, it can be very confusing and difficult to understand.

One cause of this difficulty is that the base rate of SDH is three times the base rate of SONET. SDH has to do the same things as SONET but it must fit into a frame that's three times the size of the lowest level SONET frame. And the standards people, being good engineers, appear to have provided the maximum flexibility (which leads to maximum confusion). But enough talk. As J. Alfred Prufrock said, "Let us go and make our visit" to SDH land.

The basic STM-1 SDH frame is 270 columns by 9 rows, three times the size of the SONET STS-1 frame. This frame has nine columns of overhead, called Section OverHead (SOH). This is almost the same as the transport overhead in SONET, except that row four is excluded from the section overhead. So the section overhead is the first nine columns, less row four. See Figure 38. The first three rows of the SOH are known as the Regenerator Section OverHead (RSOH), while the last five rows of the SOH are known as the Multiplex Section OverHead (MSOH).

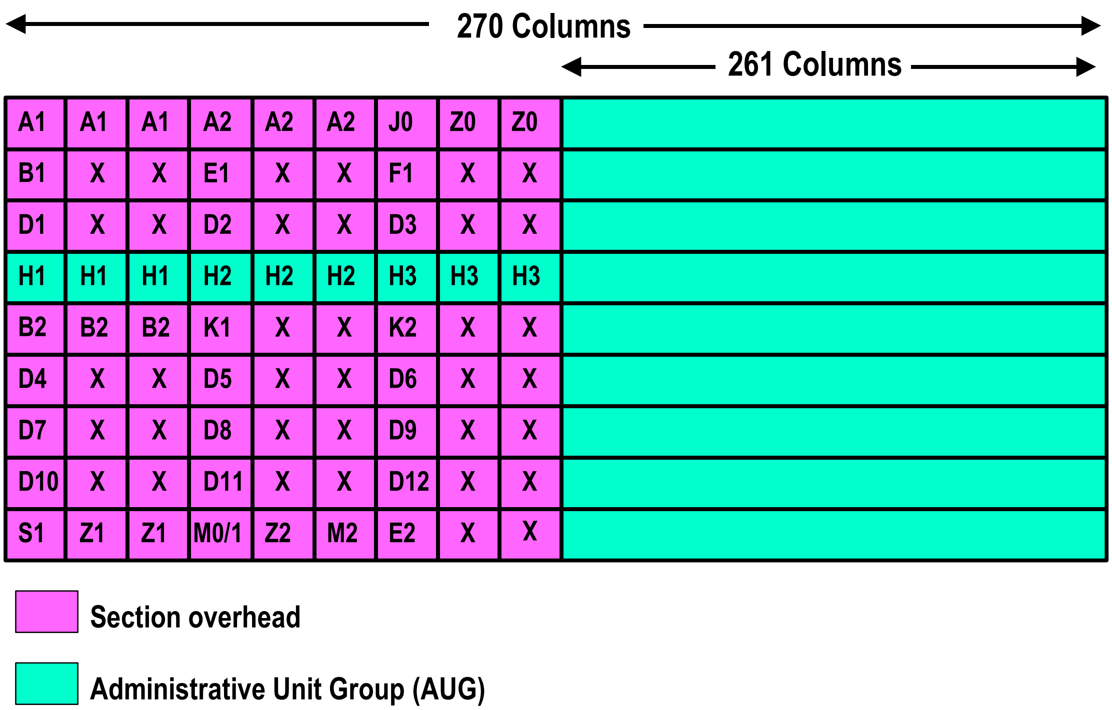

Figure 38: An SDH STM-1 frame showing the section overhead and the administrative unit group. The figure is not to scale – the overhead is shown much larger.

[22] Once you understand the naming conventions, you see that the naming is very regular. However, the G.707 recommendation does not give an explanation of the naming conventions, forcing the reader to discover the conventions by reading the document. A truly frustrating task.

The remaining 261 columns, plus row four of the first nine columns, is known as the Administrative Unit Group (or AUG). An AUG can consist of three Administrative Units, known as Administrative Unit level 3 (AU-3), or one Administrative Unit level 4 (AU-4). As you might suspect, an AU-3 consists of 87 columns, plus three octets of pointers. An AU-4 consists of 261 columns, plus 12 octets of pointers. It causes as extreme amount of confusion to talk about both of these at the same time so I'm going to begin by focusing on the AU-3 and how traffic is mapped into it. Later, I'll discuss the AU-4 and how traffic is mapped into it.

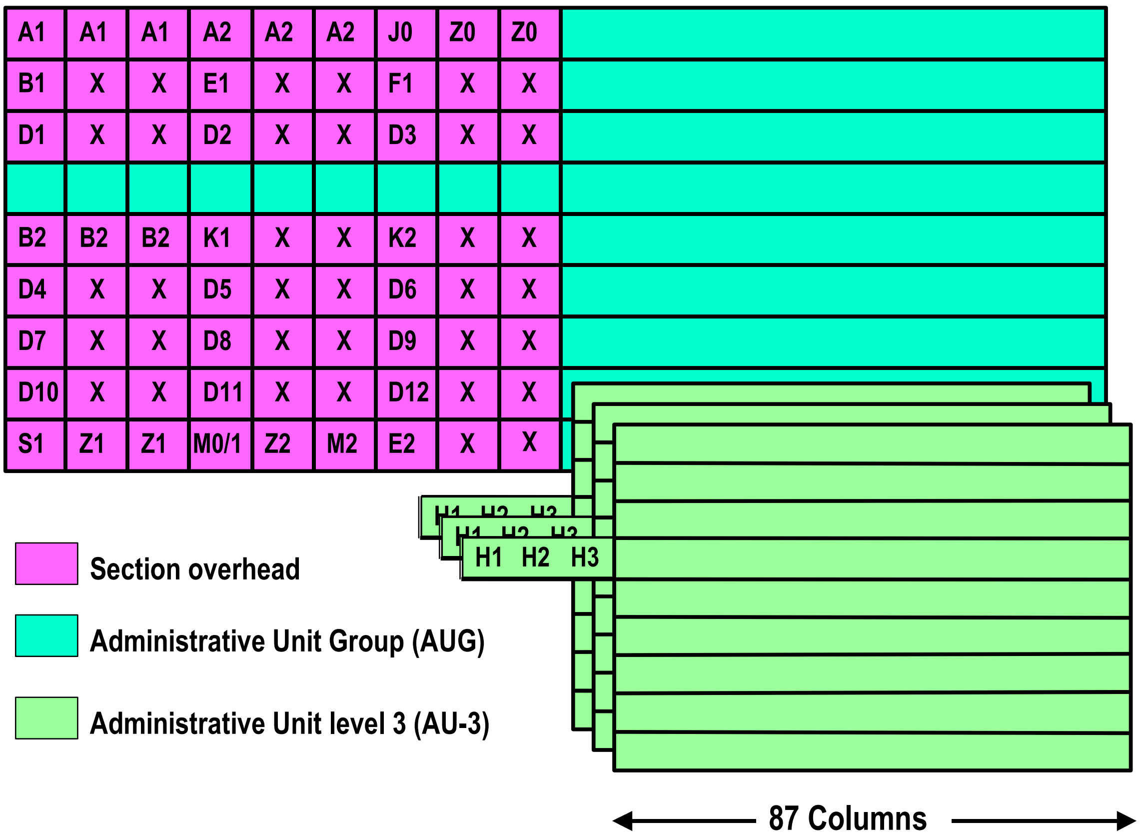

Figure 39: An SDH STM-1 frame with three administrative unit level 3s (AU-3) carried in the AUG. Not to scale.

An AU-3 is a fixed logical structure of 87 columns plus three octets of pointers – the AU-3 does not "float." To provide the ability for the payload to float, as has been described for SONET, SDH introduces the concept of a Virtual Container (VC). A virtual container which fits into an AU-3 is called a VC-3. And just like a SONET STS-1 payload, the VC-3 has one column of payload overhead (POH) and its mapping into an AU-3 includes two fixed stuff columns at columns 30 and 59. However, the VC-3 does not include the fixed stuff columns at columns 30 and 59, so the VC-3 is 85 columns with one column of POH at the beginning. When the VC-3 is mapped into an AU-3, columns 30 and 59 are skipped. See Figure 40.

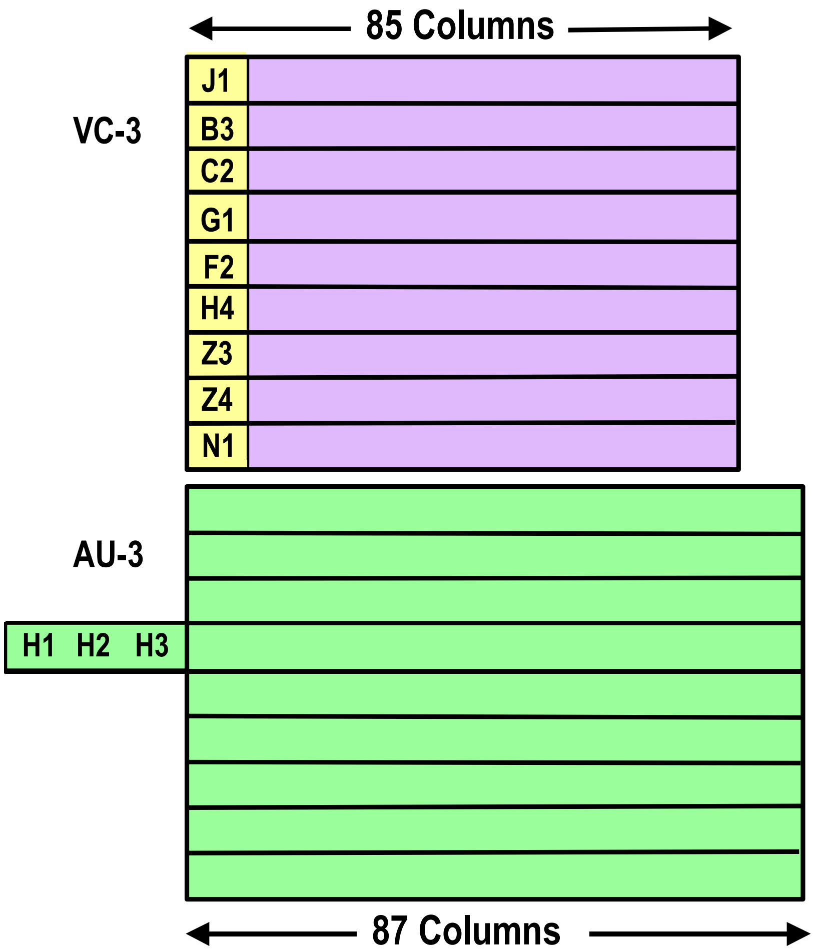

Figure 40: An SDH AU-3 and a Virtual Container level 3 (VC-3) which floats within the AU-3. Note that the VC-3 only has 85 columns while the AU-3 has 87. Two columns (30 and 59) are fixed stuff (same as in SONET) but are not included in the VC-3. When the VC-3 is mapped into the AU-3, these columns are skipped.

The net effect is that a VC-3, plus its mapping into an AU-3, is the same as the payload of a SONET STS-1. Note that the AU-3s are interleaved into the SDH STM-1 structure, in exactly the same fashion as three STS-1s are interleaved into an STS-3. In fact, throughout this section, each structure is interleaved in exactly the same fashion as in SONET.

In the SONET payload we had seven Virtual Tributary Groups (VTG). In SDH we have same thing except that they're called Tributary Unit Groups (TUG). In this case, they're known as TUG-2, perhaps because multiple (seven) TUG-2s map into a VC-3. Handling TUG-2s is exactly the same as handling VTGs in SONET. Now, in SONET the VTG could contain VT-1.5s, VT-2s, VT-3s or a VT-6. SDH provides a similar breakdown but calls them by different names. The equivalent of the VT-1.5 is called the Tributary Unit 11 or TU-11. The equivalent of the VT-2 is the TU-12. There's no equivalent to the VT-3 in SDH – I suppose they decided not to carry DS-1C signals for some reason. The equivalent of the VT-6 called the TU-2 (because it fully fills the TUG-2) and is specified to carry a DS-2 signal.

I'm simplifying things a bit here because there's an additional structure which I skipped. Inside each TU-x is a Virtual Container level x (VC-x) which actually carries the information. But the VC-x is simply the TU-x without the pointer. I skipped this step because it just makes things more confusing to try to add it to the explanation.

A DS-1 signal is mapped into the TU-11 in exactly the same fashion as in SONET (for the VT-1.5). Likewise, the E1 signal is mapped into the TU-12 in the same fashion as an E1 is mapped to a VT-2 in SONET. And likewise for the TU-2 and the DS-2 signal.

At this point, you can start to discern the naming conventions used in SDH. The level 1 is the lowest level structure, carrying the PDH information. The information is carried in a "container" which becomes a "virtual container" (VC) with the addition of the path overhead (POH). The addition of the pointer creates a "tributary unit" (TU). The lowest level of this is the VC-11/TU-11 (level 1 structure, first type) and the VC-12/TU-12 (level 1 structure, second type). The physically smallest gets the lowest number.

So a VC-x and a TU-x (when x is the same) are the same structures except that the TU contains the pointer and the VC does not.

When a structure essentially fills the next level structure, it gets the same number, e.g., a TU-2 fully fills a TUG-2, and a VC-3 fills an AU-3.

TUs then fit into a "tributary unit group" (TUG), which will then fit into a higher-level VC or "administrative unit" (AU). AUs then fit into an "administrative unit group" (AUG).

This concludes the discussion of the AU-3 and its mappings. These mappings are very similar to the SONET mappings so they're fairly easy to understand. Now, let's look at how all of this is mapped when we start with an AU-4 (261 columns plus 9 octets of pointers).

We start with the same AUG as we did for the AU-3 case, but this time we only have one AU to map into it. This AU is known as an AU-4 (to differentiate it from the case of three AU-3s) and completely fills the AUG. See Figure 41.

The pointers are shown as H1, H1, H1, H2, H2, H2, H3, H3, H3. Note that only the first H1 and H2 will carry valid pointers. The second and third H1s will contain 1001 xx11, the concatenation indicator. The second and third H2 octets will contain all ones, 1111 1111, also the concatenation indicator. The H3 octets will be utilized because pointer adjustments will be made three octets at a time. See the SONET section of this paper for more details on concatenation indicators in pointers.

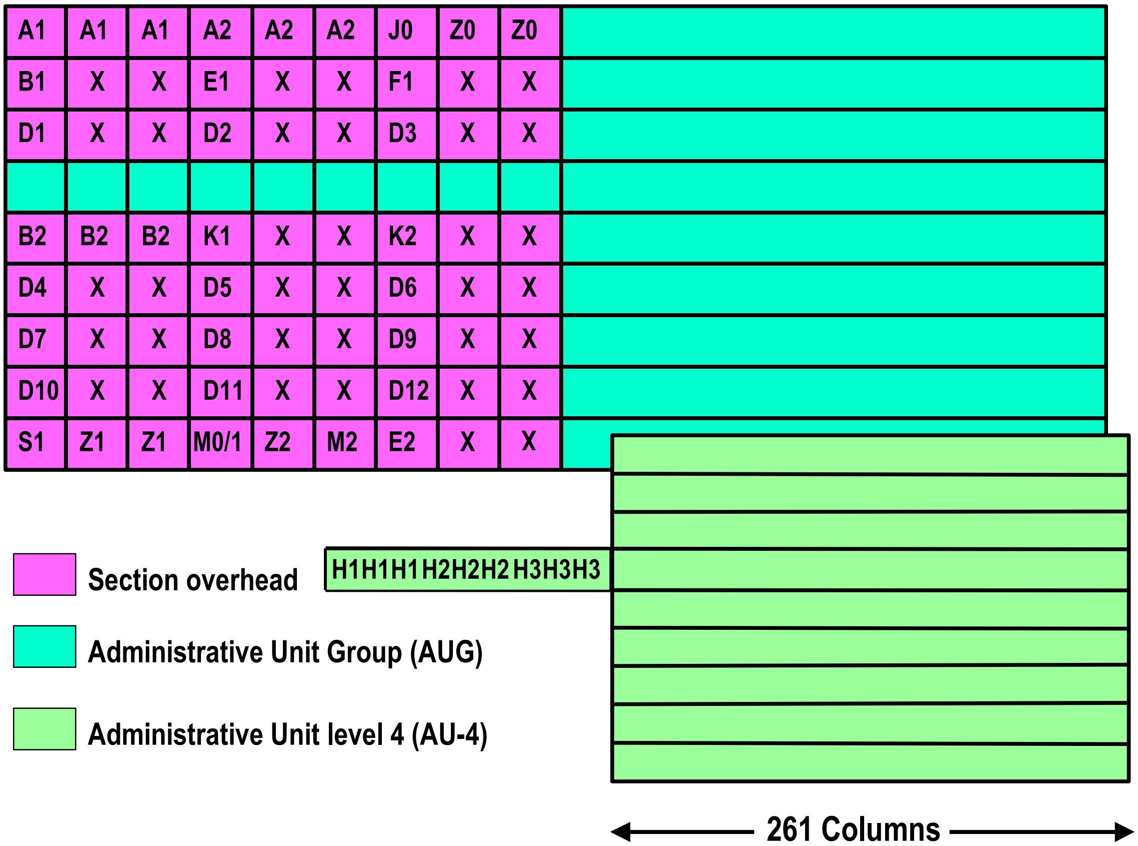

Figure 41: An SDH STM-1 frame with one Administrative Unit level 4 (AU-4) carried in the AUG. Not to scale.

We want to carry the same plesiochronous traffic as described earlier for the AU-3 so we have to find some way to eventually map the TU-11, TU-12, and TU-2 into the AU-4. In fact, if we can find some way to map VC-3s into an AU-4, we can use the same process to get to the TU-11, etc. as we used above.

This is done by defining a structure known as a Tributary Unit Group level 3 (TUG-3), which turns out to function a lot like an AU-3. A VC-3 consists of 85 columns. The TUG-3 consists of 86 columns, with the extra column carrying pointer octets H1, H2, H3. See Figure 42. The H1, H2, H3 pointers in the extra column of the TUG-3 allows the VC-3 to "float" exactly as it does in an AU-3 (the AU-3 provides the pointers to the VC-3). Another note on terminology – an AU-3, for example, consists of the payload area plus the pointer. SDH uses an equivalent structure here, known as the Tributary Unit level 3 (TU-3). This consists of the VC-3 plus the three pointer octets, H1, H2, H3, which are part of the TUG-3. So if you see TU-3, it's just a VC-3 plus the H1, H2, H3 pointers which are in the TUG-3.

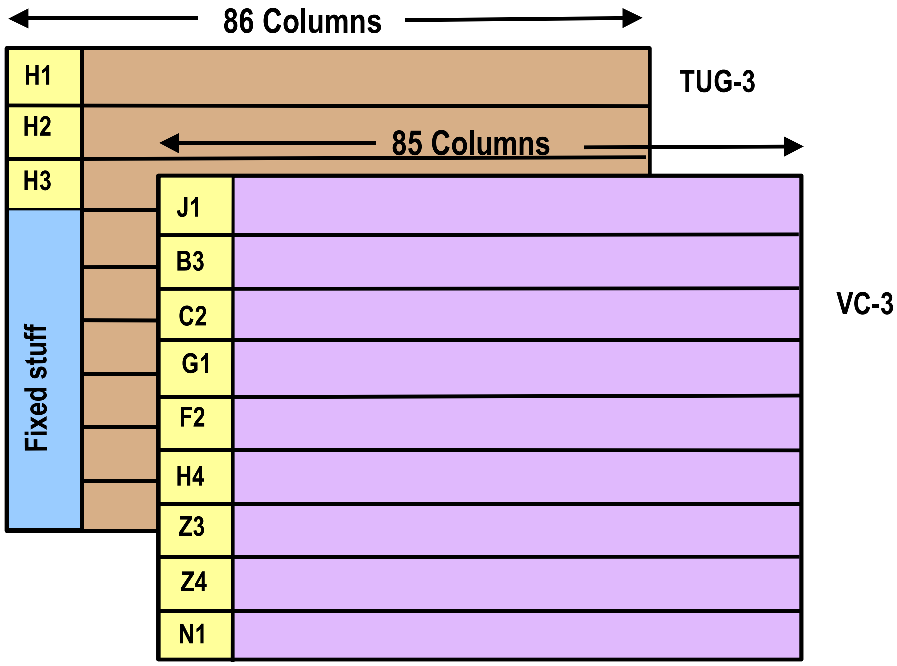

Figure 42: Three VC-3s are carried in an AU-4 by introducing a structure known as a Tributary Unit Group level 3 (TUG-3). Note the extra column in the TUG-3 which carries the pointers which allow the VC-3 to float within the TUG-3.

Remember that the AU-4 is a 261-column structure, while the TUG-3 is an 86-column structure. Three TUG-3s will take 258 columns. Of the remaining three columns of the AU-4, one is dedicated to the payload overhead (POH). The two columns following the POH are fixed stuff to make the number of columns come out right. The TUG-3s are interleaved in the remaining 258 columns.

And just as above, seven TUG-2s are interleaved within the VC-3, and TU-11s, TU-12s, or a TU-2 is carried within the TUG-2, in exactly the same fashion as described above.

So SDH does the same things as SONET but just calls it by different names.

Appendix B – Modulation in Optical Communications

This appendix attempts to provide some additional detail on the format of the modulation of the light in optical systems.

The basic technique of modulation is on-off keying (OOK) where the light is turned off and on to signal the presence of binary ones and zeros. But there's different ways of doing this.

The simplest way is to provide light for the duration of a "one" bit time and turn the light off for the duration of a "zero" bit time. This technique is known as non-return to zero (NRZ). Optical NRZ is different than electrical NRZ because there's no negative light as there is negative voltage. The diagram indicated as NRZ in Figure 43 would be known as return to zero (RZ) in the electrical world.

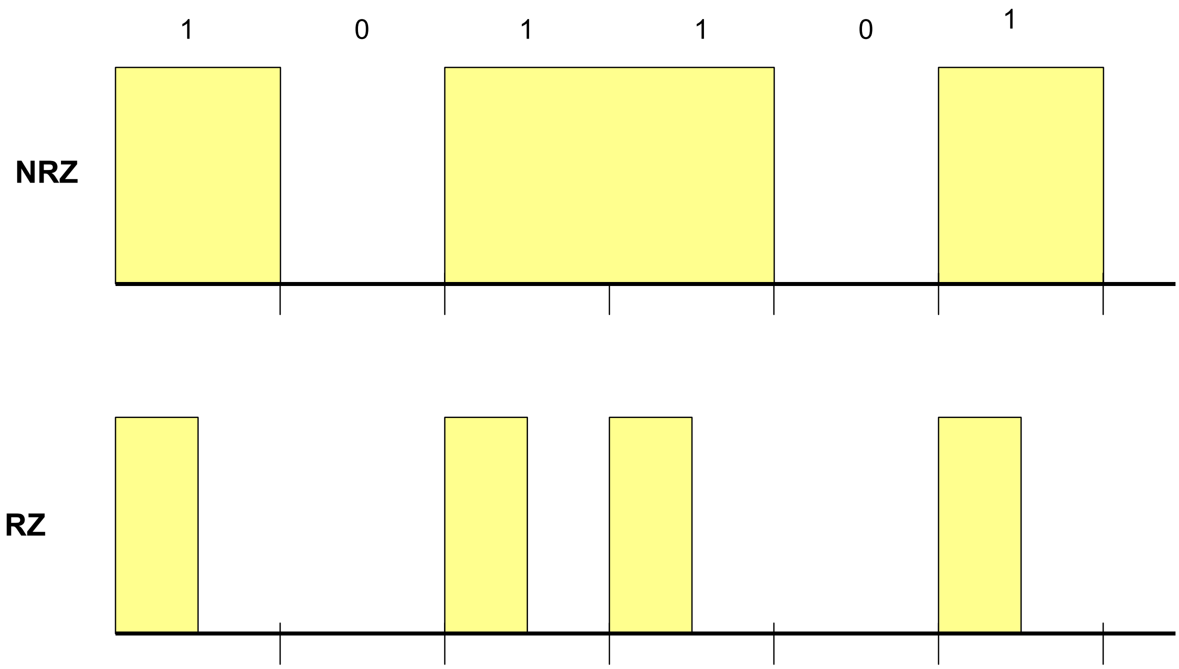

Figure 43: Two common methods of modulating the light signal on optical fibers: non-return to zero (NRZ) and return to zero (RZ).

The other technique is to turn the light on for a one bit, but to turn it off before the bit time, allowing the signal to go dark. The reason for using RZ modulation is to assure transitions for log strings of one bits. With NRZ modulation if a long string of ones was to be transmitted, the light would be turned on and would stay on for the duration of the one bits. Depending on the length of the string, this could cause the receiver to lose timing and be unable to decode the bits correctly.

While RZ solves this problem for long strings of ones, it still permits long strings of zeros and reduces the number of photons seen by the photodetector by half for each one bit. This is a serious problem so RZ is not used for long distance transmission to any great degree.

Another problem is DC balance. DC balance is another term taken from electrical communications but has a slightly different meaning in optical communications. In optical communications DC balance means that the average transmitted power is constant. This is important at the receiver because it affects the decision threshold.

For long distance communications NRZ is typically used along with some technique to minimize long strings of ones and zeros and to maintain DC balance.

The two techniques typically used to avoid long strings of ones and zeros are line codes and scrambling. I won't describe either of these techniques in detail but will note that scrambling does not add any additional bits to the transmitted stream but can not guaranteed to eliminate long strings of ones or zeros. However, it can make the probability of their occurrence low. It also cannot guarantee absolute DC balance.

Line codes map some number of bits, say 8 bits, to a subset of a larger number of bits, perhaps 10 bits. Since I have more numbers with 10 bits than I can have with 8 bits, I can select a subset of the 10 bit numbers and assign one to each 8 bit number. If I do this properly, I can guarantee a sufficient number of transitions (I can limit the number of sequential ones or zeros) and I can guarantee DC balance. The price I pay is that I must transmit more bits; in this case 10 bits must be transmitted for each 8 information bits.

References

G.707 "*Network node interface for the synchronous digital hierarchy (SDH).*" ITU document G.707, 1996, available at http://www.itu.int (last time I checked, G.707 was not available because it was under review, but expected to be published shortly.)

G.707(d) Study Group 15, "*Report of Working Party 3/15, Multiplexing and Switching (Geneva Meeting, 3-14 April 2000), Part II B.5 – Draft Determined Revised Recommendation G.707/Y.1322.*" Document COM15-R71, available at http://www.itu.int/itudoc/itu-t/com15/reports/

G.707(e) "*Network node interface for the synchronous digital hierarchy (SDH) – Editor's draft.*" ITU Working Party 3 working document g707-2000_ww9.doc, available through http://ties.itu.int/u/tsg15/sg15/wp3/ but you need an ITU Ties account to access it.

G.783 *"Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks."* ITU document G.783, available at http://www.itu.int

Gor2000 Goralski, Walter J., *SONET, Second Edition*. McGraw- Hill, 2000. (I hesitate to list this book because, although it's a big book, the portion really devoted to SONET is small.)

GR-253 "*Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria.*" Telcordia Technologies GR-253-CORE, Issue 2, December 1995, with revision 2, January 1999.

GR-1400 "*SONET Unidirectional Path Switched Ring (UPSR) Equipment Generic Criteria.*" Telcordia Technologies GR-1400-CORE Issue 2, January 1999.

GR-1230 "*SONET Bidirectional Line-Switched Ring Equipment Generic Criteria.*" Telcordia Technologies GR-1230-CORE Issue 4, December 1998.

T.105base "*Optical interface rates and formats specifications (SONET*)." ANSI T1.105-1995, available through http://www.atis.org

T.105base(d) "Revised *Draft T105 SONET Base Standard.*" Document number T1X1.5/2000-193R1, available through http://www.t1.org/t1x1/_x1-grid.htm (Check T1X1.5 year 2000 docs.)

T.105.01 "*Synchronous Optical Network (SONET) – Automatic Protection Switching.*" ANSI T1.105.01-1994, available through http://www.atis.org

T.105.01(d) "*Revised Draft T105.01 SONET Automatic Protection Standard.*" Document number T1X1.5/1999-065R1, available through http://www.t1.org/t1x1/_x1-grid.htm (Check T1X1.5 year 1999 docs.)

T.105.02 "*Synchronous Optical Network (SONET) – Payload Mappings.*" ANSI T1.105.02-1995, available through http://www.atis.org

T.105.02(d) "*Revised Draft T105.02 Payload Mapping Standard.*" Document number T1X1.5/2000-192, available through http://www.t1.org/t1x1/_x1-grid.htm (Check T1X1.5 year 2000 docs.)

T.403.02 *"Network and Customer Installation Interfaces – DS-1 Robbed-Bit Signaling State Definitions."* ANSI T1.403.02-1999, available through http://www.atis.org (this specification provides information on the DS-1 channel associated signaling, the A, B, C, and D bits)

T1Rpt36 "*A comparison of SONET (Synchronous Optical NETwork) and SDH (Synchronous Digital Hierarchy)."* ANSI T1X2 Working Group, Technical Report 36, available through http://www.atis.org