Communication Developers Handbook

- ***Top Gotchas and Shortcuts for WAN Access Developers***
- ***Framer Management, Line Testing, Signaling and Performance Monitoring Software***

Updated content includes:

- Sync Status Messaging
- Ethernet OAM

Interim Edition

NComm, Inc.

NOTICE

Communication Developers Handbook

Current Interim Edition February 2009

Printed in the USA

Contents

Contributors 1

Introduction 2

Main Components of a LAN/WAN Access Project 4

- Hardware 5
  - Processor 5
  - Memory 6
  - Framer 6
  - Vendor Equipment Application 10
- Software 10
  - Operating System 11
  - Management of your WAN Access Trunks 13

The Industry Standards, and the “Real World” 15

What is Trunk Management Software? 17

- Main Components of Trunk Management Software 19
  - Configuration Manager 19
  - Alarm Manager 19
  - Maintenance Manager 20
  - Signaling Manager for T1/E1 21
- Sync Status Messages and Clock Distribution Management 22
- Trunk Management and SNMP 25
- Automatic Protection Switching 27
  - History 27
  - Description 28
- WarmStart Capability 28
  - System Behavior with WarmStart 29
  - WarmStart Implementation Objectives 31
  - Framer and Transceiver Devices 32

Designing for High Availability 32
Ethernet in the Wide Area Network 33
HOW-TO Overcome Ethernet OAM challenges in the WAN 34
Ethernet OAM Implementation 42
Ethernet OAM Design Considerations 42
EOAM Alarms (Fault Management) 46
EOAM PMon Module (Performance Monitoring) 48
EOAM APS Module (Automatic Protection Switching) 48
Specific Issues of T1/E1 49
Overview of T1 49
Alarms 50
Framing 51
In-band Loopback Activation and De-Activation 55
Signaling 55
Overview of E1 58
Framing 59
Alarms 64
Signaling 65
Standards Requirements 66
Software Architecture 67
Specific Issues of T3/E3 68
Overview of T3 68
G.747 78
Overview of E3 79
G.751 Framing 80
G.832 Framing 82
Standards Requirements 86
Alarms and Configuration 86
Performance Monitoring 88
Software Architecture 88

Specific Issues of SONET/SDH 89
Overview of SONET 89
Section Overhead 91
Line Overhead 91
Path Overhead 92
Overview of SDH 93
Standards Requirements 99
Alarms and Configuration 100
Performance Monitoring 100
Japanese SDH 101
Handling Groups 102
Automatic Protection Switching 104
Interoperability and NComm's TMS 106
Software Architecture 106
Trends in Embedded Hardware 108
The history of bus architecture 108
PMC/PTMC 111
COMPACTPCI 113
The Future: PCI Express, ATCA, AMC… 114
Growing Demand for VoIP 115
The evolution of telecom equipment 117
Outsourcing solutions that make “cents” 118
Completing the System Design 118
Go to the experts 120
Testing Issues and Gotcha’s 121
Estimating the Development Time for a WAN Access Project 127
Best Options for Development 129
Using NComm’s TMS for SNMP Management 130
Glossary 131

About NCOMM.....138
About LSI Corporation.....139

Contributors

NComm gives its sincere thanks to the following companies that have provided support for this book.

LSI Corporation

Network Systems Design e-zine

Introduction

With the ever-increasing options for accessing the Wide Area Network (WAN), designing communications interfaces has become a more challenging task. New developments, new platforms, new applications of Ethernet and increasing customer requirements make the design of access interfaces a never-ending challenge. If you are designing equipment that requires T1/E1/T3/E3/SONET/SDH or Ethernet interfaces, you need to read the rest of this book. If you are not, you will still find a wealth of knowledge but you might find a good novel more interesting.

You made it to the second paragraph, which means you have such a project to complete. As a manager or senior developer, you are charged with developing a product that includes a WAN or LAN interface, but the project has many other items for you to consider. You have circuit boards to design, a microprocessor to choose, complier and development systems to select, an operating system to select, debuggers/emulators to select, change control systems to choose and implement, etc. etc. You have many items on your plate to make your project, and company, successful.

Fortunately, even with all of these issues to consider, meeting a schedule is never a problem. But wait. Schedule is a problem! Since you aren't doing basic research that may never lead to results, you must deliver products that will pay the bills, including your own salary. So, you need to figure out the best way to get the job done. You have a product to complete on time.

Adding value to the traffic that traverses your product is core to the value of your company. Getting the traffic into and out of your product is probably not a focus; you need to do it, you need to do it correctly, but no one will purchase your product because of it. The most important objective is to get the interface completed and released so that it works well and supports the value added of your particular company.

This handbook will review the major components that need to go into a LAN/WAN access project. It will discuss the technical issues that are essential components of your design. It will outline how to make the proper decisions, and present you with options. It will also discuss make vs. buy decisions, and some of the tools and components available for purchase.

Main Components of a LAN/WAN Access Project

First, why do you have an access interface? Unless you are in the CSU/DSU[1] or NIC[2] business, the value-add in your product relies on traffic being carried over the public network, and the access interface is the mechanism to get that traffic into and out of your product. The traffic can be voice or data, but the traffic still must traverse the interface. And if it is data, it could be ATM, Frame Relay, Multi-Link PPP, etc.

At the Customer Premise Equipment (CPE) or Remote Terminal (RT) side of the network, LAN/WAN interfaces are found on equipment including:

- PBXs
- Routers
- PRI (ISDN) Terminal Adapters
- Remote Access Servers
- M13 and SONET Multiplexers
- Voice Over IP Devices
- Terrestrial bases for wireless systems
- Edge Switches

On the Central Office (CO) side of the network, it is fair to say that most of the equipment will have one of these interfaces. Among these will include:

- Class 5 Switch Cards
- Office Channel Units (OCU) including those with Data Ports (OCU/DP)

---

1 CSU = Channel Service Unit. DSU = Data Service Unit

2 NIC = Network Interface Card

- Digital Loop Carrier (DLC) equipment
- Repeaters/Regenerators

So, what do you do to complete your WAN access project? You start by determining what main component pieces need to be scheduled, and then determine how to best accomplish each of them. Real design means setting priorities. For example, you have probably decided to not write your own complier and linker, right? So, why is that? BECAUSE IT IS NOT SOMETHING THAT ADDS VALUE TO YOUR PRODUCT. It is not a core competency that enhances your company's success (let alone your own career). When was the last time you didn't purchase a product because the vendor did not write his or her own complier? So, to get your project completed, you need to purchase the tools that will help you advance most efficiently and then make what you must to complete your project.

Before we move forward, I'll give you a little advice originally from a top Xerox manager. He said that, "There are three parameters that need to be managed: cost, quality, and schedule. Cost and schedule are performance issues. Quality is an employment issue." Shipping product that doesn't work properly isn't just expensive to fix; it can do your company's reputation irreparable harm.

So, what do you need to do to get the interface going?

Hardware

Assuming you have already determined the physical circuit board parameters -- connectors, bus structure, power, form factor etc., the remaining decisions will focus on the components that are required to make your WAN access design operational.

Processor

One of the most important considerations in the choice of processor is the amount of processing power, usually expressed in MIPS (Millions of Instructions Per Second). You

need to establish a MIPS budget that considers the current *and follow-on* demands of the project. The old rule of thumb still applies that your processor should be roughly twice the capacity that your system requires under normal operating conditions.

As many of you have already experienced, the real MIPS requirement can easily be underestimated because of estimation errors and omissions, real world differences from the theoretical, changes in product definition and “feature creep”, software inefficiencies and follow-on releases on the original product platform. The market has a nasty habit of needing new features and exposing feature holes that usually consume significantly more processing power. While cost is an important consideration, it is probably more costly to cut the processor too close and find yourself mining for MIPS. Eventually the Laws of Physics take over and you are in the middle of a redesign.

Other considerations include the processor track record in telecom/datacom applications, availability of board support packages, third party software and source code availability, vendor design assistance and availability. A significant consideration is staff experience with the device. If your engineering staff has used a specific processor in prior projects, the learning curve is substantially reduced when using the same processor in follow-on projects.

Memory

The basic rule: you have to make sure that you have enough! Memory will be consumed by operating system, and application data. Be very careful to fully accommodate dynamically allocated memory. It may seem obvious, but it is a requirement often underestimated.

Framer

The framer is the hardware component that takes your data stream and maps it to the WAN service (T1, SONET, etc). Framer choice involves a number of considerations. Many come down to cost; this is best looked at as “total” cost. You first need to determine the

number of trunks or circuits that your interface will handle. Will the product offer both US and European flavors of transmission? If there will be multiple flavors, will a single product offering contain all capabilities or will the base be built with different hardware populations and different software to become different models in a line of offerings. Your response to these questions might be a single framer that does T1 and E1, or it might be two different yet pin-compatible devices.

Other considerations include the framer's impact on the MIPS budget, part availability, framer standards compliance, expected part longevity, vendor support and, of course, price. The following two tables list the most popular framers in our experience. The second table lists the framers for handling LIUs.

Popular Framers for WAN Access Projects

T1	E1	T3	E3	SONET
LSI				
Hypermapper (TMXF33625) SuperMapper (TMXF28155) UltraMapper (TMXF84622) UltraFramer (TFRA84J13)	Hypermapper (TMXF33625) SuperMapper (TMXF28155) UltraMapper (TMXF84622)	Hypermapper (TMXF33625) SuperMapper (TMXF28155) UltraMapper (TMXF84622) UltraFramer (TFRA84J13)	Hypermapper (TMXF33625) SuperMapper (TMXF28155) UltraMapper (TMXF84622)	Supermapper (tmxf28155) Ultramapper (tmxf84622) Hypermapper (tmxf33625) MARS10G (TSOT1610G) TSOT16106A TADM042G5
AMCC				
		Orinoco (S1204)		

T1	E1	T3	E3	SONET
Dallas Semiconductor (MAXIM)				
DS2151 DS2152 DS21352/ DS21552 DS21Q352/ DS21Q552 DS21Q48 DS2155 DS21Q55 DS21455 DS21458 DS26502 DS26503 DS26528 DS26519 DS3100	DS2153 DS2154 DS21354/ DS21554 DS21Q354/ DS21Q554 DS21Q48 DS2155 DS21Q55 DS21455 DS21458 DS26502 DS26503 DS26528 DS26519 DS3100	DS3112 DS314X DS316X DS317X DS318X	DS3112 DS314X DS316X DS317X DS318X	
Exar				
		XRT72L71		
IDT				
82P2288 82P2284 82P2282 82P2281	82P2288 82P2284 82P2282 82P2281			
Infineon Technologies				
FALC54 PEF2254 FALC-LH PEF 2255 FALC56 PEF 2256 Quad FALC PEF 22554 Quad LIU PEF 22504	FALC54 PEF2254 FALC-LH PEF 2255 FALC56 PEF 2256 Quad FALC PEF 22554 Quad LIU PEF 22504 Octal-FALC PEF22558	M13FX (renamed to TE3-MUX) (PEF 3445) TE3-FALC (PEF 3460)	TE3-FALC (PEF 3460)	FALC-56 (PEF 2256)
Intel				
IXF3208	IXF3208			IXF19301

T1	E1	T3	E3	SONET
PMC-Sierra				
COMET PM4351 TOCTL PM4388 COMET-Quad PM4354 TEMUX PM8315 TEMUX84 PM8316 TECT3 PM4328 TEMUX84E3 PM8320 PM8321	COMET PM4351 COMET-Quad PM4354 TEMUX PM8315 TEMUX84 PM8316 EOCTL PM6388 TEMUX84E3 PM8320 PM8321	S/UNI QJET PM7346 TEMUX PM8315 TEMUX84 PM8316 TECT3 PM4328 TEMUX84E3 PM8320	S/UNI QJET (PM7346) TEMUX (PMC8315) TEMUX84 (PM8316)	Spectra-2488 PM5315 TUPP+622 PM5363 PM5319 PM5360 TEMUX84 PM8316 Spectra-4x155 PM5316 S/UNI Multi PM5354 Arrow 155 PM5320 TEMUX84E3 PM8320
Transwitch				
				PHAST®-12N TXC06312b PHAST®-12P TXC06412b
Vitesse Semiconductor				
VSC9675		VSC9675		

Popular LIU Framers for WAN Access Projects

T1 LIU	E1 LIU	T3 LIU	E3 LIU
Dallas Semiconductor (MAXIM)			
DS21Q48	DS21Q48	DS3154	DS3154
Infineon Technologies			
QuadLIU (PEF 22504)	QuadLIU (PEF 22504)	QuadLIU (PEF 22504	

Vendor Equipment Application

WAN interfaces are showing up more and more frequently in more and more products. Many people don't realize that even with technologies like Frame Relay and Primary Rate ISDN, a physical level access is still required. Often it is T1 or E1, but can be T3, E3, SONET or SDH. These interfaces will be at both ends of the transmission. Each segment must be terminated properly for correct operation and segmentation for fault location.

Also, T1, E1 and T3 will often be found being carried inside SONET streams. Multiplexing of these lower speed interfaces into SONET or SDH will result in multiple layers of alarm and performance information that requires monitoring at various places.

Bottom-line, you need to get the physical layer interface working and to be able to interface to any other system. Your hardware designers may have some suggestions, but you still have to fit your software architecture with the standards that are required. For example, all hardware framers are not created equal. Some devices do very well implementing some features, but require external hardware (e.g. FPGAs) to implement other features. You need to ensure that your choice of framer can implement the features your product requires, and can also handle the other constraints of your system.

Software

Many software decisions are interdependent on hardware decisions. And, a cool head needs to be kept. Too often you see hardware designers dumping responsibility on the software team, or the software group trying to drive the hardware decisions to make their jobs easier. The real consideration is the impact on the company; this requires a look at the total cost, quality and that schedule thing again. Any addition in hardware cost, means extra dollars ship out the door with every unit sold. On the other hand, adding a capability in hardware might prove more reliable than the same feature in software. Both of these considerations need to be taken against the schedule backdrop. All else being equal, you

can never recover lost time to market. So, these decisions become an optimization, not a maximization issue.

The software component is block diagrammed into basic parts in Figure 1:

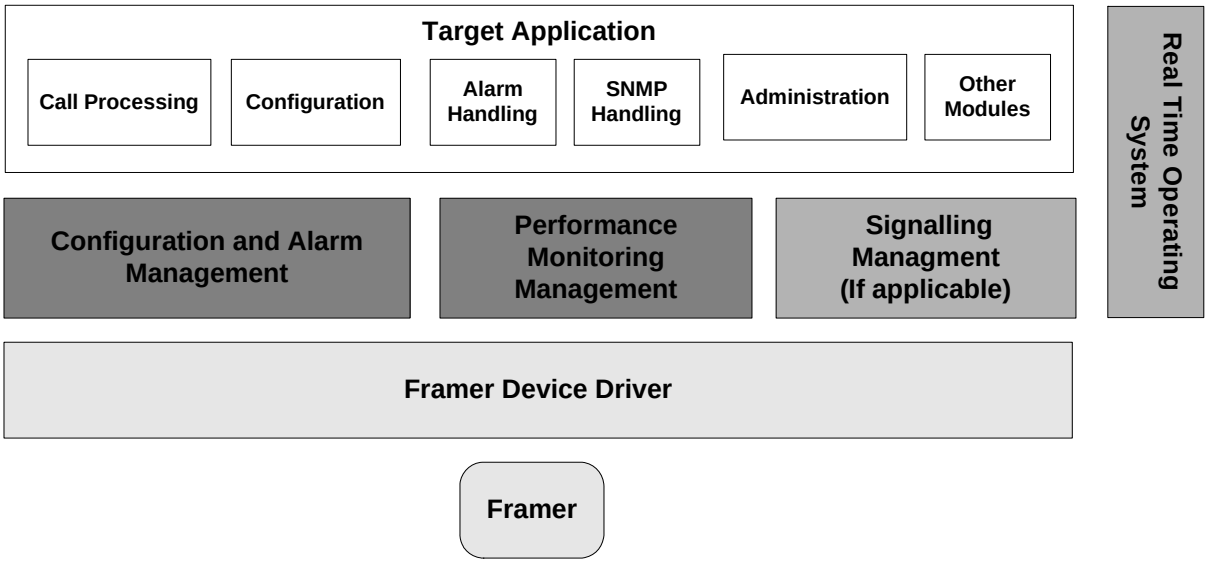


Figure 1. Basic Software Components

Now let's look at two of the most important software pieces and some key considerations:

Operating System

The operating system used for most (if not all) telecom/datacom systems needs to be a Real Time Operating System (RTOS). To properly implement standards, you need to be able to count on processing that is very deterministic. Things need to happen when they must happen, and cannot be held up because some extraordinary event occurred. It is true that some improved OS's that are not, strictly speaking, real-time have included some prioritization that tends to overcome some of our concerns. But, be careful. There are instances where you have less than 10 ms to react and your system better be there to know it!

It used to be that most development groups designed or “rolled” their own OS. We are seeing less of this every year and now there are high-quality, off-the-shelf alternatives. Unless you have some extraordinary requirements, which we haven’t seen yet, you should probably be spending your time adding unique value to your offering. What is the payback of successfully reinventing the wheel?!?

There are a small number of operating systems that once dominated a majority of the WAN access development projects. VxWorks (Wind River), pSOS (Wind River), OSE, and Nucleus (Accelerated Technology) are the real-time operating systems that we saw the most. They all support the top hardware platforms. Each system has its strengths and weaknesses. It is a good idea to gather information from each vendor, and then decide if there are significant design reasons to choose one over the other.

Now, we see a major interest in using Linux OS in communications applications. Originally, Linux was not a real-time OS. In 2003, extensions were available to affect a more real-time performance. With version 2.6, these extensions were integrated even more closely within the OS. Although growing popularity cannot be denied, it should be recognized that while it is “free” there are still significant support costs. Many of our Trunk Management Software users have successfully based their systems on Linux. NComm has now added the Linux operating system as one of our standard pre-ports.

Many design decisions are, and should be, based on the experience of your development team. Having developers already experienced with one OS will reduce your learning curve, and improve problem solving. This is a good rule as long as the OS delivers the real-time features required by your application.

Management of your WAN Access Trunks

The most important items that separate a WAN interface from a LAN interface are that the WAN interface is “expected” to be available 100% of the time, and the WAN interface is more likely to have multiple equipment manufacturers and service providers end-to-end. If your house is burning, you expect that when you dial 911[3], you will always connect to the fire department. To provide high levels of service, the standards for WAN interfaces have been developed so that problems can be located, isolated, and in some cases, fixed before an outage occurs.

Trunk management software is software that manages the WAN (or LAN see the Ethernet OAM section) interface in your product. It sits on top of your framer control code and below your value added application. It permits the proper configuration of the framer, alerts your application to important things that are happening on the line, controls the overhead functions of the WAN circuits, and provides standard compliant operation of your T1, E1, T3, E3, SONET, or SDH interface.

Prime considerations, or design goals, for this trunk management software should include compliance to the applicable standards, code efficiency, and a well-defined Application Programmer Interface (API) that will promote application code reuse, hardware flexibility, and future product growth plans. As will be discussed later, trunk management software is the component that controls the actual WAN interface and is essential for the proper operation of your product. It covers the specialized set of controls and signaling, that makes operation and interaction on the public network possible, standard compliant and interoperable with multi-vendors equipment.

---

3 911 is the number to call in most United States locations for emergency services such as fire, police, and medical.

The specialization of trunk management software means that software developers, without the experience of having written trunk management software previously, have a high initial learning curve. After designing and writing the code, they will experience an unusually long test and debug phase as the variations in what the published standards versus the true operational realities become apparent. Lack of real world experience in this area greatly increases the risk that the WAN interface released to your customers will not perform as expected and required.

The Industry Standards, and the "Real World"

The first thing you need to determine is which standards cover your product. There are a number of standards that may apply; from regulatory standards (FCC and EU), to trade organizations (ANSI, EIA, TIA, and ITU), and carrier standards (AT&T, WorldCom, Telcordia). If your marketing organization has done their job, they should have provided the list of standards for you to implement. You still need to figure out to what extent they apply to your product, and if you really need to implement them. To make your job more difficult, there are different interpretations to many of the standards as well as industry-accepted practices. Here is where you need to start depending on the WAN interface that you are adding to your product. The following matrix shows the most essential standards, for each type of WAN interface:

	T1	E1	T3	E3	SONET	SDH
Alarms	T1.231	I.431 G.731 ETSI 300-233	T1.231 G.747	G.751 G.832	T1.231 T1.105 GR-253	G.783 G.784
Maintenance & Performance Monitoring	T1.403 T1.231 TR54016	G.826 SA bit processing will conform to G.704	T1.231 G.704	G.826 G.751 G.832	T1.231 T1.105 GR-253	G.783 G.784 G.826

	T1	E1	T3	E3	SONET	SDH
Signaling	T1.403 TR-08 GR-303 (Tables 12-3 & 12-4) GR-506 ATT Pub 43801	Q.421 Q.422	N/A	N/A	N/A	N/A
Automatic Protection Switching	N/A	N/A	N/A	N/A	GR-253 T1.105	G.841

The way standards are written often makes it difficult to understand what the standard's body intended. There are sometimes interpretations of the standard that the developer is required to make. On the other hand, subtle changes have been made to the formal standards that have resulted from some field test, experience, and best practice. This adds to the confusion as to what has to be implemented.

A well thought-out design – both hardware and software greatly improves the situation. A good example of this occurred when we went through EU certification for E1. Small changes had to be made to the software to pass the testing criteria. A good software architecture allowed for such adjustments with ease.

What is Trunk Management Software?

Trunk management software is the software required to put a WAN interface e.g., T1/E1/T3/E3/SONET/SDH interface into the network. It provides the ability to integrate hardware devices such as framers into a complete software system. Most framer silicon vendors provide “drivers” that are geared toward performing device design validation. In other words, they verify that the framer is performing the way it was designed.

Trunk management software addresses a different need. The end-user equipment designer does not need to validate the design of the framer manufacturer, but instead needs to put the framer into their product and make sure that it implements the applicable standards correctly.

Application Programmer Interfaces (APIs) help to modularize the design and development of any sizable software project by providing a layered model and are widely used. The following figure shows the layered approach to the software architecture design that uses APIs.

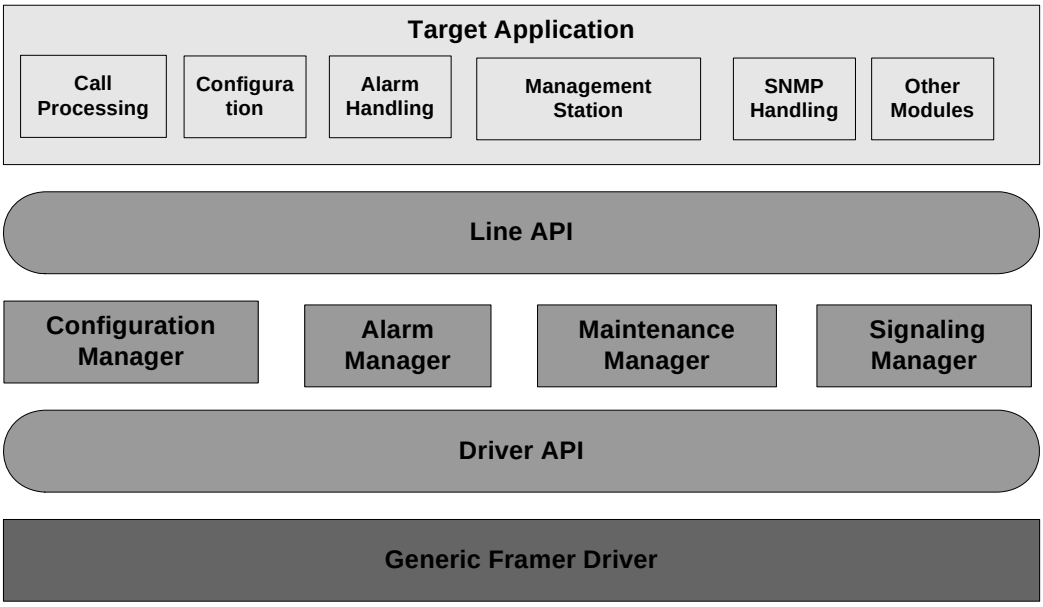


Figure 2. Software Architecture

The top layer of software is the application. This is normally the company specific or value-added layer.

The middle layer(s) is the trunk management layer that must be there to provide the line management or trunk processing protocols. An API at the top of the trunk management layer interfaces with the application, and a similar API interfaces with the framer device driver and its associated hardware. The trunk management layer should include different modules based on the requirements of the application. These include the *Configuration Manager,* the *Alarm Manager,* the *Maintenance Manager* and a *Signaling Manager* module plus a module that is a part of the APS system. These are described later.

The bottom layer is the driver layer that interfaces directly with the hardware and provides an autonomous interface so that it is possible to develop the application independent of the hardware.

Device drivers provide a portable interface to assist the application development on the industry standard devices. The device driver should be implemented so that it does not

require operating system services and thus is portable across different operating systems and platforms.

Main Components of Trunk Management Software

From here on out, we will regularly refer to examples of functionality from NComm's Trunk Management Software. Whether or not you are using this specific software, the capabilities and standards are exactly those that you will need to duplicate in your product.

Configuration Manager

Configuration management sets the proper parameters that match the line service to the WAN interface. It also sets user configurable parameters that regulate the kinds of information that will be reported and the rules of reporting. For example, the default integration time per T1.231 for declaring a red alarm is 2.5 seconds. If an application desires this declaration to happen in 1 second, it can be reconfigured as such. Major parameters that may need to be configured include:

- Line Encoding
- Line Framing Format
- Line Build-Out (Short haul/Long haul w/ distance parameters)
- Alarm integration including programmable integration timers
- Selection of clear channel, idle channel, and/or signaling model. This needs to be selectable on a per voice channel basis.
- Selection of user side or network side for the interface
- Configuration of addresses for maintenance protocols
- Selection of remote, local, payload, and diagnostic loopbacks under application control

Alarm Manager

WAN interfaces use the terminology of "Alarm" to signify any failure of the interface. For instance, three different alarms (red, yellow, and blue) are used to indicate different

problems in the transmission or reception of data in a T1 system. Other functions included in alarm management include:

- Standards compliant detection, declaration, and clearing of all alarm conditions per T1.231
- Programmable alarm integration timers for OOF, LOS, AIS, and RAI. Standards compliant responses to far-end alarm conditions as per T1.231
- For E1 and E3, the alarm capabilities must meet standards per I.431, G.732, ETSI 300-233
- For SONET and SDH, handling Enhanced RDI, TIM defect detection, PLM defect detection and PTI strings.

Maintenance Manager

The Maintenance Manager handles the gathering, processing and communication of performance data, initiating and responding to loop back signals, monitoring and providing threshold crossing alerts, establishing/managing special links like the FDL (Facility Data Link) and PMDL (Path Maintenance Data Link), and passing BOC/BOM messages (Bit Oriented Code/Bit Oriented Message). Other functions contained in the maintenance manager may include:

- Manages and responds to the Facility Data Link (FDL) per TR-54016
- Bit Oriented Message/ Bit Oriented Code (BOM/BOC) handling per T1.403
- Programmable Loop Back codes
- In band and out of band Loop Up and Loop Down code reception and transmission
- 1 second performance report collection and transmission per T1.403
- 96 (or 192), 15-minute performance data bucket collection and transmission for the Near End per TR-54016 and T1.231.
- Responding to and transmitting 15-minute performance data buckets over the FDL for the Near End per TR-54016

- Far end performance data collection in 96, 15-minute buckets and 24-hour summary based upon receipt of T1.403 PRMs from the far end per T1.231.
- For E1, E3 and SDH, performance monitoring must meet the standards per G.826 and provide a 15-min/24-hour data performance database as well. SA bit processing must conform to G.704
- For E1, control of Sa, Si, and X bits.
- Threshold Crossing Alerts

Signaling Manager for T1/E1

The Signaling Manager sends and receives Robbed-Bit and Channel Associated Signaling. Along with Signaling System 7 (SS7) and the D Channel in ISDN, Robbed-Bit Signaling and CAS are the only ways to place conventional telephone calls. This signaling tends to be used at the edges of the network as opposed to the core.

Common Channel Signaling (CCS) (Not to be confused with Channel Associated Signaling) is used by either T1 or E1 and refers to a system that does not use a specific bit structure for signaling. Instead, all or part of a channel is used to pass messages between two systems to indicate how a channel is being used. This type of system is commonly found in ISDN, which uses a D channel to pass messages.

Functions that need to be supported in the module include:

- Robbed-Bit or Channel Associated Signaling (CAS) communication that will accommodate a different signaling model per timeslot
- T1 Robbed-bit signaling allows customer choice of signaling models per T1.403 – 1999, TR-08, or GR-303-CORE-Rev3 (Tables 12-3 & 12-4). Choice may be done on a per DS0 basis
- E1 CAS signaling models as defined in Q.421, Q.422 and including on a per E0 basis
- Processes signaling freezing and debouncing
- Provides call state information such as wink, flash, on-hook, loop open, etc. to the application layer per the selected signaling model

- Transmission and reception of dial pulse digits.
- Timers associated with signaling such as wink, hook flash, and digit on/off ratios, must be programmable on a per timeslot basis
- TR-08/SLC-96 and T1.403 Tri-level signaling supported with external hardware support for generation and detection of the toggling state.
- Publication 43801

Sync Status Messages and Clock Distribution Management

The telecom network functions as a synchronous finite state machine where the timing is sourced from a common source. The timing is then distributed throughout the network from one central office to the next via the equipment located in each central office. Since timing can be distributed across many cites and geographical locations, designing the timing distribution throughout the network becomes an engineering task – the synchronization plan.

The synchronization plan is critical to the network design to prevent timing related problems. In networks where timing is NOT properly design will result in:

- Pointer justifications in SONET and SDH networks
- Slips in T1 and E1 networks.
- Uncontrolled slips in channel banks

All of these problems will result in lower quality services being provided by the network. A network designed with a properly synchronization plan will eliminate these problems and will have a higher quality level provided to the customers of the network.

So, we design a valid synchronization plan and we are all done? Well, not really. How do you know the synchronization plan is implemented correctly? What happens when equipment fails are facilities between offices fail? These items will cause even a valid synchronization plan to fail. That is, unless there are some automatic recovery mechanisms in place to automatically address these problems and let the network recover

from these types of failures. Automatic Clock management via Synchronization Status Messages (SSM) is the mechanism design to manage the distribution of clocking throughout the network.

Clock Management in the Network

Clock management and clock traceability have always been important in the network. Sync Status Messages (SSM) provide clock traceability information that is used to make decisions about which available clock to use. SSMs are nothing new, and specifications were standardized and implemented decades ago. SSM message can be carried over:

T1: In T1, SSMs are carried via Bit Oriented Code words (BOCs, also know as Bit Oriented Messages, BOMs).

E1: In E1, SSMs are carried via the national bits on a CRC4 formatted E1.

E3: In E3, the SSMs are carried via the Timing Mark or SSM field in the E3 frame.

SONET/SDH: In SONET and SDH, the SSMs are carried via the S1 Byte in bits bits 5-8.

T3: T3 cannot be used as a timing reference and has no SSM capabilities.

Each type of facility listed above can transmit and receive SSMs. The goal of Clock management is to examine the quality of the clocks being received by the system, select the best quality clock, and then propagate the selected clock to down stream network elements. In addition to the facilities listed above, a system can be synchronized to a BITS (Building Integrated Timing Supply) source. Some common sources of BITS clock are GPS (Global Positioning System) satellites.

The SSM Mechanism

An SSM system sends and receives Sync Status Messages and performs algorithms to optimize the choice of available clocks and propagate timing to down stream elements.

Although its implementation requires complying with standards (GR-253, GR-1244, ETSI 300-417-6-1 and G.781), the standards provide the objective without the corresponding implementation details. Thus, a successful implementation of the standards relies on a detailed understanding of networks and their pit falls.

Standards issues aside, the successful SSM system must meet a number of requirements and provide certain functions. These include:

- Maintaining traceability to an identifiable primary clock reference source.
- Ensuring that higher stratum (accuracy) clocks are selected over lower stratum clocks.
- Ensuring that all Network Elements have both a primary source and a secondary source in case of a primary failure.
- Ensuring that timing loops are avoided.
- To meet these objectives, the system must do the following:
- Determine the proper source from which to generate timing by processing the SSMs
- Interface to a Timing Device Driver (i.e. DS3100 chip's timing functions).
- Process the timing inputs from external Line cards (e.g. S1 bytes, TM, BOCs, SA bits) A USELESS comment.
- Provide support for redundant Timing Card control and switching between them to meet equipment redundancy requirements.
- Provide transmission relay of the proper SSM messages.
- Execute algorithms for controlling the timing per GR-253 for SONET/T1, and ETSI 300-417-6-1 and G.781 for E1/E3/SDH.
- The algorithms should default to the standards, but permit selectability and configurability by the user.
- Provide a well-defined API to enable portability across software and hardware implementations.

There is one last thought to keep in mind when implementing some of the less specific areas of the standards. Meeting carrier SLAs (Service Level Agreements) will be partially

dependent on the effective functioning of your SSM system. Network availability is important to all users of the network. In the final analysis, design decisions must meet these real-world needs.

Trunk Management and SNMP

Not that many years ago, Simple Network Management Protocol (SNMP) was implemented primarily for LAN applications and within that, mostly at the customer end of the network. As newer WAN technologies like frame relay were deployed, SNMP was extended to control the network end to end. The underlying transport layers continued to use various Operations Administration Maintenance and Provisioning (OAMP). SNMP has been receiving wider and wider acceptance as the protocol of choice to manage WAN products.

SNMP, created in 1988, provides a UDP/IP based protocol for managing devices over a network. At the core of the SNMP protocol is the Management Information Base (MIB) that describes the items of a device that can be managed by the SNMP protocol. Each MIB is defined in a Request for Comments (RFC) Document as managed by the Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG). MIBs can cover many items such as printers, protocols, etc. as well as WANs.

The SNMP protocol implements three basic primitives: SET, GET, and TRAP. These primitives allow the SNMP management station to modify and/or retrieve the behavior of the device managed by SNMP. The MIB describes the “objects” that can be managed by the SNMP protocol. SET allows new values to be assigned to the objects, GET allows the objects values to be retrieved and TRAP allows the device to inform the manager about “important” events.

SNMP is being selected as a replacement to Operational Support System (OSS) protocols such as TABS and/or TBOS. Increased focus in the network for data transport of voice traffic (such as VOIP) has also increased the demand for SNMP management of WAN

products. SNMP and Trunk Management Software are not substitutes for each other. The SNMP MIBs map into the information that the trunk management software provides. For Wide Area Networks, the RFCs that are currently in force are listed below:

- DS0/Voice Channels – RFC 2494
- T1/E1 – RFC 3895
- T3/E3 – RFC 3896
- SONET/SDH – RFC 3592
- Automatic Protection Switching for SONET/SDH – RFC 3498

Implementing the SNMP management for a WAN device is not as complex as it seems because lots of the work is already done for you. It should be an exercise in mapping existing information provided by the TMS interface to the MIB objects. In a typical product development environment, you will want to select a SNMP development system such as that available from SNMP research (http://www.snmp.com/).

The SNMP development system will contain the software that understands the SNMP protocol and has the software interface routines required for the networking software, such as Ethernet. The development system will also contain other tools that make developing an SNMP Agent (the software that will run in your device) easier to build and debug. One of these tools is the MIB complier, which will read the RFC[4] that contains the MIB and generate a set of “stub” routines. The stub routines will already work with the networking software so that your equipment will execute the SNMP protocol correctly. However, these are just stubs that have the SET and GET functions and they will need to be filled out so that they actually implement the correct SET and GET functionality.

---

4 Literally use the RFC as the input to the MIB complier. MIBs are defined in a standardized ASN.1 format that may be read by a program.

Automatic Protection Switching

Protection switching addresses the network aspect of availability rather than reliability. Reliability can be looked at as Mean Time Between Failures (MTBF). Availability is Mean Duration Of Failure (MDOF) or once a failure occurs, how quickly is network connectivity restored. Simply put, it is uptime. Availability is also described by “5 9's” or 99.999% up time. The motivation of implementing Automatic Protection Switching (APS) is that no matter the reliability of a circuit, even a short duration outage is very expensive and painful.

History

Backing up mission critical circuits or those with service agreements attached has always been a key network design consideration. Automatic fail-over to alternate facilities has been available for decades. Low speed (1200Kbps – 19.2 Kbps) lease-line analog modem circuits were backed up with switched (dial-up) analog circuits. The digital circuits (2,400 Kbps – 64 Kbps) were backed up first by switched analog circuits, by switched 4-wire, and then 2-wire digital service. Rarely, the back up was an idle leased line. As services like T1 became more common, the idle leased line method became more prevalent. These methods were often referred to as dial-restoral and hot stand-by, respectively.

Protection methods like these often meet the needs of the customers served. The down side is that they were all proprietary to a single equipment provider. Mixing of different manufacturer’s equipment was impossible. Changing vendors meant not only forklift changes of hardware, but also “forklift” changes in operations manuals and retraining of entire staffs. This situation still exists today when considering technologies like T1, E1, T3 and E3. The problem is that there are no open standards available to encourage uniform approaches to protection switching to achieve interoperability and compatibility between equipment vendors.

The newest physical layer WAN technologies are beginning to change that. Both SONET and SDH have specific standards for the operation of protection. These are broadly called

Automatic Protection Switching (APS), and sometimes Multiplexed Protection Switching (MPS) for SDH. Since SONET/SDH (OCs or Optical Carriers) can be configured in point-to-point and ring network architectures with different needs, different standards and techniques are applied to each.

Description

APS is often used to describe two different kinds of protection switching. One is equipment protection. In the event of a piece of hardware failing, another piece is switched in to restore service.

A second type of protection switching protects from facility or fiber/coax/copper failure. Should the transport medium become severed or otherwise compromised, a mechanism is put in place to supply an alternate physical path. This is what the standards define.

The methods to achieve equipment and facility protection are different. The standards only address facility APS. It is VERY important for the marketing and engineering teams to keep these two objectives clear and distinct. We often see them confused and mixed with each other during design discussions.

WarmStart Capability

Before you finalize your design specification, or better yet, your Statement of Market Requirements, the ability to upgrade software running on the host processor or to reboot the processor should be considered. Sometimes referred to as “warm restart,” NComm calls this function WarmStart. This capability significantly improves equipment uptime, because user traffic continues to be processed even when the host or the host software is not running. Until fairly recently, this was not a requirement in traditional telephony equipment.

One common practice in place of WarmStart is the use of dual memory banks. The equipment runs in one bank while the second was being loaded with the new software.

Once the new software is resident locally, the equipment would be brought down and then re-initialized with the new code. Dual banks can also be used as backup in the case of software corruption. Both in cases of software upgrades and unexpected corruption, this method reduces downtime, but still results in service disruptions, including the abrupt termination of any calls in progress. Further, depending on the size of the memory banks, duplicating them may add significant additional cost to the product.

A second method for addressing the problem is to use redundant equipment. In this scenario, an active system carries the traffic while an inactive protection system stands ready to take over for the active system if it fails. When a software upgrade is required, the protection system is upgraded to the new software first. After that upgrade is complete, the switch occurs and the protection system takes over for the active system. The previously active system is then upgraded. Using redundant equipment can be expensive, since it requires twice as much hardware. In addition, the new software must be designed so that it can interact properly with the older version in terms of the protection switch. If, for some reason, the older version either is not compatible with, or cannot be made compatible with the new version, this method may fail to accomplish the restart goals.

WarmStart takes the software reloading process to a new level. There are parts of a system (processing user traffic) that can continue autonomously without the operating system and embedded software running. WarmStart creates an environment where this can take place and provides the process necessary to execute the software reload and bring up in a way that is transparent to existing calls and traffic. It is important to note that during Warm Start, overhead functions such as alarms and performance messages will not be processed, nor will new calls be set up. However, established calls and data traffic will not be disrupted.

System Behavior with WarmStart

When considering the overall availability or “uptime” of a communication system within a network, that you must take into consideration the anticipated failure rate of the host

processor software. This is in addition to the failure rates of the host processor hardware and the devices processing the user traffic (framers, mappers, transceivers, network processors, etc.).

In typical WAN systems, the host processor is an off-the-shelf microprocessor that is responsible for configuration and control of the network processor. The host processor is typically loaded with an operating system that runs a program containing an API that controls the data traffic processing elements such as framers, transceivers, mappers, network processors, switching devices, and application software.

Given that software failure rates tend to be higher than hardware failure rates, systems designed for high availability must be able to recover from host processor software failures (unplanned host processor outages caused, for example, by divide by zero traps or watchdog timer expirations) without having to reset or otherwise interrupt the flow of user traffic through the data processing elements.

Similarly, it also important to eliminate or minimize user traffic disruption during planned host software upgrades. In other words, it is imperative to be able to reboot the host processor without having to reset the line card or network interface card or any device on it. The following figure depicts the system behavior before, during and after the host reset. Agere semiconductors are used as an example.

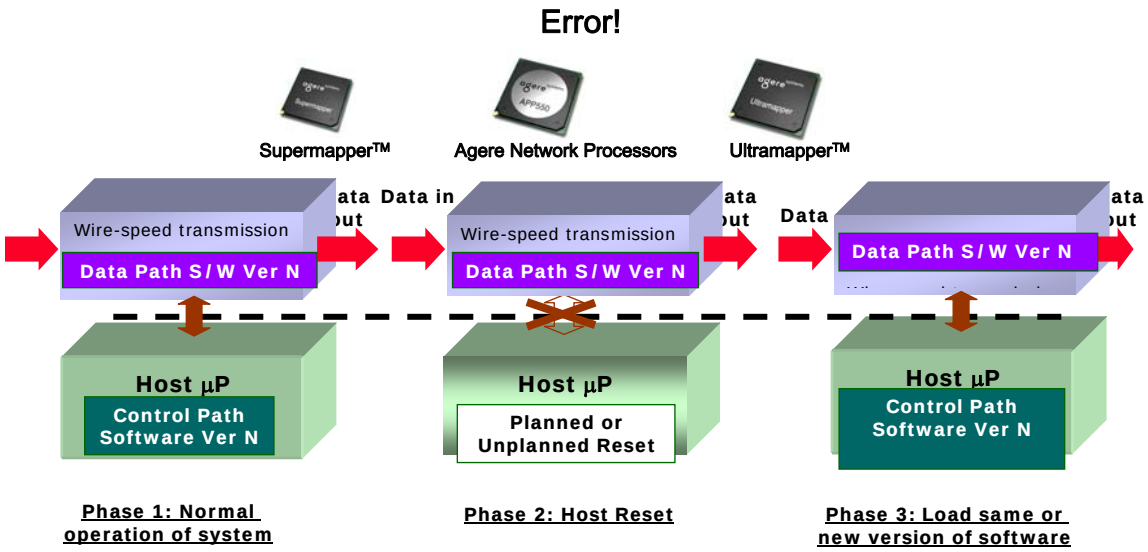


Figure 3. System Behavior during WarmStart

WarmStart Implementation Objectives

WarmStart is not specified in any standard. It is up to the equipment vendors to design specific implementations. However, there are general objectives that suggest where the design work needs to be done. Depending on the hardware and software architectures, there may be more or less of the system capable of taking advantage of WarmStart. The objectives of any WarmStart-enabled product include:

- Maintain user traffic flowing through the equipment
- Keep established calls
- Re-establish the state of the system prior to the initiation of the WarmStart sequence
- Adjust the state of the system for any changes that took place during the WarmStart sequence

Several hardware and software design considerations need to be examined to insure that the WarmStart function can be implemented. Not looking at these prior to design can make it difficult or impossible to retrofit the feature. Most importantly, the portion of the hardware

processing user traffic needs to be able to run autonomously. If the data processing hardware cannot run on its own, WarmStart cannot be implemented. Most, if not all, framers and transceivers should be able to support WarmStart. Similarly, some network processors employed for protocol processing and traffic management can also support WarmStart. The entire hardware architecture needs to be planned to function correctly. In the following sections we address WarmStart for framer and transceiver devices, as well as for network processors.

Framer and Transceiver Devices

In the case of WAN interfaces, this involves the framer or transceiver device. We believe that most framers can be used in a WarmStart system. However, it is best to either verify in-house that your choice of devices is WarmStart-capable, or have NComm to verify it for you. Early verification mitigates the risk and costs associated with surprises late in the development cycle.

Generally, once you assemble the appropriate hardware and software, the operation, seen at a high level, is simple. A snapshot is taken of the configuration and state of the framer device(s). Then, the framer is isolated (or disconnected) from the normal initialization sequence that would reset the device. New software is loaded and brought on-line without touching the framer operation and assumes that the configuration and state information have not changed. Then, the assumed and actual configuration and state are compared and reconciled to current. The software is fully interfaced to the framer device and a successful WarmStart has been achieved.

Designing for High Availability

Systems designed for high availability typically require software that enables the WarmStart capability. Some device vendors and third party software providers supporting a device provide software that the system developer can use to implement the capability in

the application layer software. The software usually will have some sort of an API to make integration easier.

Vendors like Agere provide such software for their network processors (NP) that include support for the WarmStart feature. A key element of the design includes the ability to recover the configuration state that is lost when the host processor is reset. In the case of Agere, the NP WarmStart design uses non-volatile host memory (NVM), where NVM is defined as a block of host DRAM whose contents are preserved across a host processor reset and/or reboot.

Assuming that DRAM contents are unchanged when the host processor is reset, NVM can simply be a block of reserved host DRAM, not included in the general memory pool. Note that the NVM must be reserved so that it is not cleared or allocated for other purposes when the operating system reboots. This principle holds in most, if not all WarmStart implementations. For further details regarding the Agere API, please refer to the RTE/API Reference Guide for the specific Agere network processor.

> There is a definite trend towards making WarmStart a requirement in both data and telephony equipment. For companies that do not currently provide this feature, it is time to start planning to include this critical feature in next releases of product.

Ethernet in the Wide Area Network

This represents a new area of focus as the application line between LAN and WAN technologies continues to blur. Service and equipment providers are beginning to experience the same challenges WAN has faced for yours as well as some unique to Ethernet.

(The following was published in Network Systems Designline (December 19, 2005)

HOW-TO Overcome Ethernet OAM challenges in the WAN

Although Ethernet equipment without operations, administration, and maintenance (OAM) functionality exists in mission-critical networks, embedded OAM is gaining as Ethernet plays a larger role in the WAN. One challenge lies in achieving the same level of OAM support that users receive with traditional carrier networks. Here's how to overcome the challenges of implementing OAM in the WAN.

Ethernet is beginning to join and even take the place of traditional wide-area networking (WAN) technologies while, simultaneously, WAN technologies are applied to LAN applications to increase the reach of what were once simple, distance-limited local networks. The result is today's "super-LAN."

Carrier networks are known for sophisticated and effective management capabilities, and set a high standard for reliability, availability, fast failover, and recovery to ensure that service level agreements (SLAs) made by providers can be met.

One of the greatest challenges facing service providers as they deploy Ethernet-based solutions in WAN environments, lies in achieving the same level of operations, administration, maintenance and provisioning (OAM&P) support that users are accustomed to receiving with traditional carrier networks. Specifically, service level agreements attached to specific services need to be met without regard to the underlying technologies used to provide them. Ethernet OAM is one set of the capabilities required to meet SLAs.

Until recently, OAM&P functions were not even defined for Ethernet implementations—and it is still unusual for even the OAM subset of these functions to be considered where Ethernet is concerned. However, standards bodies and major equipment manufacturers are increasingly addressing the issue. While optional today, soon Ethernet OAM will be a base-line requirement for equipment providers.

What is OAM&P?

The Network Dictionary defines Operation, Administration, Maintenance and Provisioning (OAM&P) as "a group of management functions that provides system or network fault indication, performance monitoring, security management, diagnostic functions, configuration and user provisioning."

- Operations: coordinating actions among administration, maintenance, and provisioning
- Administration: designing the network, processing orders, assigning addresses, tracking usage, and accounting
- Maintenance: isolating and repairing faults when malfunctions occur
- Provisioning: installing equipment, setting parameters, verifying that the service is operational, updating, and de-installation.[1]

Operations administration maintenance (OAM) refers to "managing and maintaining a network or network device. The P in 'OAM&P' adds 'provisioning' to the list, which is a telephone company term for setting up a service."[2]

The OAM subset of OAM&P omits the provisioning piece since it is implemented at the Application layer and involves such functionality as turning on services and subscriber billing data. For this discussion it is also less relevant, because engineering efforts associated with these applications are normally specified, designed and implemented by different groups. Yet capabilities built into lower layers will impact provisioning and will, in some cases, be critical for provisioning features to function correctly. One element builds on the other.

Whether OAM code is internally developed or outsourced, it is important for developers to understand the standards embedded in their products. The key standards for Ethernet OAM include:

- ITU-T Y.1730 – Ethernet-based networks and services
- IEEE 802.3ah – Ethernet link OAM

- IEEE 802.1ag (draft) – Ethernet Maintenance Domain Connectivity Fault Management
- ITU-T Y.1730 is a fairly short and general document. It sets the stage for the Ethernet OAM requirements. It does not specify how things are to be done, but does provide a quick view of what types of functions need to be implemented, as well as some of the underlying reasons for implementing the function. It provides the OAM objectives, but does not provide direction for interoperability.

IEEE 802.3ah and 802.1ag specify how OAM is to be implemented within the Ethernet network achieving the objectives of Ethernet OAM and interoperability. Interoperability in this case covers that between devices, as well as the co-existence of network and service providers *(See Figure 1)*.

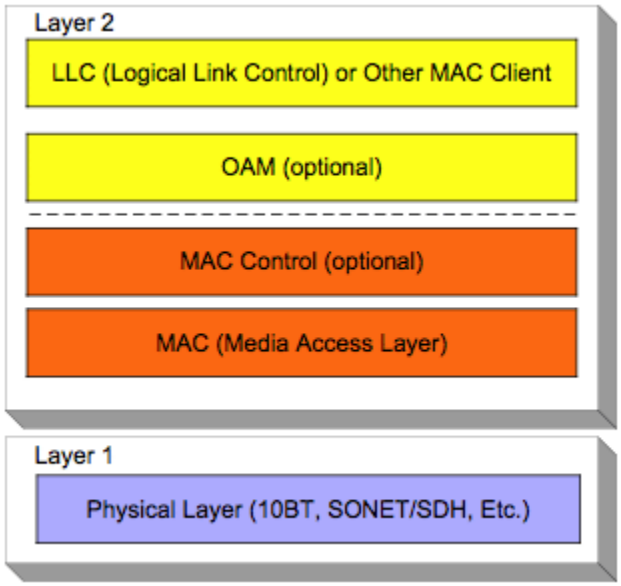


Figure 1. This figure is adapted from the IEEE 802.3ah figure 57-1

How the need for Ethernet OAM evolved

OAM is a fairly new concept in the world of Ethernet networks. Originally intended for relatively small, locally controlled data networks, there was minimal need for WAN-like performance monitoring in Ethernet applications. Over the last few years, however, LANs have transformed into metropolitan area networks (MANs) increasingly based on an

Ethernet that has matured to the point where it barely resembles the simple unmanaged protocol of its LAN beginnings.

As usage of Ethernet in networks began to grow both geographically and in complexity, the need for telco-like OAM capabilities became apparent. Now, as LAN technologies are taking the place of what was traditionally the WAN purview, they face all of the complexities of traditional WAN environments and even add some complexity of their own.

Added complexity

High-speed Ethernet networks often exist in environments with multiple end-user organizations, multiple networks and service providers, and multiple converged (voice-video-data) services running over the same physical and virtual networks. Increasingly, these Ethernet networks are becoming mission-critical.

Downtime cost

Currently, when the network goes down, business stops--customers cannot connect to make reservations or place orders either by phone or the Internet. Outbound telemarketing comes to a grinding halt. Security quotes become unavailable as does the ability to buy and sell stocks, and bonds, and option trading becomes problematic. Literally, millions of dollars is often lost during a brief outage.

Management complexity

WAN standards for T1/E1, T3/E3 and SONET/SDH already specify what capabilities a compliant circuit must have. New standards are under development by the IEEE and other standards bodies and vendor groups, to allow comparable management of large Ethernet networks being deployed across metropolitan areas and beyond. However, both requirements and solutions are foreign to many veteran LAN experts.

How Ethernet OAM addresses challenges

OAM is the set of tools or capabilities that enable the monitoring and quick restoration of a network in the case of failure. Given that the network is typically comprised of equipment with different owners and from many different manufacturers, OAM had to be specified by a standard to ensure consistency and interoperability.

OAM entities are network-aware in that they use information from, and provide information to other network entities. They cooperate, providing consistency and conformity critical to an entity's OAM functions.

As stated in IEEE P802.1ag/D4.1, Connectivity Fault Management (CFM) can be sub-divided into a number of categories that may seem obvious, but remember that in traditional Ethernet applications everything was managed quite well without them *(See Figure 2)*.

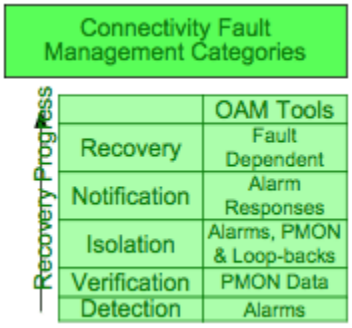


Figure 2. Connectivity Fault Management

Carrier-class fault management

The first CFM category is fault detection. To credibly provide mission critical services, it is advantageous for the providers to be aware of network problems before a user calls to report it. A fault can be a hard failure or deteriorating service. The root cause may or may not lie with the Layer 2 Ethernet, but the existence of the fault must be known. An alarm will likely be the source of detection.

Once a fault is detected, tools must be in place to verify that the fault actually exists and that it is of the nature first reported. For instance, the reported fault may just be the result of an underlying primary failure at a different point in the network. Or, perhaps the fault was totally spurious and has gone away. Without verification, achieved through the examination of PMON (Performance Monitoring), data, time, money and customer satisfaction are all at risk.

The root cause of the fault must be isolated through an examination of what is reported by the alarm and the PMON data, coupled with loop-back tests. Fault notification follows fault isolation to ensure that traffic is diverted to other routes while fault area is repaired and service is restored. One possible consequence of poor notification is that traffic may misinterpret troubleshooting and testing for full network recovery.

Fault recovery is the final step. Given the information collected as to nature and location of the fault, decisions can be made to determine the best method for resolving the fault. A truck roll may be needed, or if one of the indicators was the "dying gasp," a simple phone call to the site might get the equipment plugged back in or verify that a general power outage occurred. This level of troubleshooting and restoration/recovery requires management information not provided by traditional Ethernet networks. This is also where new OAM standards activities are focused. Let's next take a look at the broad categories of capabilities being added to traditional Ethernet.

PMON and alarms

The standards define two types of OAM information. One type, an alarm, provides an indication that a hard failure of some kind has occurred. The other type, performance monitoring or PMON, provides a warning that something is going wrong, but a hard failure has not yet occurred.

Alarm states are declared based on a particular defect persisting for a certain length of time. Defects "integrate" into alarms of a set period. A normal integration time in the WAN is 2.5 seconds. The reason that the integration time is non-zero is to account for momentary line "glitches," which occur from time to time.

Protocols at the layer of the defect, and at higher layers, are equipped to handle real-world hiccups, but when they persist, they become real problems and must be addressed. That is the purpose of integration time.

Problems inevitably occur. It is only when a fault such as an Out of Frame (OOF) condition integrates into an alarm that the receiver responds with an RAI (Remote Alarm Indication)

in the direction of the defect. The SNMP-based network management system is notified of the alarm condition, and action is taken to identify the root cause so network service can be restored.

Performance Monitoring (PMON) plays a different role. If alarms are idiot lights, then PMON are gauges. As specified by specific standards, performance data is collected in 15-minute buckets for either a 24- or 48-hour period. This information includes errored seconds, severely errored seconds, and so forth. By tracking and analyzing PMON information, negative trends can be identified and addressed before a hard failure occurs. PMON allows proactive and preemptive action to be taken.

One of the users of this PMON data is the Threshold Crossing Alert (TCA). TCAs can be set on a number of parameters that will generate an indication in the event that the specified threshold is exceeded. TCAs are used to provide notification as service is degrading, with the objective of initiating remedial action before a hard failure occurs.

Finally, there's the loop-back, which allows the received path to be "looped back" to the transmit path. Depending on the type of loop-back, packets may be terminated and re-created, or the loop-back might run the packets back untouched. This is necessary, since there can be source and destination addressing issues which must be dealt with. Remote loop-backs are very important in problem isolation. In a well-segmented network, systematic loop-backs can quickly and accurately pinpoint the fault.

Ethernet OAM mechanics

OAM was originally instituted in synchronous protocols using overhead bits within frames. The payload and the operations and administration overhead information were all contained in the same frame. Synchronous OAM information was transmitted in a very deterministic way, consistently appearing at specific and predictable points in time.

Ethernet, being an asynchronous protocol, requires a different approach. Payload and OAM information are separated. Slow Protocol frames called Protocol Data Units (PDUs) are used to provide both alarm and PMON information. At the Link layer, IEEE 802.3ah

calls these OAMPDUs. Spanning multiple links is the Maintenance Domain. IEEE 802.1ag (draft) specifies Connectivity Fault Management Protocol Data Units (CFMPDUs) at this level.

Information received in PDUs is parsed and processed according to the relevant standard. All of the capabilities discussed above flow from this information combined with local events. [Note: Although the information contained in this article reflects the latest information available at the time of writing, both standards and drafts are subject to change. Prior to specifying and designing Ethernet OAM functionality, developers should ensure that they are working with the latest updates.]

OAM: Today and tomorrow

There is already a great deal of Ethernet equipment without OAM functionality sitting in mission-critical networks. Recognizing this, the OAM discovery process has also been defined so that OAM-capable remote DTE (Data Terminal Equipment) can be identified and the OAM functions used by the network. DTE without these advanced features will be recognized for what they are and will not hinder or confuse the network. In other words, an environment will be established where equipment with and without OAM can co-exist in the same network.

Embedded OAM functions in CPE are becoming increasingly necessary as Ethernet technology begins to play a larger role in the wide-area network. Ethernet OAM functionality, while still "optional" from a standards perspective, is already certainly a de facto requirement in order to sell equipment to carriers and service providers. End-user customers need OAM implemented in CPE devices to ensure that they get quality service from their providers and to know if the Service Level Agreements (SLAs) they're paying for are being met.

Over the next decade, Ethernet will continue to augment or even replace traditional WAN technologies in fulfilling user needs for accurate and dependable voice and data services. These services will continue to be delivered alongside streaming video and other emerging

applications and across network segments owned by a variety of carriers and providers. And the need for increased OAM functionality will continue to grow, if Ethernet is to be considered a carrier-class transport and be a viable WAN technology.

Ethernet OAM Implementation

Operations, Administration, Maintenance and Provisioning (OAM&P) used to be the province of carriers Wide Area Network technologies. Telecom and Datacom equipment destined for the WAN was required to have standards compliant OAM&P functionality to insure both efficient manageability and interoperability between equipment vendors. LAN technologies originally designed for more limited geographic networks had no such requirements.

Things have changed. Now Ethernet is being deployed in almost every area of the network. As it increasingly finds itself in applications formerly filled by WAN technologies, the lack of OAM&P capabilities has become a challenge. To meet this challenge, new standards are being developed and implemented to allow carriers to employ this LAN technology while maintaining the traditional monitoring, problem isolation and diagnostic tools required for geographically dispersed networks.

For Carrier Grade Ethernet, OAM (Provisioning is controlled at a higher layer) must be a consideration for the vast majority of equipment manufacturers that incorporate this access. Whether developed in-house or in the form of purchases source code, Ethernet OAM functionality is within the reach of every equipment manufacturer.

Ethernet OAM Design Considerations

Following the lead of NComm's Trunk Management Software for WAN technologies, the ideal solution will be first and foremost standards compliant and flexible. Flexibility encompasses hardware and operating system independence. Employing well-defined APIs promotes faster first time integration and ease of reuse. The source language of choice is ANSI-C for ease of maintenance.

The following diagram illustrates offers one alternative of how the Ethernet OAM function fits into the existing technology. As can be seen, the implementation dependent portion is isolated in the driver and the entire OAM processing system exists in kernel or protected space for efficient processing (basic OAM processing needs to be in real time). Reporting and management functions that tend to be less time critical and processor intensive are placed in user space. Two APIs support system portability and flexibility; one is between the driver and OAM software and the other between the OAM software and the users' applications in user space.

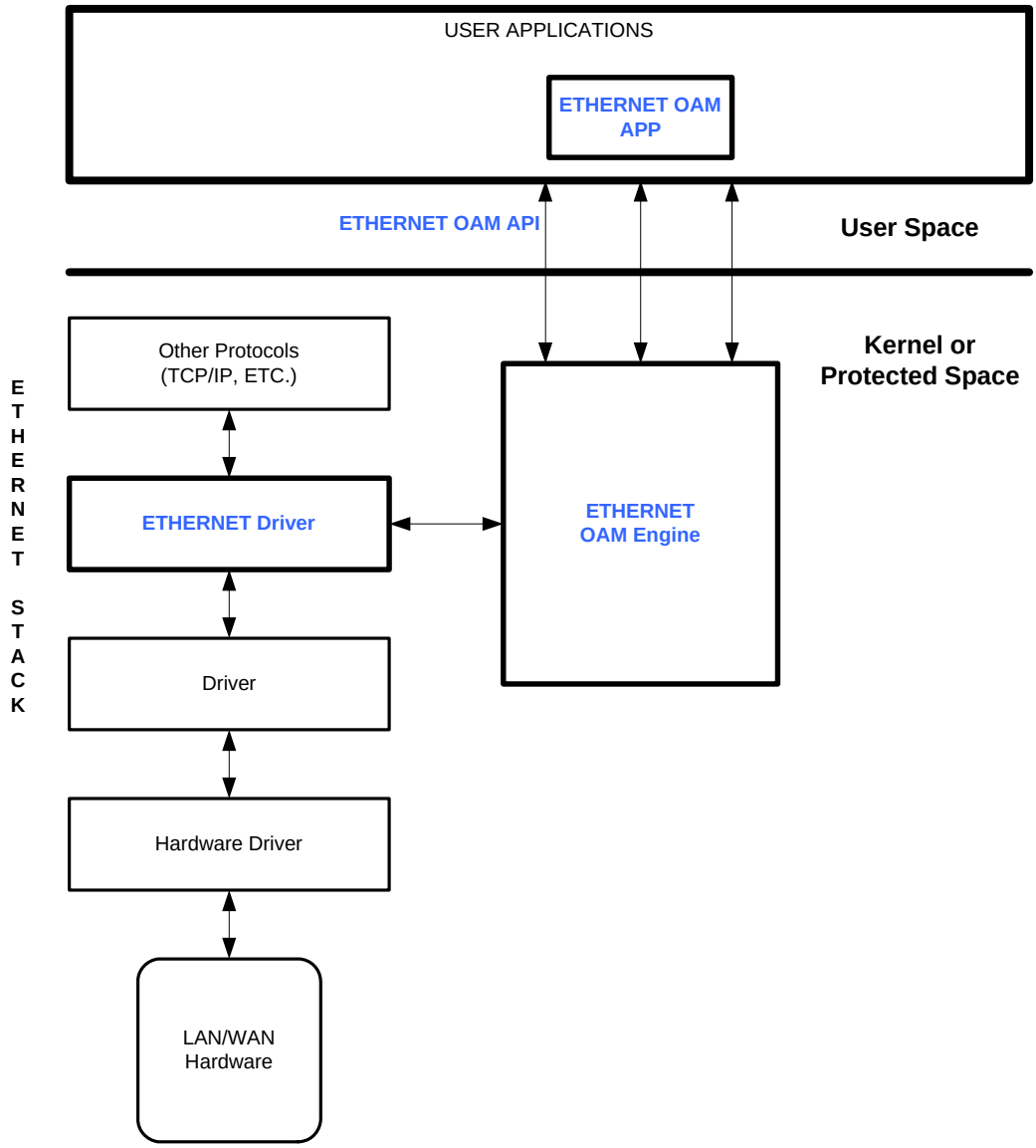


Figure 4. Driver and OAM Software

Ethernet OAM (EOAM) needs to conform to the relevant portions of:

- IEE 802.1ag
- IEE 802.2ah
- ITU Y.1730
- ITU Y.1731

EOAM Alarms (Fault Management)

The following functions are to be included as specified by the standards:

Continuity Check (ETH-CC)

The ETH-CC detects a loss of continuity (LOC) between two Maintenance Entity Group End Points (MEPs) in a Maintenance Entity Group (MEG). It also can detect unintended connectivity between two MEGs, within the MEG (unexpected MEP) as well as other defects. This check is useful for fault management, performance monitoring and protection switching.

Loopback (ETH-LB).

Unicast loopback is used to verify the bi-directional connectivity between a MEP, a Maintenance Entity Group Intermediate Point (MIP) or a peer MEP, or to perform in-service or out-of-service tests between pairs of MEPs.

Multicast loopbacks are used to verify bi-directional connectivity between a MEP with all of its peer MEPs. The result of a multicast loopback on a MEP is a list of all of its peer MEPs with which it has verifiable connectivity.

Link Trace (ETH-LT).

The Ethernet Link Trace is used for Adjacent Relation Retrieval and Fault Localization. When invoked, the relationship between the MEP and a remote MEP or MIP is revealed by the series of MIP MAC addresses until the target MEP/MIP is reached. In Fault Localization, the difference between the expected and retrieved series of MAC addresses provides information about the location of the fault.

Alarm Indication Signal (ETH-AIS)

EOAM will initiate the transmission of AIS in the presence of certain defect conditions and will take the appropriate actions on its reception. Transmission will be stopped and AIS state cleared following the nominal integration periods.

Remote Defect Indication (ETH-RDI)

RDI can be used only when the Continuity Check function is enabled. It is used both for fault management and performance monitoring. RDI is transmitted on the detection of a defect and cleared when the defect clears. RDI reception is cleared when all peer MEPs stop transmitting RDI.

Locked Signal (ETH-LCK)

The Ethernet Locked Signal is used to indicate that data traffic is intentionally being interrupted for administrative purposes like testing. It helps to differentiate between a defect condition and controlled suspension of traffic. EOAM handles both the transmission and reception of this signal with the appropriate integration periods.

Test Signal (ETH-TEST)

The Ethernet Test Signal function is used to perform one-way, in-service and out-of-service diagnostic tests.

Maintenance Communication Channel (ETH-MCC)

The Ethernet MCC is used as a communication channel between pairs of MEPs. The EOAM should provide general access to the MCC.

Vendor Specific OAM (ETH-VSP)

Vendor Specific OAM needs can be generically accommodated from a communications point of view. The actual meaning, processing and use of this are

left to individual equipment manufacturers requirements. No interoperability between equipment manufacturers is expected nor required in this area.

EOAM PMon Module (Performance Monitoring)

OAM functions for Performance Monitoring are currently only defined for point-to-point connections. The four parameters available for trending are Frame Loss Ratio, Frame Delay, Frame Delay Variation and Throughput. As appropriate, these measurements are collection for single- and dual-ended loss, and one- and two-way delay.

EOAM APS Module (Automatic Protection Switching)

Automatic Protection Switching for Ethernet is very similar to Linear APS for SONET/SDH. The 1+1 and 1:1 concepts are applied in unidirectional and bi-directional modes. Protection should be configurable for revertive and non-revertive operation. The primary difference is that instead of the K1 and K2 bytes that are used for SONET/SDH APS communication, packets are used. More detailed information can be found in G.8031/Y.1342

To execute such a development, there are many details of the standards and the design to be worked out. The foregoing is intended to provide some preliminary directions and important starting points. Keeping checking our website for the latest thinking and developments in this important area for Carrier Grade Ethernet equipment.

Specific Issues of T1/E1

Overview of T1

T1 provides a 1.544 MHz electrical interface. The T1 signal can carry channelized traffic or unchannelized traffic. The T1 signal consists of payload bits that are used to carry the data over the T1 line, and framing bits that are used to determine where the payload is located. In unchannelized applications, the payload bits carry data traffic such as frame relay or ATM. In channelized traffic, the payload is partitioned into timeslots and is used to carry voice traffic or call control such as ISDN or SS7.

The T1 signal consists of a time-multiplexed frame with one framing bit and 192 payload bits as shown in the following diagram. In unchannelized applications, the payload will consist of a stream of bits. In channelized T1 applications, the payload will be divided into 24, 8-bit timeslots.

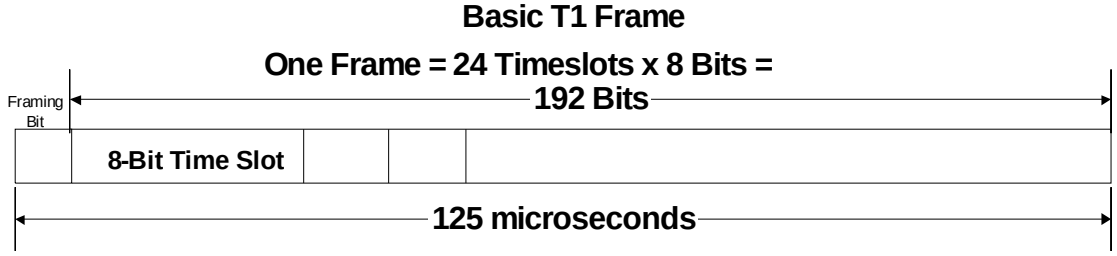


Figure 5. Basic T1 Frame

The T1 frame is repeated every 125 microseconds, which leads to the frequency of 1.544Mhz (193/0.000125). There are three main types of framing which are present on T1:

1. Super Frame format – also known as D4
2. Extended Super Frame – also known as ESF

3. SLC-96 or TR-008 Framing format

These different framing formats all use the same basic T1 frame, but the definition of the framing bit is different and will be described later.

Alarms

Alarms are used to detect and notify maintenance personnel of problems on the T1. There are three types of alarms:

1. RED alarms
2. BLUE alarms also known as Alarm Indication Signal (AIS)
3. YELLOW alarms also known as Remote Alarm Indication (RAI)

Alarms are created from defects. Defects are momentary impairments present on the trunk or line. If a defect is present for a sufficient amount of time (the integration time), then the defect becomes an alarm. Once an alarm is declared, the alarm is present until after the defect clears for a sufficient period of time. The time it takes to clear is called the de-integration time. The table below shows the defects, the alarms and the typical integration and de-integration times for T1 per ANSI T1.231.

Defect	Alarm	Integration Time	De-Integration Time
Loss of Signal	RED	2.5 Seconds	10 Seconds
Loss of Frame			
Remote Alarm Indication (RAI)	YELLOW	0.5 Seconds	0.5 Seconds
Alarm Indication Signal (AIS)	BLUE	2.5 Seconds	10 Seconds

Framing

The different framing formats carry the alarm information differently. To understand this, we need to look at the details of the framing formats. As indicated before, the Framing bit in a T1 frame repeated every 125 microseconds in the 193rd bit. The framing bit position consists of two types of bits, the Terminal Framing (Ft) and Signaling Framing (Fs) bits. In SF and SLC-96, the Ft bits are the same – a repeating 0, 1, 0, 1, 0, 1 pattern while the Fs bits are different.

Super Frame Framing

In Super Frame Framing, the framing patterns is as follows:

Frame	1	2	3	4	5	6	7	8	9	10	11	12
Fs		0		0		1		1		1		0
Ft	1		0		1		0		1		0	

Figure 6. T1 Super Frame

In Super Frame Framing, frame number 6 and frame number 12 are signaling frames. In channelized T1 applications using robbed-bit signaling, these frames are used to contain the signaling information. In frame numbers 6 and 12, the least significant bit of all 24 timeslots is “robbed” to carry call state information. The bit in frame 6 is called the A bit and the bit in frame 12 is called the B bit. The combination of AB defines the state of the call for the timeslot that these two bits are located in.

Extended Super Frame Framing

Extended Super Frame (ESF) framing is similar to Super Frame except that the super frame has been “extended” to 24 frames instead of 12 frames. In addition, the advancements in technology have eliminated the need to have a framing bit every 193rd

bit. With ESF, the framing bit occurs once every 772 bits (4 frames) as shown in the FPS position below:

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
FPS				0				0				1				0				1				1
FDL	M		M		M		M		M		M		M		M		M		M		M		M	
CRC		C1				C2				C3				C4				C5				C6		

Figure 7. Extended Super Frame

The other bit positions are used for the Facility Data Link (FDL) and a CRC-6 check sum.

The FDL is used as a point-to-point link between the customer premise and the network and is used for facility maintenance functions. The FDL does not pass through the network. That is, once a local T1, between the network and the customer premise, connects to the network, the FDL is terminated.

The FDL carries two types of traffic:

1. Bit Oriented Codes (BOC)
2. High-level Data Link Control (HDLC) Packets

BOC codes are repeated 16-bit long Binary ”111111110CCCCC0” sequences where “CCCCCC” is the BOC command word. BOC codes are used to control loopbacks, to indicate timing synchronizations source, to indicate Yellow Alarms, etc. To send a valid BOC sequence, it must be present on the T1 line for a minimum of 10 repetitions. Most recognition algorithms will recognize a BOC sequence if it receive 7 valid sequences out of 10.

The other type of traffic on the FDL is HDLC packets. Two standards cover the HDLC packets carried on the FDL.

1. ANSI T1.403 – This standard requires that, once per second, a packet is transmitted that contains performance data representing performance data that the receiver is detecting. Four seconds of information is transmitted so that recovery operations may be initiated in case an error corrupts a packet.
2. AT&T TR-54016 – This standard contains requirements for monitoring the performance of the T1. Once the performance data is collected, it can be retrieved via the FDL from the far end via a command-response protocol.

The last item carried in the framing bit position is the CRC-6 checksum. The CRC-6 pattern contains the checksum over the previous frame. It allows bit errors on the T1 to be detected.

In Extended Super Frame framing, frame number 6, 12, 18 and 24 are signaling frames. In channelized T1 applications using robbed-bit signaling, these frames are used to contain the signaling information. In frame numbers 6, 12, 18 and 24, the least significant bit of all 24 timeslots is “robbed” to carry call state information. The bit in frame 6 is called the A bit and the bit in frame 12 is called the B bit, the bit in frame 18 is called the C bit, and the bit in the 24 frame is called the D bit. The combination of ABCD defines the state of the call for the timeslot that these four bits are located in.

SLC-96 Framing

AT&T invented SLC-96 framing for their SLC-96 product. SLC-96 is also generally known as TR008. We will use the term TR008 to describe our product features. The detailed description of SLC-96 can be found in the Telcordia Document GR-8-CORE. The purpose of the SLC-96 product was to provide standard telephone service (POTS e.g., Plain Old Telephone Service) in areas of high subscriber density, but back-haul the traffic over T1 facilities. To support the equipment, which is likely in an underground location, the T1s needed methods to provide:

1. Indications of equipment failure to maintenance personnel
2. Indications of failures of the POTS lines

3. Testing the POTS lines
4. Redundancy on the T1s

The manner that SLC-96 framing supports these features is using the framing bit position in the SLC-96 frame. The SLC-96 Super Frame, which is 72 frames long follows:

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Fs		0		0		0		1		1		1		0		0		0		1		1		1
Ft	1		0		1		0		1		0		1		0		1		0		1		0	

Frame	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Fs		C1		C2		C3		C4		C5		C6		C7		C8		C9		C10		C11		0
Ft	1		0		1		0		1		0		1		0		1		0		1		0	

Frame	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
Fs		1		0		M1		M2		M3		A1		A2		S1		S2		S3		S4		1
Ft	1		0		1		0		1		0		1		0		1		0		1		0	

Figure 8. SLC-96 Frame

The C1-C11 bits are the concentration bits. These bits are used to map POTS lines to timeslots on the T1 especially in oversubscribed conditions. That is, when more POTS lines are provided than can be carried on the T1s.

The M1-M3 bits are used for maintenance activities.

The A1-A2 bits are used for conveying alarm information from the remote device.

The S1-S4 bits are used for controlling protection switching.

In SLC-96 framing, frame number 6 and frame number 12 are signaling frames and every set of 12 frames there after. In channelized T1 applications using robbed-bit signaling, these frames are used to contain the signaling information. In frame numbers 6 and 12, the least significant bit of all 24 timeslots is “robbed” to carry call state information. The bit in frame 6 is called the A bit and the bit in frame 12 is called the B bit. The combination of AB defines the state of the call for timeslot that these two bits are located.

In-band Loopback Activation and De-Activation

Loopbacks are used to test T1 lines. To support testing, an in-band loopback is used to place the T1 in remote, also known as line, loopback. A remote loopback causes the bits received on the T1 to be looped, un-modified, back to its source.

Sending the loopback pattern activates an in-band loopback. The pattern must be sent for at least 5 seconds. The pattern overwrites the entire payload in the T1, thus corrupting any calls or data traffic. The framing bit may or may not still be present. The loopback is invoked when the pattern is removed.

The loopback is torn down when an in-band loop down pattern is transmitted for a period of 5 seconds. Of course, the times mentioned in this section are the nominal times per ANSI T1.403 but may be changed in different installations.

Signaling

Signaling is how calls are passed on the T1 facility via the signaling bits. There are signaling bits in both the receive and the transmit direction. These bits described the state of a call on the timeslot.

In ESF framing, the ABCD bits are used, while in SF and SLC-96 framing the AB bits are used. So, how many different call states can you potentially have with two bits, AB? The obvious answer is 4. However, this is not correct. As it turns out, there are potentially 9 different call states. The way this is done is using the concept of tri-level signaling. If we

look just at the A bit, it can be a 0 or a 1 but it can also be a toggle. A toggle is when in one super frame the bit is a 0 while in the next it is a 1 and it toggles every other super frame. Thus, the A bit can have 3 different states, which is termed tri-level signaling. In SF and SLC-96, a call can have 9 different states in each direction. ESF framing does not use tri-level signaling and has 16 possible states.

The method to determine how to interpret the signaling bits depends upon the call model being used. ANSI T1.403, AT&T PUB 43601, GR-303, and GR-8 all define different call models to interpret the signaling bits. The call models defined by these standards include the following

- Loop start
- Loop start with RLCF
- Ground Start
- Ground Start with RLCF
- Loop-Reverse Battery Signaling
- Network provided reverse battery signaling
- E & M Signaling
- Customer-installation-provided loop-start supervision (FXS/FXO)
- Private line auto ring
- Ring down
- Superimposed Ringing Multiparty
- Direct Inward Dialing Dial Pulse Terminating
- Frequency Selective Ringing Multiparty
- Single Party
- Superimposed Ringing Multiparty
- Universal Voice Grade
- Coin CF/DTF
- Multiparty Signaling

Channel Assisted Signaling (CAS) is the original signaling system used by E1 and provides 4 signaling bits for every channel. In CAS, channel 16 is reserved for signaling and the A/B/C/D bits for each channel are divided among 16 frames. Frame 0 contains the alignment signal, alarm, and spare bits. Frame 1 contains the A/B/C/D bits for channel 1 in the upper half of the channel and the A/B/C/D bits for channel 16 in the lower half. The remaining 14 frames follow the frame 1 format accordingly. In recent years, the term Robbed-Bit Signaling (RBS) has been replaced by CAS, which is now often used to refer to bits that are associated with a specific channel whether it is in the T1 or E1 format.

Overview of E1

E1 provides a 2.048 MHz electrical interface. The E1 signal can carry channelized traffic or unchannelized traffic. The E1 signal consists of payload bits that are used to carry the data over the E1 line, and framing bits that are used to determine where the payload is located. In unchannelized applications, the payload bits are used to carry data traffic such as frame relay or ATM. In channelized traffic, the payload is partitioned into timeslots and is used to carry voice traffic or call control such as ISDN or SS7.

The E1 signal consists of a time-multiplexed frame with 8 framing bits and up to 248 payload bits as shown in the following diagram. In unchannelized applications, the payload will consist of a stream of bits. In channelized E1 applications, the payload will be divided into up to 31, 8-bit time slots and one framing time slot.

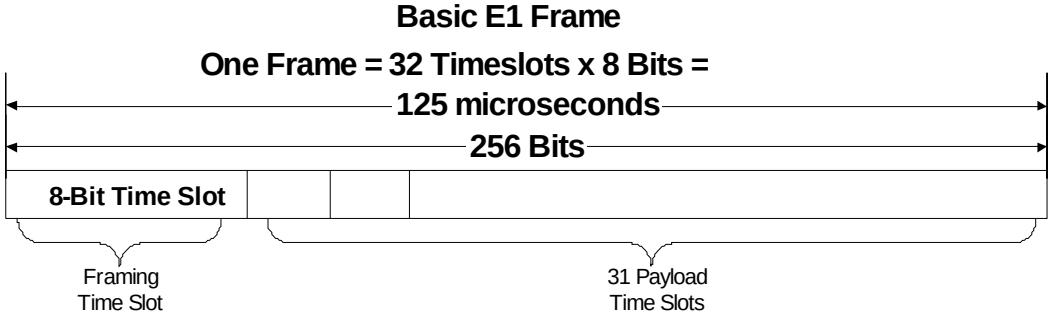


Figure 9. Basic E1 Frame

The E1 frame is repeated every 125 microseconds, which leads to the frequency of 2.048 MHz (256/0.000125). Four types of framing are present on E1:

1. Basic E1 Framing
2. E1 Framing with Signaling Multi-Frame Alignment.
3. E1 Framing with CRC4 Multi-Frame Alignment.
4. E1 Framing with CRC4 Multi-Frame Alignment and Signaling Multi-Frame Alignment.

These different framing formats all use the same basic E1 frame, however the differences in the framing sequence are described below.

Framing

E1 framing consists of a dual frame pattern. The first frame pattern, shown below as frame 0, contains the Frame Align Signal (FAS). The FAS is what is used to find the remaining parts of the E1 frame. The second frame pattern contains the Non-Frame Align Signal (NFAS).

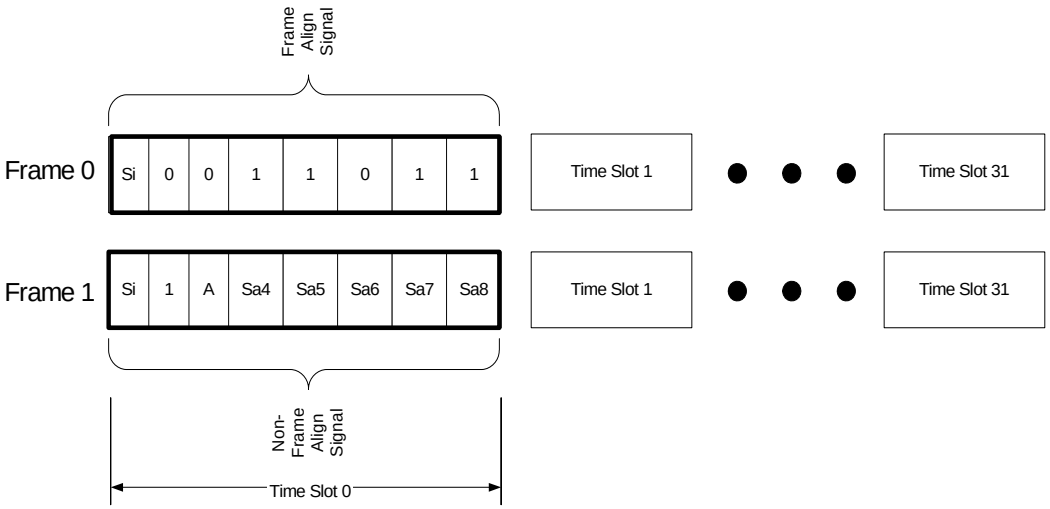


Figure 10. E1 Frame Pattern

The first time slot in each frame is used to provide frame detection. In E1, an entire time slot is dedicated to framing information as well as other information. In the FAS frame, time slot 0 contains the Si (international) bit and the bit pattern 0011011. An E1 framer will look for this bit pattern to establish basic frame alignment.

In addition to the FAS pattern, the NFAS will be checked to detect the Non-Frame Alignment Signal. The “1” in the NFAS is used to validate the NFAS signal. The NFAS also contains the Si bit, the A-bit and five Sa bits. The A bit is used to indicate the Remote Alarm Indication (RAI) to the far end. When the A-bit is a 1, the RAI is asserted and when

the A-bit is a 0 the RAI is not asserted. The Si bit is the international bit. Its use is reserved for crossing international boundaries. In most cases, the Si bit will be set to a one. There are five Sa bits, Sa4 through Sa8. The Sa bits are the national bits and are nominally set to all 1s when not used. When used, the Sa bits are for synchronization status messages, loop back requests, and other uses.

E1 Framing with Signaling Multi-Frame Framing

The E1 Framing with Signaling Multi-Frame framing extends the E1 frame from a two-frame sequence to a 16-frame sequence as shown in the following diagram.

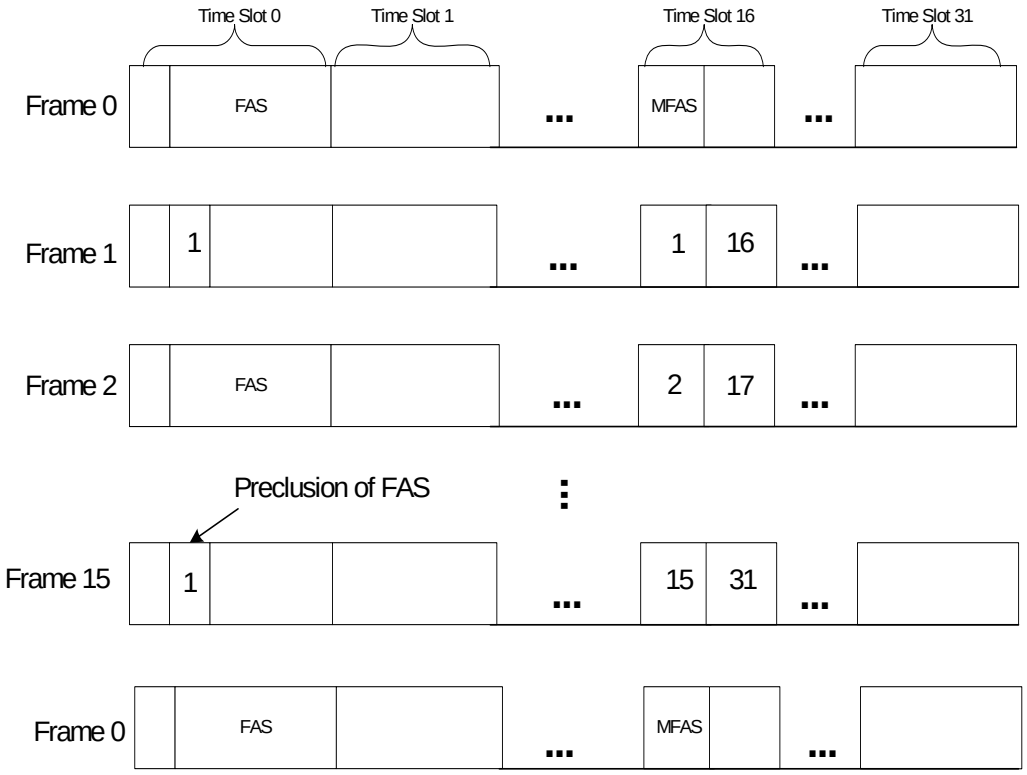


Figure 11. E1 Multi-Frame

The E1 frame is extended to 16 frames by the use of time slot 16 to carry the CAS signaling information. The first frame, named “Frame 0” above, is used to indicate the start of the 16-frame sequence. In this frame, time slot 16 contains the Multi-Frame Alignment

Signal (MFAS) that consists of a 4-bit pattern of all zeros in the first four bits of the time slot. The second 4 bits contain the three Extra Bits (X-bits) and the Multi-Frame Remote Alarm Indicator bit (Y-bit). When E1 is used to carry Channel Associated Signaling, it must use one of the E1 frame formats that contains the MFAS. Because of the use of all zeros as the Multi-Frame Alignment Signal, no other entry in time slot 16 can contain all zeros.

In addition to carrying the MFAS, time slot 16 also carries the Channel Associated Signaling bits for call processing. There are 4 bits for each voice channel and are grouped two sets of 4 bits in each timeslot. The 8 bits continued in Frame 1 correspond to the signaling bits for timeslot 1 and timeslot 17. The 8 bits contained in Frame 2 correspond to the signaling bits for timeslot 2 and timeslot 18. This pattern continues for all 30-voice channels.

E1 Framing with CRC Multi-Frame Framing

In the E1 Framing with CRC Multi-Frame the Si bits are re-defined and are used to carry a CRC-4 checksum pattern. The 16-frame Multi-Frame is divided into two Sub Multi-Frames (I and II) and the alignment is done via the CRC frame alignment sequence in the first bit. In Sub Multi-Frame I, this sequence is the pattern B"0001" and in Sub Multi-Frame II, the sequence is B"011", with both patterns located in the Si bit position of the frame. In addition to the CRC alignment bits, the first bit position contains a CRC-4 checksum and E-bits. This is shown in the chart below:

	Sub Multi-Frame	Frame Number	Bits 1 to 8 of the Frame							
			1	2	3	4	5	6	7	8
Multi-Frame	I	0	C_1	0	0	1	1	0	1	1
		1	0	1	A	S_{a4}	S_{a5}	S_{a61}	S_{a7}	S_{a8}
		2	C_2	0	0	1	1	0	1	1
		3	0	1	A	S_{a4}	S_{a5}	S_{a62}	S_{a7}	S_{a8}
		4	C_3	0	0	1	1	0	1	1
		5	1	1	A	S_{a4}	S_{a5}	S_{a63}	S_{a7}	S_{a8}
		6	C_4	0	0	1	1	0	1	1
		7	0	1	A	S_{a4}	S_{a5}	S_{a64}	S_{a7}	S_{a8}
	II	8	C_1	0	0	1	1	0	1	1
		9	1	1	A	S_{a4}	S_{a5}	S_{a61}	S_{a7}	S_{a8}
		10	C_2	0	0	1	1	0	1	1
		11	1	1	A	S_{a4}	S_{a5}	S_{a62}	S_{a7}	S_{a8}
		12	C_3	0	0	1	1	0	1	1
		13	E*	1	A	S_{a4}	S_{a5}	S_{a63}	S_{a7}	S_{a8}
		14	C_4	0	0	1	1	0	1	1
		15	E*	1	A	S_{a4}	S_{a5}	S_{a64}	S_{a7}	S_{a8}

Figure 12. E1 Framing with CRC-4 Multi-Frame

The CRC-4 checksum bit contains the CRC-4 checksum of the previous CRC-4 Sub Multi-Frame. The E-bits are used to inform the far end of received CRC-4 errors. When a bad CRC-4 Sub Multi-Frame is received, an E-bit in the reverse direction is set to indicate the error. If both CRC-4 Sub Multi-Frames are in error, both E-bits are set. Using the CRC-4 errors and E-bits, both ends can determine which direction has difficulties in passing traffic error free.

When using the CRC-4 Multi-Frame, the Sa bits also have expanded functionality. Instead of being only one bit, each Sa bit becomes 8 bits in the CRC-4 Multi-Frame. One use of the expanded Sa bits is to carry the Synchronization Status Message. Since the SSM is only 4-bits long, the Four Sa bits are repeated in each Sub Multi-Frame. This is shown in the chart below:

S_{an1}, S_{an2}, S_{an3}, S_{an4} **n = Sub-Frame** **$_a$ = 4, 5, 6, 7, 8** **Depending upon the network**	**Synchronization Quality Level (QL) description**
0000	Quality unknown (existing synchronization network)
0001	Reserved
0010	See ITU G.811
0011	Reserved
0100	SSU-A - See G.812
0101	Reserved
0110	Reserved
0111	Reserved
1000	SSU-B – See G.812
1001	Reserved
1010	Reserved
1011	Synchronous Equipment Timing Source (SETS)
1100	Reserved
1101	Reserved
1110	Reserved
1111	Do not use for synchronization

E1 Framing with CRC Multi-Frame and Signaling Multi-Frame Framing

The E1 Framing with CRC Multi-Frame and Signaling Multi-Frame combine both the CRC-4 multi-frame as well as the Signaling Multi-Frame. Although both Multi-Frame signaling are 16 frames long, they are not necessarily aligned with each other. When both framing schemes are present, the features associated with each are available.

Alarms

Alarms are used to detect and notify maintenance personnel of problems on the E1. The alarms present in E1 are very similar to those of T1. These alarms are defined below:

1. Loss of Signal (LOS) alarms
2. Loss of Frame (LOF) alarms
3. Alarm Indication Signal (AIS) alarms
4. Remote Alarm Indication (RAI) alarms

Alarms are created from defects. Defects are momentary impairments present on the trunk or line. If a defect is present for a sufficient amount of time (the integration time), then the defect becomes an alarm. Once an alarm is declared, the alarm is present until after the defect clears for a sufficient period of time. The time it takes to clear is called the de-integration time. The table below shows the defects, the alarms and the default integration and de-integration times for E1. The times selected for E1 are the same as the times for T1 since there is no specific time specified for E1 integration and de-integration timers.

Defect	Alarm	Integration Time	De-Integration Time
Loss of Signal	LOS	2.5 Seconds	10 Seconds
Loss of Frame	LOF	2.5 Seconds	10 Seconds
Remote Alarm Indication (RAI)	RAI	0.5 Seconds	0.5 Seconds
Alarm Indication Signal (AIS)	AIS	2.5 Seconds	10 Seconds

A note should be made about the Loss of Frame defect in E1. With the four different types of framing in E1, the Loss of Frame defect is a composite of the defects associated with each framing component as shown in the following table:

	Basic E1 Frame	**E1 with Signaling Multi-Frame**	**E1 with CRC-4 Multi-Frame**	**E1 with CRC-4 Multi-Frame and Signaling Multi-Frame**
Loss of Basic Frame Alignment	Used to detect LOF defect	Used to detect LOF defect	Used to detect LOF defect	Used to detect LOF defect
Loss of CRC-4 Frame Alignment			Used to detect LOF defect	Used to detect LOF defect
Loss of Signaling Frame Alignment		Used to detect LOF defect		Used to detect LOF defect

Signaling

Signaling is how calls are passed on the E1 facility. This is done via the signaling bits indicated in the two E1 framing formats that have Signaling Multi-Frame. Signaling bits are transferred in both the receive direction and the transmit direction. Unlike T1 where the

signaling bits describe the state of the call, interpreting the signaling bits in E1 depends upon the previous state of both the receive and transmit direction.

The method to determine how to interpret the signaling bits depends upon the call model being used. ITU specification Q.422 defines the call model to interpret the signaling bits in E1. As an alternative to CAS signaling, ISDN and SS7 may be used to place phone calls over an E1 facility. Typically, the timeslot used for carrying ISDN or SS7 is also time slot 16. Consequently, basic E1 or E1 with CRC-4 Multi-Frame framing schemes must be used if ISDN or SS7 is to be carried.

Standards Requirements

The software must handle the requirements of:

- ANSI T1.231, Telecommunications - Digital Hierarchy - Layer 1 In-Service Digital Transmission Performance Monitoring
- ANSI T1.403, Telecommunications - Network and Customer Installation Interfaces - DS1 Electrical Interface
- ANSI T1.408/T1.403.01, Telecommunications - Network and Customer Installation Interfaces - ISDN Primary Rate Layer 1 Electrical Interface Specification
- ATT TR-54016, Technical Reference Requirements For Interfacing Digital Terminal Equipment To Services Employing The Extended Superframe Format
- ITU-T G.703, Series G: Transmission System And Media, Digital Systems And Networks; Digital Transmission Systems - Terminal Equipment - General; Physical/Electrical Characteristics Of Hierarchical Digital Interfaces
- ITU-T G.704, Series G: Transmission Systems And Media, Digital Systems And Networks; Digital Transmission Systems - Terminal Equipment- General; Synchronous Frame Structures And Used At 1544, 6312, 2048 and 44 736 Kbit/S Hierarchical Levels

- ITU-T G.826, Series G: Transmission And Media, Digital Systems And Networks; Digital Transmission Systems - Digital Networks - Quality And Availability Targets; Error Performance Parameters And Objectives For International, Constant Bit Rate Digital Paths At Or Above The …
- ITU-T Q.422, Clauses For Exchange Line Signaling Equipment

In addition, there are the robbed-bit signaling requirements of:

- ANSI T1.403, Telecommunications - Network and Customer Installation Interfaces - DS1 Electrical Interface
- AT&T TR-008, Digital Interface Between The SLC-96 Digital Loop Carrier System And A Local Digital Switch
- BELLCORE/Telcordia GR-303, Integrated Digital Loop Carrier System Generic Requirements Objectives And Interface, Tables 12-3 and 12-4.
- ATT PUB 43801, Digital Channel Bank Requirements and Objectives, November 1982.

Software Architecture

Products requiring T1/E1 interfaces face the daunting task of providing many low-level functions so that the applications conform to the different T1/E1 standards. At the same time, for ease of development, internally they must supply a set of function calls that permit high-level application software development independent of the hardware implementation that conforms to the appropriate standards. Using the layered architecture shown in the previous section will provide the necessary base to develop a T1/E1 project.

Specific Issues of T3/E3

Overview of T3

T3 also known as DS3 provides a 44.736 MHz electrical interface. The T3 electrical interface consists of two coaxial cables, one for transmit and one for receive, at a line impedance of 75 Ohm. The electrical T3 signal looks as shown below:

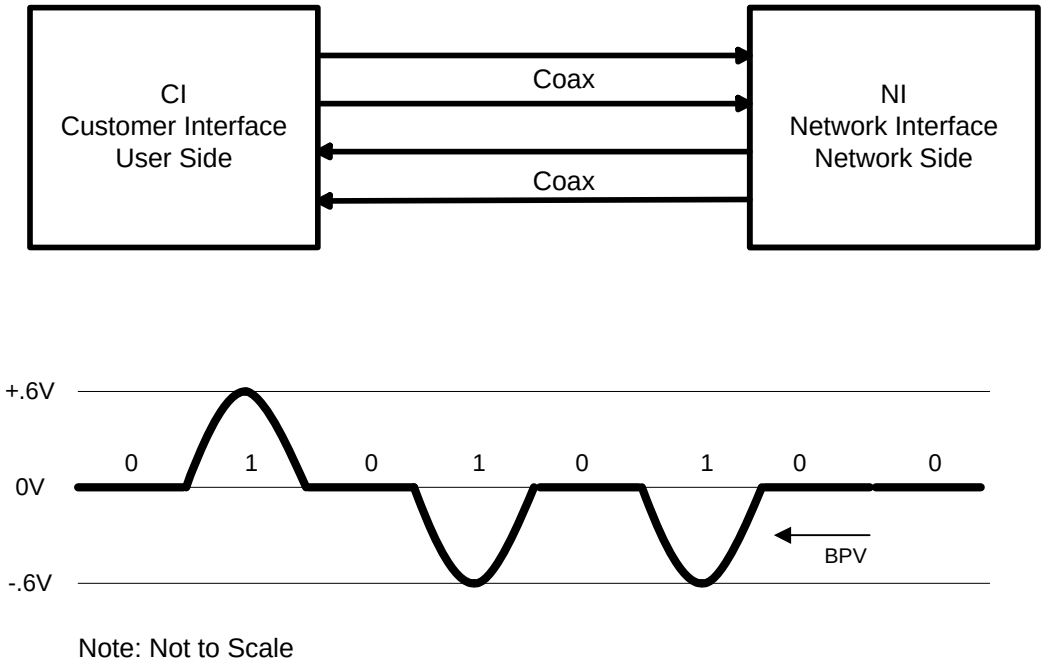


Figure 13. Basic T3 Interface

As shown above, the T3 interface consists of the Customer Interface (CI) at one end and the Network Interface (NI) at the other. A nominal DS3 signal is a Bi-Polar signal with amplitude of about 0.6V. Each “one” (or Mark) on the line is a positive or negative polarity pulse and each “zero” is no pulse. Each Mark must alternate in polarity – that is, if the first 1 is a positive pulse, then the next pulse is negative. This requirement gives rise to the term “Alternate Mark Inversion” or AMI line encoding.

However, there is an issue with AMI encoding. When the T3 signal contains long strings of zeros, there are no transitions in the signal that the electronics can look for to determine where the bits are. And after too many zero (175 plus/minus 25), the electronics will determine that the signal has entered a failure condition known as Loss of Signal. To prevent this problem, a line encoding scheme known as B3ZS - bipolar with three-zero substitution – was created. The B3ZS encoding scheme will substitute a string of 3 consecutive zeros with the special pattern. The encoding schemes use bi-polar violation(s) as a method to send long strings of zeros as listed below:

000 will be substituted with 00V where V is a positive violation pulse if the previous 1 was also a positive pulse

or

000 will be substituted with 10v where v is a negative violation pulse if the next bit is a positive pulse.

The bi-polar violations that are sent and received as part of a B3ZS encoded string of zeros are not counted as part of the performance errors on the line.

The T3 signal can either be channelized or unchannelized. A T3 signal consists of payload bits, used to carry the data over the T3 trunk, and overhead bits, used to determine where the payload is located (i.e. framing bits) and other overhead bits used for performance monitoring.

The T3 frame looks as follows:

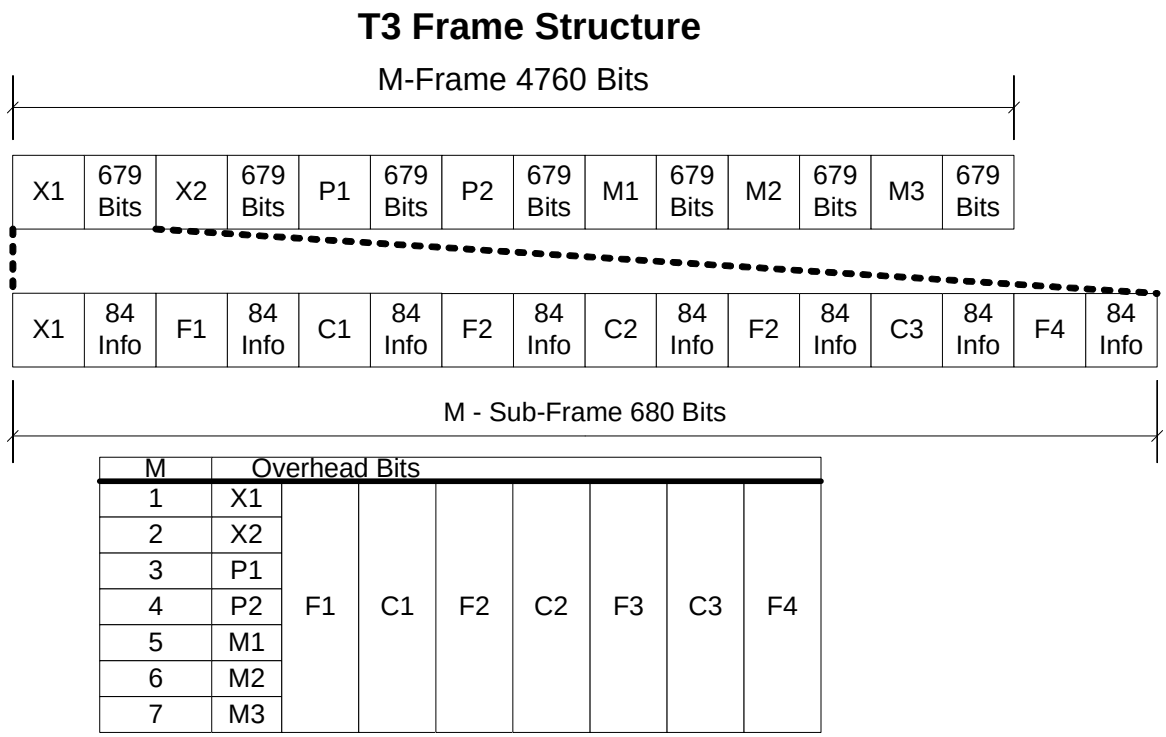


Figure 14. Basic T3 Frame

The T3 frame consists of a 4760 bits per frame. This translates into a frame rate of 44.736M/4760 = 9,398.32 frames per second. The T3 frame is constructed from 7 M-sub frames of 680 bits each. Each M-sub frame is the same with the exception of the first bit.

The overhead bits in the first M-sub frame consists of X1, F1, C1, F3, C2, F3, C3, and F4. The four F bits are used for establishing T3 frame alignment. The three C-bits are used for stuffing or other overhead purposes depending upon the framing format.

In the first bit position of the M-sub frame, the bits are used as follows:

X1, X2 – This is the remote alarm indication bit. When X1 and X2 are 0, the remote alarm is asserted and when they are 1, the remote alarm is clear.

P1, P2 – These bits should always be the same and represent the EVEN parity of all the information bits of the previous T3 frame. NOTE: The overhead bits are excluded from the parity calculation.

M1, M2, M3 – These are the multi-frame alignment bits. The M-bits are used to determine which M-sub frame is the first M-sub frame. The M1, M2, and M3 bits are always the value 010.

In each M-sub frame, the bits are used as follows:

F1, F2, F3, F3 – These are frame alignment bits. The F-bits are used to identify where an M-sub frame is in the signal. The F1, F2, F3, and F4 bits are always the value 1001.

C1, C2, C3 – The stuff bits are used differently depending upon the frame format. In M23 framing they are used as stuff indicator bits. In C-bit parity, the C-bits are used to carry other overhead information. The use of the C-bits will be described later.

A T3 interface can be used in two manners: unchannelized or channelized. In unchannelized applications, the payload bits are used to carry data traffic such as frame relay. All T3 related parameters such as alarms, configuration, loopbacks, FEAC channel, performance monitoring are still applicable in unchannelized applications.

In channelized applications, the T3 payload consists of seven multiplexed T2 signals. There are two methods to multiplex the T2 signals:

- M23 Framing
- C-bit parity Framing

These two methods differ substantially in how the multiplexing is done. In M23 framing, which is called M13 framing by most people and is used throughout this document, assumes the T2 signals are asynchronous to each other. The overhead bits, called C-bits, are used to control stuffing of the T2 signals. Stuffing is a method that allows the T2 signal to be adapted to a higher frequency clock by adding additional bits. Slower T2 signals will

have more bits added while faster T2 will have less bits added. The C-bits indicate how many bits have been added to the T2. After the stuffing operation, the resulting T2 signals are synchronized to each other and are multiplexed into the T3 signal. Additional overhead bits are used to provide framing for the resulting signal.

The stuffing bits in M13 framing are also used to control loopbacks. Normally, the three C-bits are the pattern 000 or 111 and the receiver uses a majority vote to determine what the value is to prevent errors from causing incorrect operation. However, when the far end wants to request that the near end goes into loopback, the third C-bits will be inverted. Thus, when the near end receives 001 or 110, the far-end is requesting that a loopback be invoked. The loopback control by the inverted 3rd C-bit is a T2 Line loopback. The C-bits in the first M-sub frame control the loopback of the first T2, the C-bits in the second M-sub frame control the loopback of the second T2 and so forth. In C-bit parity format, loopbacks are requested via the FEAC channel. However, via the FEAC channel, only T3 level and T1 level loopbacks can be requested. The FEAC channel cannot request a loopback at the T2 level.

As the Public Switched Telephone Network (PSTN) has evolved, the network has become more closely synchronized. Also, although T2s can be deployed as stand alone interfaces, they are almost never used this way. Thus, the T2s can be required to be synchronous. When the T2s are synchronized, the C-bits used for stuffing are no long required and can be used for other purposes. The C-bit parity format defines how these C-bits are used.

The C-bits are used for:

- FEAC Channel – The Far End Alarm and Control (FEAC) channel is used to inform the far-end piece of equipment of alarm and fault conditions as well as request loopbacks.
- PMDL Channel – The Path Maintenance Data Link (PMDL) is a 28.8 Kbits/sec channel that provides a communications channel to the far-end piece of equipment.

Packets carried over the PMDL channel are HDLC (High Level Data Link Control) formatted.

- FEBE bits – The FEBE (Far-End Block Error) bits are used to inform the far end that you have received a T3 frame that contained parity errors. The received FEBE bits are used to accumulate far-end performance information.

For both the M13 and C-bit parity format, each T2 contains 4 T1s. The T1s are multiplexed together in a similar fashion that the T2s are multiplexed together in M13 framing. The framing at the T2 level is shown in the following diagram:

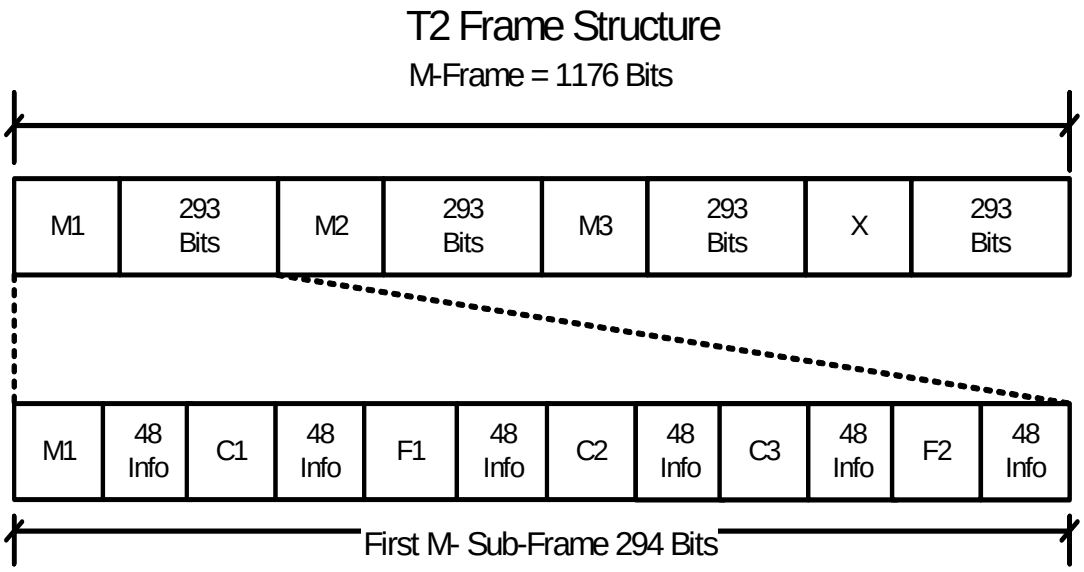


Figure 15. T2 Frame Structure

The T2s are built from 4 M-sub frames in a similar fashion that the T3s are built form 7 M-sub frames. Please be careful of the terminology. The T2 frame and the T3 frame use the same nomenclature, but they mean different things – very confusing.

The T2 frames have the following overhead bits:

F1, F2 – These are the T3 frame alignment bits – The F1, F2 bits are set to 01.

M1, M2, M3 – These are the multi-frame alignment bits – The M1, M2, M3 bits are set to 011.

X – The X-bit is used as the remote alarm indication bit. When X is a 0, the remote alarm is asserted and when X is a 1, the remote alarm is clear.

C1, C2, C3 – These are the stuffing bits. The stuffing bits are used in similar manner that the stuff bits in the T3 frame are used. . Stuffing is a method that allows the T1 signal to be adapted to a higher frequency clock by adding additional bits. Slower T1 signals will have more bits added while faster T1 will have less bits added. The C-bits indicate how may bits have been added to the T1. After the stuffing operation, the resulting T1 signals are synchronized to each other and are multiplexed into the T2 signal. Additional overhead bits are used to provide framing for the resulting signal.

The algorithm used to multiplex 4 T1s to a T2 is the same in C-bit parity framing and M13 framing. Normally, the three C-bits are the pattern 000 or 111 and the receiver uses a majority vote to determine what the value is to prevent errors from causing incorrect operation. However, when the far end wants to request you to go into loopback, the third C-bits will be inverted. Thus, when you receive 001 or 110 the far-end is requesting that a loopback be invoked. The loopback control by the inverted 3rd C-bit is a T1 Line loopback. The C-bits in the first M-sub frame control the loopback of the first T1, the C-bits in the second M-sub frame control the loopback of the second T1 and so forth. In C-bit parity format, loopbacks of the T1s are requested via the FEAC channel and inversion of the 3rd C-bit is not used.

Note: These C-bits are in the T2 frame format and should not be confused with the C-bits used in the T3 frame format; they are different C-bits that are used to stuff the T1 signals into the T2. The algorithm used to multiplex the T1s into the T2 is the same for C-bit parity and M13 formatted T3 signals. The following table summarizes some of the major differences between M13 and C-bit parity formats.

	M13 Format	C – Bit parity format
Loopbacks	No loopback at the T3 level is defined. Loopbacks at the T2 level are requested by inverting the third C-bit in the T3 frame associated with the T2. Loopbacks at the T1 level are requested by inverting the third C-bit in the T2 Frame associated with the T1.	Loopback of the T3 is requested by sending a FEAC command. No T2 loopbacks are defined. T1 loopbacks are requested by sending a FEAC command.
Near End performance	Near end performance data is accumulated for performance measures such as bi-polar violations, errors in the framing bits, parity errors, etc.	Near end performance parameters are extended from the parameters available from the M13 Frame format.
Far End performance	The only far end performance data accumulated for M13 format is the FCCP-PFE parameter.	The Far End performance data is accumulated via processing information from the far end. The FEBE bits, RAI, etc. are used to accumulate the far-end performance information.

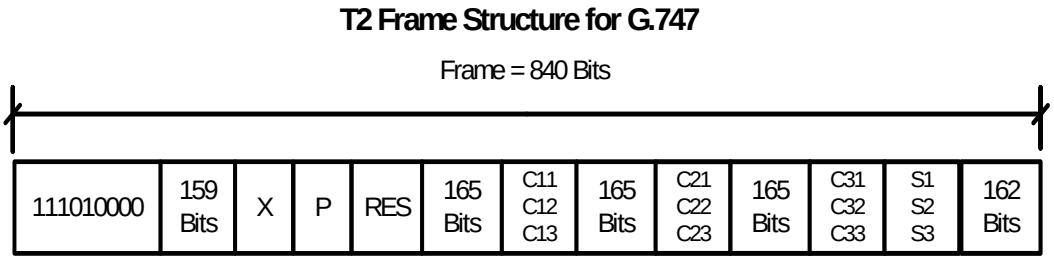
	M13 Format	C – Bit parity format
FEAC	Not Applicable	The FEAC channel contains two types of traffic: Signals and Commands. Signals consist of a continuous stream of the following pattern: 11111111 0SSSSSS0 Where 'SSSSSS' is the signal code. Signals are sent for a variety of reasons such as in response to alarms conditions. The other type of traffic is commands. Commands consist of 10 repetitions of a loop back activate or loop back deactivate pattern followed by 10 repetitions of the type of loopback requested. The T3, a specific T1 or all T1s can be requested.

	M13 Format	C – Bit parity format
PMDL	Not Applicable	The ANSI T1.107 standard defines three standard messages for the PMDL link. These are: Test Signal Identification T3 Path Signal Identification Idle Signal Identification Nominally, these signals are sent once per second.

G.747

G.747 is an ITU standard that defines a method of how to multiplex 3 E1s into a T2. This is typically encountered in M13 Multiplexers where the resulting T2s are multiplexed into a T3.

We have been seeing increasing use of G.747 in non-North American applications because, using G.747, a T3 can carry 21 E1s. This compares to E3 where an E3 can only carry 16 E1s. Since the physical cable, 75-Ohm Coaxial Cable, is the same for both T3 and E3, T3 becomes more efficient for large numbers of E1s.



The T2 frame for carrying E1s is shown below:

Figure 16. T2 Frame Structure for G.747

In the T2 frame structure for G.747, the first 9 bits are used as the frame alignment sequence. This allows the start of the T2 frame to be identified. The other bits are as follows:

159 bits, 165 bits, 162 bits – This are the bits from the E1 tributaries.

X – The remote alarm bit. When 1, the remote alarm is asserted and is clear when the X-bit is 0

P – is the even parity calculation over the previous frame

RES – is reserved for future use. Set to 1 when not used.

Cji = stuffing bits. Where i = the E1 tributary the stuffing bit is associated with and j = the stuffing bit. A majority-voting scheme is used to determine if stuffing is required or not.

Overview of E3

E3 provides a 34.368 MHz electrical signal that can be either channelized or unchannelized. The E3 signal consists of payload bits, used to carry the voice or data, and overhead bits that are used to locate the payload information (i.e. framing patterns) and other overhead bits used for clock justifications and alarming. E3 has two types of framing formats, which we will cover in the following sections:

- G.751 – See the ITU-T specification G.751
- G.832 – See the ITU-T specification G.832

G.751 Framing

G.751 framing is used for multiplexing lower speed signal, E1s, into the E3. In G.751 framing, the E1 signals are first multiplexed into E2 signals as shown in the following diagram.

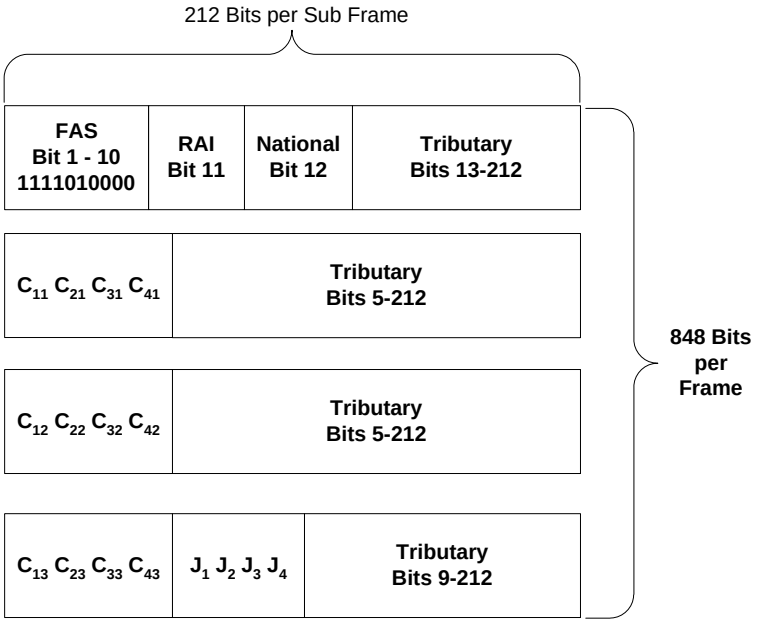


Figure 17. G.751 Frame

As shown in the diagram, the E2 frame consists of four sub-frames of 212 bits each. Each sub-frame consists of some overhead bits and some tributary bits.

In the first sub-frame, the first 10 bits are the Frame Alignment Sequence (FAS) bits. These bits provide information required to locate the tributary bits in the signal. Bit 11 is the Remote Alarm Indication (RAI) bit. The RAI bit is used to convey to the equipment at the far end that the receiver is having problems recovering the signal. For example, if you are in an Out Of Frame (OOF) condition, you will inform the far end via the RAI bit. Bit 12 is the National bit and may be defined by the country where the equipment is provided. When crossing a national boundary, the National Bit must be a 1.

In the 3 other sub-frames, the C1n, C2n, C3n and C4n bits are used as justification indicators. A value of 000 indicates a negative justification while a 111 indicates a positive justification. A majority vote of the three bits is used to prevent bit errors from creating incorrect justifications moves. The J1, J2, J3, J4 bits are the justification bits for each one of the E1 tributaries. The E2 frame uses an all ones pattern to indicate an Alarm Indication Signal (AIS).

The E3 frame is very similar to the E2 frame. Instead of containing 4 E1s, the E3 frame contains 4 E2s. The frame structure for the E3 is shown below.

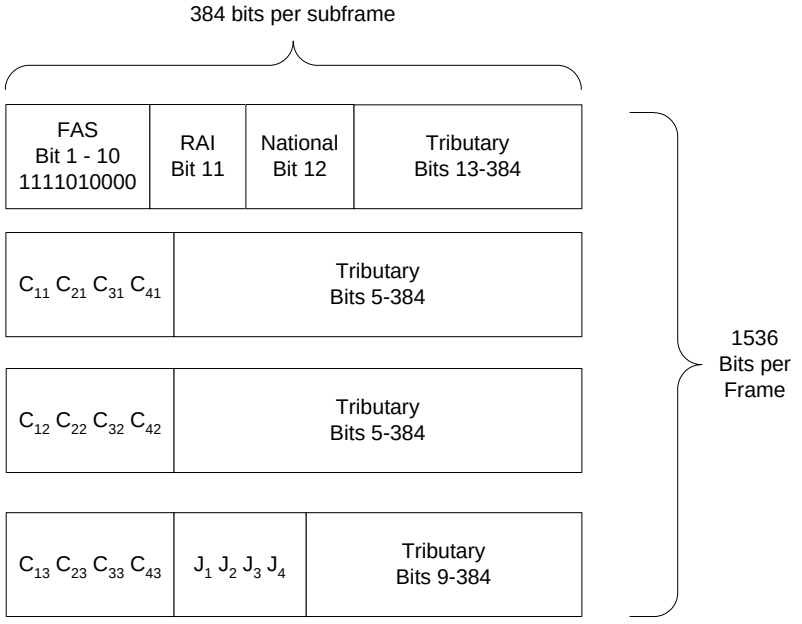


Figure 18. E3 Frame

As shown in the diagram, the E3 frame consists of four sub-frames of 384 bits each. Each sub-frame consists of some overhead bits and some tributary bits.

In the first sub-frame, the first 10 bits are the Frame Alignment Sequence (FAS) bits. These bits provide information required to locate the tributary bits in the signal. Bit 11 is the Remote Alarm Indication (RAI) bit. The RAI bit is used to convey to the equipment at the far end that the receiver is having problems recovering the signal. For example, if you are

in an Out Of Frame (OOF) condition, you will inform the far end via the RAI bit. Bit 12 is the National bit and may be defined by the country where the equipment is provided. When crossing a national boundary, the National Bit must be a 1.

In the 3 other sub-frames, the C1n, C2n, C3n and C4n bits are used as justification indicators. A value of 000 indicates a negative justification while a 111 indicates a positive justification. A majority vote of the three bits is used to prevent bit errors from creating incorrect justifications moves. The J1, J2, J3, J4 bits are the justification bits for each one of the E2 tributaries. The E3 frame uses an all ones pattern to indicate an Alarm Indication Signal (AIS).

G.832 Framing

G.832 E3 framing is very different than G.751 E3 framing. Where G.751 is designed to transport E1s in a multiplex manner, G.832 is designed to transport ATM traffic. Consequently, the framing is very different as shown below:

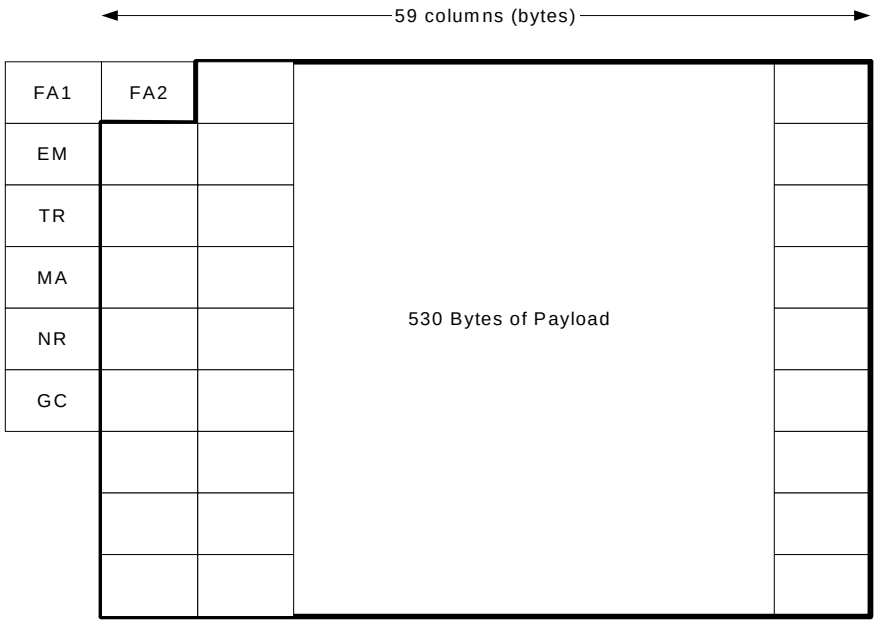


Figure 19. G.833 E3 Frame

If you are familiar with SDH (Synchronous Digital Hierarchy) optical framing, the G.832 framing for E3 has a number of similarities.

The FA1, FA2 pattern provides the framing of the signal and consists of the pattern:

FA1: 11110110

FA2: 00101000

This is the same framing bit pattern that is used in the SDH optical signal.

The next overhead byte is the Error Monitoring (EM) byte. The EM byte provides the ability to detect errors in the E3 frame via a BIP-8 error-checking scheme. The BIP-8 does a byte-by-byte even parity calculation over the previous frame and places the value in the current frame's EM location. Thus, the number of BIP-8 errors that are possible in one frame ranges from 0 to 8.

The Trail Trace (TR) byte location contains the trail access point identifier. The TR byte contains a 16-byte long string that is sent over 16 frames. The string allows the destination to determine if it is connected to the proper source. The 16-byte string is structured as follows:

Byte Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Bit 1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit 2	C1	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 3	C2	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 4	C3	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 5	C4	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 6	C5	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 7	C6	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 8	C7	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

The first byte consists of a 1 in the first bit followed by seven CRC bits. These CRC bits represent a CRC-7 code over the other 15-byte locations. The polynomial for the CRC-7 code is $X^7 + X^3 + 1$.

The 15 other bytes have a 0 as the first bit position, followed by a 7-bit character from the ITU-T specification T.50. ITU-T T.50 specification is very similar to the ASCII character set.

The Maintenance and Adaptation Byte (MA) contains the following bit fields:

Bit	1	2	3	4	5	6	7	8
	RDI	REI	Payload Type			Multi-Frame Indicator		SSM

Figure 19. MA Byte

The Remote Defect Indication (RDI) is similar to the RAI used in G.751 E3 framing. The RDI bit is used to indicate to the far end that the equipment is receiving an impairment of some kind.

The Remote Error Indication (REI) is used to inform the far end that one or more errors where received in the EM byte location. The REI indication can be used by equipment to determine that there is a problem with the E3 being transmitted.

The Payload type field indicates the type of traffic that is being carried on the E3 trunk. The values that are defined in ITU-T G.832 are:

Code	Payload Type
000	Unequipped
001	Equipped, non-specific
010	ATM
011	SDH TU-12s

The Multi-Frame indicator and the SSM bit are used in conjunction with each other. The SMM bit is a four-bit Synchronization Status Message (SMM) with one bit being provided each frame. The Multi-Frame bit provides a binary count from 00 to 11 such that when the MF is a 00, the most significant SMM bit is present and when the MF is a 11, the least significant SMM is present. From ITU-T G.707, the values of the SSM message is as follows:

SSM Bits MSB - LSB	Synchronization quality level description
0000	Quality unknown
0001	Reserved
0010	Rec. G.811
0011	Reserved
0100	Rec. G.812 transit
0101	Reserved
0110	Reserved
0111	Reserved
1000	Rec. G.812 local
1001	Reserved
1010	Reserved
1011	Synchronous Equipment Timing Source (SETS)
1100	Reserved
1101	Reserved
1110	Reserved
1111	Do not use for synchronization

Some older equipment does not use the SSM scheme for indicating the source of synchronization. For these pieces of equipment, a all 0 pattern indicates that the source is traceable to a primary timing source while a 1 indicates that it should not be used as a timing source.

The Network Operator byte (NR) can be used in one or two manners. The NR byte may be use for maintenance activities or for a communications link in Tandem applications.

The final byte, the General Purpose byte (GC), can be used by maintenance personal for a voice/data link.

Products requiring T3/E3 interfaces face the daunting task of providing many low-level functions so that the applications conform to the different T3 or E3 standards.

Standards Requirements

The software must handle the following requirements:

- ANSI T1.107 and ANSI T1.107a, Telecommunications - Digital Hierarchy – Format Specifications
- ANSI T1.404, Telecommunications – Network-to-Customer Installation – DS3 Metallic Interface
- ANSI T1.231, Telecommunications – Layer 1 In-Service Digital Transmission Performance Monitoring

Alarms and Configuration

The alarm and configuration module includes processing of the following T3/E3 functions:

1. Transmission and reception of alarms. The alarm and configuration module will process line defects, such as loss of frame, into alarm conditions. The integration times for T3 will default to the times defined in ANSI T1.231 but can be changed via an API call. The module also handles G.826 for E3.
2. Configuration. The alarm and configuration module will allow the application to set frame format (C-bit/M13 or HDB3), line encoding, line build-out, and channelized/un-channelized application.

3. User versus Network Side. As with most telecommunications interfaces, the T3/E3 protocol is not symmetrical. Either user side (also known as Customer Interface – CI) or Network Side (NI) can be selected.
4. FEAC channel – The FEAC channel is only defined for the C-bit parity frame format. The FEAC channel is used to communicate alarm and failure information, called FEAC signals, as well as request loopbacks be put up or taken down, called FEAC commands. For FEAC signals, the T3 alarm and configuration manager will perform prioritization of the FEAC code per T1.107 before the code is transmitted. On the receive size, the configuration and alarm module can be configured to alert the application software after a valid FEAC signal has been recognized. FEAC commands can be transmitted or receive by the alarm and configuration module. When transmitting a FEAC command, the application software will be notified by a callback after the FEAC command has been sent. On the receive side, the alarm and configuration manager will detect a properly formatted command request and, if configured to, notify the application and/or implement the command.
5. PMDL channel – The PMDL channel is only defined for the C-bit parity frame format. The PMDL transfers HDLC based packets between the CI and NI sides. The application can transmit as well as receive PMDL packets.
6. Loopbacks – In C-bit parity format, loopbacks are requested via the FEAC channel (See above), but in M13 format, loopbacks are done by inverting C-bits. C-bits at the M13 frame level control loopbacks of the T2s while C-bits at the T2 framing level control T1 loopbacks. The alarm and configuration module will implement as well as transmit loopback requests of the appropriate C-bits. In unchannelized T3, this feature is not applicable.
7. E3 Messages – Handle the National Bit, Status Synch message, Payload Type message and Trail Trace message.

Performance Monitoring

The performance monitoring module provides the following features:

1. Performance Monitoring – The performance monitoring software records the performance parameters specified in for both T3 and E3. The T3 performance parameters for ANSI T1.231 are stored in 15-minute buckets for a total of 192 buckets (representing 48 hour information). When configured in C-bit parity format, the performance monitoring includes the far-end information. For E3, performance monitoring uses G.826 with 96 15-minute buckets.
2. Time-of-day-processing – T1.231 specifies that performance information be tracked according to the time of day starting at 12:00 midnight and proceeding every 15 minutes. However, there are reasons that the time of day may need to be changed – daylight savings time for example. The performance monitoring will manage the change to the time of day and maintain the correct information in the buckets per T1.231.
3. Threshold crossing alerts – The performance monitoring module provides the ability for the application to set thresholds on the performance parameters. During processing, when the performance monitoring module determines that the threshold has been crossed, it will notified the application via a call back.

Software Architecture

Products requiring T3/E3 interfaces face the daunting task of providing many low-level functions so that the applications conform to the different DS3 standards. At the same time, for ease of development, internally they must supply a set of function calls that permit high-level application software development independent of the hardware implementation that conforms to the appropriate standards. Using a similar design to that of the T1 would yield a layered approach with the APIs to handle the interface.

Specific Issues of SONET/SDH

Overview of SONET

SONET is an acronym for Synchronous Optical Network, which is a method for carrying data, and voice traffic over optical or electrical interfaces, usually, for long distances. SONET is a framed signal that uses the basic building block of a Synchronous Transport Signal – Level 1 (STS-1). Scaling upwards, several STS-1s can be multiplexed together to form OC-Ns. Scaling downwards, an STS-1 can contain up to 7 VT Groups (VTGs) with each group containing four VT1.5s per VTG, three VT2s per VTG, two VT3s per VTG, or one VT6 per VTG.

VT1.5s are used for multiplexing T1s onto a SONET signal. Four T1s are combined into one VTG allowing a total of 28 T1s to be carried in a STS-1 – the same as a T3.

VT2s are used for multiplexing E1s onto a SONET signal. Three E1s are combined into one VTG allowing a total of 21 E1s to be carried in a STS-1 – the same as a T3 using the G.747 mapping scheme.

VT3s are used for multiplexing T1-Cs onto a SONET signal. Two T1-Cs are combined into one VTG allowing a total of 14 T1-Cs to be carried in a STS-1. Although defined by the standards, T1-Cs are not commonly found in the network.

VT6s are used for multiplexing T2s onto a SONET signal. One T2 is mapped into one VTG allowing a total of 7 T2s to be carried in a STS-1. Although defined by the standards, T2s are not commonly found in the network.

The Virtual Tributary (VT) contains the facility being transported as well as additional overhead information. The VT can contain the Alarm Indication Signal (AIS). In addition to AIS, the V5 byte can contain the signal-label. If the signal label is 0, then the VT is

unequipped. If the VT is unequipped when it is expected to be equipped the Unequipped Alarm will be raised. In addition, if the signal label byte does not match its expected value, then the Trace Identifier Mismatch alarm is raised.

Each SONET signal is made up of Line, Section, and Path overhead in addition to the payload. The OC-N signal can contain up to N STS-1 signals. If the OC-N signal contains N STS-1s, then there is one copy of the Line overhead, one copy of the Section overhead, and N copies of the Path overhead, one for each STS-1.

The Section Overhead passes from one connection to the next; the Line Overhead passes from one cross connect to the next cross connect, and the Path Overhead passes from one end to the next. The following diagram shows how this is organized.

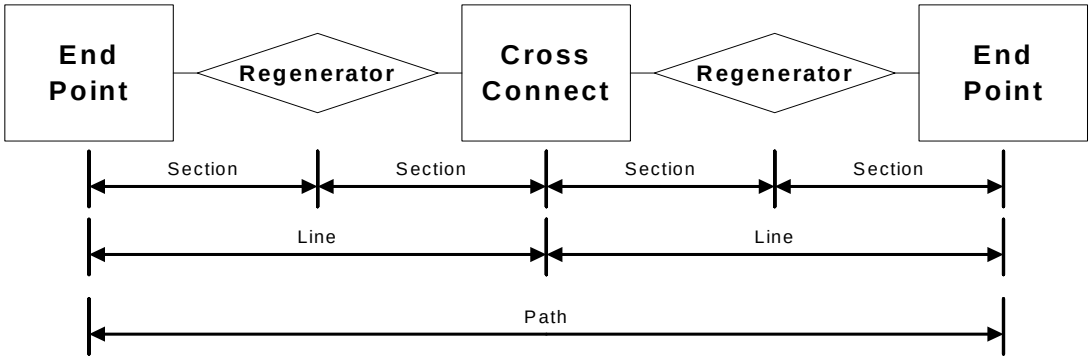


Figure 20. SONET Signal

In addition to STS-1s, the SONET signal can contain higher bit rate signals for carrying higher bandwidth signals. These are called concatenated signals such as STS-3C, STS-12C, and STS-48C. The number in OC-XC indicates the equivalent number of STS-1 in the signal (e.g. STS-12C uses the same bandwidth as 12 STS-1s). The overall bandwidth of the OC-N signal can contain combinations of STS-1s and concatenated signals up to the full bandwidth of the signal.

Section Overhead

The Section Overhead is used for the following purposes:

SONET Byte	Purpose
A1, A2	Framing – Used to acquire framing so that the payload can be extracted
J0	Section Trace Message
Z0	Section Growth
B1	Section Performance Monitoring – Contains the BIP-8 pattern
E1	Order wire – point to point voice link
F1	User Defined
D1-D3	192k bits/second Data Communication Channel (DCC)

Line Overhead

The Line Overhead is used for the following purposes:

SONET Byte	Purpose
B2	Line Error Monitoring – Contains the BIP-8 pattern
K1, K2	All 8 bits in K1 and 5 bits in K2 are used for the protection switching protocols defined for SONET. The remaining three bits in K2 (b6-b8) is used for carrying the Line AIS and Line Remote Defect Indication.
D4-D1	576k bits/second Data Communication Channel (DCC)
S1	Synchronization status message.
Z1	Reserved for future growth
M0	Used to carry the Line Remote Error Indication – The REI informs the far end that you are receiving BIP-8 errors.

SONET Byte	Purpose
M1	Used to carry the Line Remote Error Indication – The REI informs the far end that you are receiving BIP-8 errors. The M1 is used when the number of STS-1s is greater than 3.
E2	Order wire – A point to point voice channel
H1-H3	The Pointer – The pointer is used to locate the path information in the SONET frame. There is a pointer for each set of Path Overhead.
Z2	Reserved for future growth.

Path Overhead

The Path Overhead is used for the following purposes:

SONET Byte	Purpose
B3	Path Error Monitoring – Contains the BIP-8 pattern
J1	Path Trace Message
C2	Payload Label – Used to indicate the type of traffic being carried on in the path
G1	The G1 byte contains the Remote Defect Indication, either enhanced or standard, as well as the Remote Error Indication. The REI is used inform the far end that BIP-8 errors are being received.
F2	User defined channel
H4	Used for locating the payload.
Z3, Z4	Reserved for future growth
N1	Used for TANDEM connections.

Thus, the OC-N will contain one set of Line Overhead, one set of Section Overhead, and up to N copies of Path Overhead.

Hardware implementations of OC-N systems may involve various and mixed devices that access different sub-levels of an OC or STS signal. To accommodate this expected hardware architecture, the NComm SONET/SDH TMS Package has a polymorphic device driver mapping methodology. This methodology permits potentially many device drivers to appear as one virtual device to the NComm SONET/SDH TMS Package. This also maintains a commonality in general device driver development.

Overview of SDH

SDH, which stands for Synchronous Digital Hierarchy, is used primarily outside of North America and was derived from the initial SONET standards. Therefore, instead of describing SDH, this section will describe the differences from SONET. In SDH, the names of the different pieces have different names. Specifically:

SONET	SDH
Path	Path
Line	Multiplex Section
Section	Regenerator Section

The SDH view of the network then follows as:

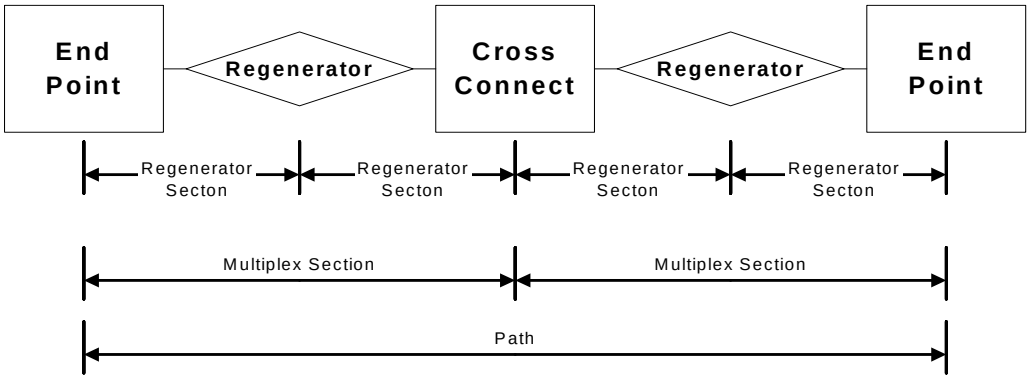


Figure 21. SDH Signal

The line rates for the SDH signal are called Synchronous Transport Module (STM) instead of STS. They are similar to the SONET standards as shown in the following table:

Rate (in MHz)	SONET Terminology	SDH Terminology
51.84	STS-1	*-NONE-*
155.52	STS-3	STM-1
622.08	STS-12	STM-4
2488.32	STS-48	STM-16
9953.28	STS-192	STM-64

One notable difference between SDH and SONET is the lack of a 51.84 MHz line rate or OC-1. In some SDH discussions the term STM-0 is used to indicate a 51.84 MHz link. However, this does not physically exist.

SDH, like SONET, is used to carry slower speed interfaces such as E1, E3, T1, etc over a high-speed optical interface.The methods used by SDH to multiplex these lower speed signals onto the optical fiber to create the STM-N optical signal is shown in the following diagram:

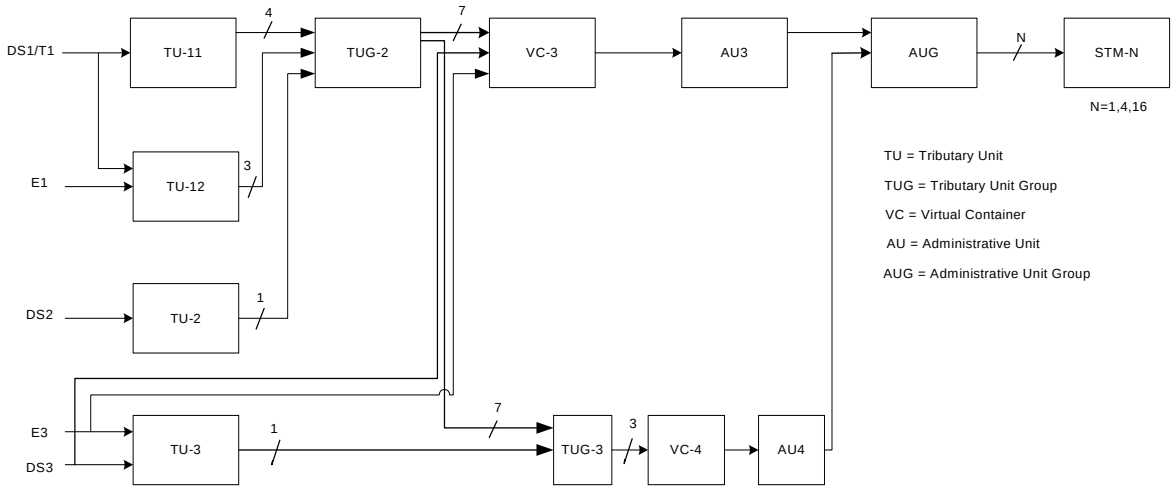


Figure 22. SDH Breakout

T1 Inputs – To carry T1 traffic over an STM-N interface, the T1 circuits can be mapped into a TU-11 or a TU-12. If mapped into a TU-11, 4 TU-11's are multiplexed into a TUG-2. The SONET version of this is mapping a T1 into a VT1.5 and 4 VT1.5's are multiplex into a VT-G. Alternatively, a T1 can be mapped into a TU-12 and three TU-12s mapped into a TUG-2.

E1 Inputs – To carry E1 traffic over a STM-N interface, the E1 circuit can be mapped into a TU-12. Subsequently, 3 TU-12's are multiplexed into a TUG-2. The SONET version of this is mapping an E1 into a VT2 and 3 VT2's are multiplex into a VT-G.

T2 Inputs – To carry T2 traffic over a STM-N interface, the T2 can be mapped into a TU-2 and, subsequently, into a TUG-2.

E3 Input – The basic unit of SDH is the VC-3 and a VC-3 can carry one E3 signal. An alternative to the VC-3, an E3 can be mapped into a TU-3.

T3 Input – A T3 like the E3 can be mapped into either a VC-3 or a TU-3.

The SDH frame is repeated every 125us. Now that you have a basic understand on how the multiplexing signal work, lets examine the SDH frame as shown in the next diagram:

STM-1 SDH FRAME

Regenerator Section Overhead

A1	A1	A1	A2	A2	A2	J0	NU	NU
B1	MD	MD	E1	MD	R	F1	NU	NU
D1	MD	MD	D2	ND	R	D3	R	R

Multiplex Section Overhead

H1	H1	H1	H2	H2	H2	H3	H3	H3
B2	B2	B2	K1	R	R	K2	R	R
D4	R	R	D5	R	R	D6	R	R
D7	R	R	D8	R	R	D9	R	R
D10	R	R	D11	R	R	D12	R	R
S1	Z1	Z1	Z2	Z2	M1	E2	NU	NU

Path Overhead

J1
B3
C2
G1
F2
H4
F3
K3
N1

Figure 23. SDH Frame

NU – National Use – These bytes are reserved for definition at the discretion of the country that the equipment is deployed in.

R – Reserved- These bytes are reserved for future use.

Regenerator Section Transport Overhead

A1/A2 – Framing bytes – The framing bytes consist of the pattern:

A1: 11110110

A2: 00101000

These bytes are used to recover the basic frame alignment of the SDH signal

J0 – Regenerator Section Trace Message – The Trace Message is used to validate that the two endpoints of the SDH facility are connected correctly. The J0 byte location contains the trail access point identifier. The TR byte contains a 16-byte long string that is sent over 16 frames. The string allows the destination to determine if it is connected to the proper source. The 16-byte string is structured as follows:

Byte Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit 1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit 2	C1	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 3	C2	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 4	C3	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 5	C4	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 6	C5	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 7	C6	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
Bit 8	C7	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

The first byte consists of a 1 in the first bit followed by seven CRC bits. These CRC bits represent a CRC-7 code over the other 15-byte locations. The polynomial for the CRC-7 code is $X^7 + X^3 + 1$.

The 15 other bytes have a 0 as the first bit position, followed by a 7-bit character from the ITU-T specification T.50. ITU-T T.50 specification is very similar to the ASCII character set.

B1 - BIP–8-parity byte – The B1 byte is used for monitoring performance of the Regenerator Section component of the SDH connection. Specifically, the B1 byte provides the ability to detect errors in the SDH Regenerator Section via a BIP-8 error-checking scheme. The BIP-8 does a byte-by-byte even parity calculation over the previous frame and places the value in the current frame's B1 location. The parity is calculated after the scrambling operation. Thus, the number of BIP-8 errors that are possible in one frame ranges from 0 to 8.

MD – Media Dependant Byte

D1 – D3 – Data communications channel – The data communications channel (DCC) provides a 196K bit per second communications channel which is used to administer the SDH network.

E1 – Orderwire – The order wire is a 64K bits/sec voice channel that provides a point-to-point voice communications channel that is used by maintenance personnel.

Multiplex Section Overhead

H1-H3 – Pointer Bytes – The Pointer bytes are used to locate the path information in the SDH frame. Although, the SDH network is nearly synchronous, there exist different clocking sources. Just like the SONET network, the pointer bytes are used to account for timing differences between different nodes in the SDH network.

D4 – D12 –Data communications channel – The data communications channel (DCC) provides a 576K bit per second communications channel which is used to administer the SDH network.

S1 – Synchronization Status Message – The S1 byte is used to carry synchronization trace information. The low order 4 bits (Bits 5-8) carry the message as shown in the following table.

S1 Bits b5-b8	Synchronization quality level description
0000	Quality unknown
0001	Reserved
0010	Rec. G.811
0011	Reserved
0100	Rec. G.812 transit
0101	Reserved

0110	Reserved
0111	Reserved
1000	Rec. G.812 local
1001	Reserved
1010	Reserved
1011	Synchronous Equipment Timing Source (SETS)
1100	Reserved
1101	Reserved
1110	Reserved
1111	Do not use for synchronization

Z1 – Z2 – Growth Bytes – Reserved for future growth.

M1 – Far End Block Error Byte – The M1 byte carries information to tell the far end that errors exist in the receive link. This allows for network maintenance personal to isolate and correct impaired networks.

K1 – K2 - Protection Switching Byte – The K1/K2 bytes are used for controlling protection switching of SDH facilities. A protocol between the two endpoints negotiates how traffic will be placed over the normal and protection SDH networks. Multiple protocols exist depending upon the network topology.

E2 – Order wire - The order wire is a 64K bits/sec voice channel that provides a point-to-point voice communications channel that is used by maintenance personnel.

Standards Requirements

The software must handle the requirements of:

- ANSI T1.105 and ANSI T1.231
- ITU-T G.783 and ITU-T Q.921
- Telcordia GR.253

Alarms and Configuration

The alarm and configuration module includes processing of the following SONET/SDH functions:

1. Transmission and reception of alarms. The alarm and configuration module will process line defects, such as loss of frame, or AIS into alarm conditions. The integration times will default to the times defined in ANSI T1.231 or ITU G.826 but can be changed via an API call.
2. Configuration. The alarm and configuration module will allow the application to set the various operating parameters of the different SONET/SDH signal levels.
3. User vs. Network Side. As with most telecommunications interfaces, the SONET/SDH protocol is not symmetrical. Either user side (also known as Customer Interface – CI) or network side (NI) can be selected.

Performance Monitoring

The performance monitoring module includes the following features:

1. Performance Monitoring – The performance monitoring records the performance parameters specified in ANSI T1.231 or ITU G.826 whichever is appropriate. The performance parameters are stored in 15-minute buckets for a total of 192 buckets (representing 48 hours of information).
2. Time-of-day-processing – T1.231 specifies that performance information be tracked according to the time of day starting at 12:00 midnight and proceeding every 15 minutes. However, there are reasons that the time of day may need to be changed – daylight savings time for example. The performance monitoring will manage the change of the time of day and maintain the correct information in the buckets per T1.231. ITU G.826 has similar properties.
3. Threshold crossing alerts – The performance monitoring module provides the ability for the application to set thresholds on the performance parameters. During processing,

when the performance monitoring module determines that the threshold has been crossed, it will notified the application via a call back.

Japanese SDH

Most of the world can be accommodated by either standard SONET or SDH. The only important exception that we are aware of is Japan. The Japanese standards JT-G707 and JT-G783 are very similar to the standards, ITU-T G.707 and ITU-T G783. However, they are not identical and the differences must be implemented to insure standard compliance and interoperability. Japanese SDH makes some tweaks to standard SDH and adds the concept of the JHG or the Japanese Handling Group. The following goes into a little more detail of what these differences are.

STM-0: The STM-0 rate defined in JT-G707 is equivalent to the STS-1 rate defined by Bellcore.

There are two options for mapping the VC-11/TU-11 bytes.

1. STM-0 supports 28 VC-11/TU-11

2. STM-1 support 3*STM-0 (i.e. 28*3 VC-11/TU-11).

There are three alarms that are maintained separately for each handling group:

LOS – loss of HG alignment

AIS – detection of AIS in the HG

RDI – presence of three consecutive 0s in the Sp/BAIS bit.

VT Group Structure: The Japanese Handling Group (JHG) implementation of SDH maps and demaps 1.544 Mbits/s using VC-11/TU-11 in ***byte synchronous mode only***. For signaling information, it uses a handling group concept. This combines the

signaling information for six DS0s into a handling group. There are four handling groups for each DS1 line. The figure below shows the details of how the individual time slots are grouped together. Only two of the four time slots are shown in the figure.

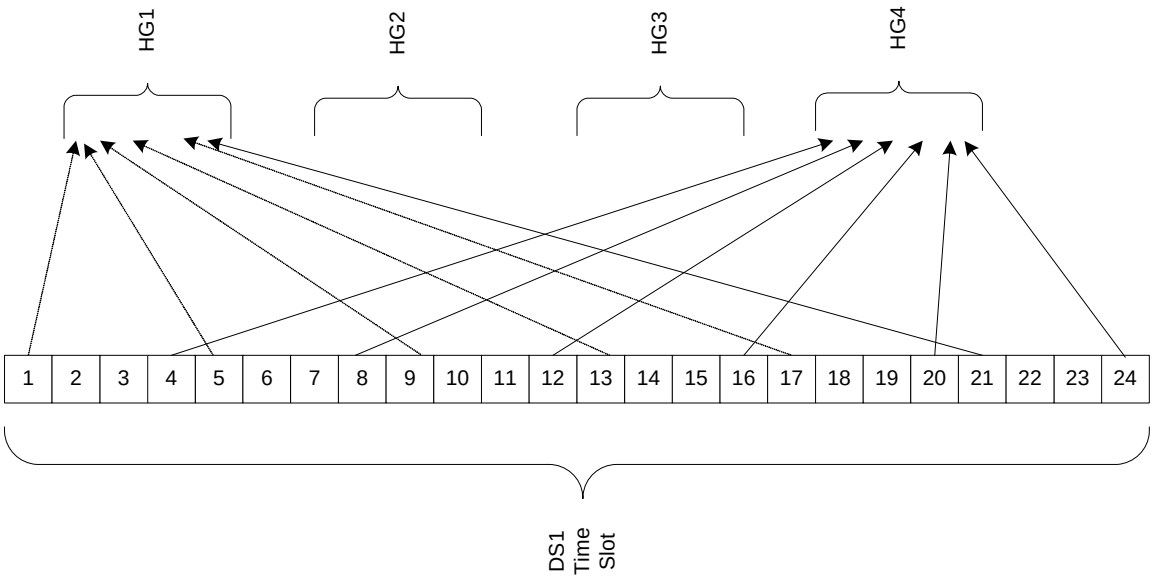


Figure 24. Time Slot Grouping

In the JHG implementation, the byte following the V5-byte in the VT SPE is called the W-byte. In byte synchronous mode, signaling bits (S1, S2, S3, S4) for each handling group get mapped into dedicated ST-bits. These bits are placed in the W-byte in the VT mapper frame as shown below.

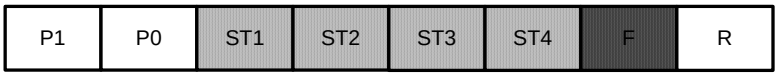


Figure 25. W-Byte Contents

Handling Groups

Each handling group consists of a framing bit (F), six signaling bits (As) and an alarm bit (Sp), which is also called Backward AIS (BAIS). The JHG implementation uses the ST frame format that consists of eight VT frames (1 ms). The signaling bits of the eight W-

bytes from the ST frame format (ST1 – ST4) are shown below (See Figure 26). The F locations are arbitrary with respect to one another.

S1/HG1	S2/HG2	S3/HG3	S4/HG4
F	A22	A19	A12
A1	Sp	A23	A16
A5	F	Sp	A20
A9	A2	F	A24
A13	A6	A3	Sp
A17	A10	A7	F
A21	A14	A11	A4
Sp	A18	A15	A8

Figure 26. W-Byte Signaling Bits

The F-bit is an ST framing bit, which toggles between 0 and 1 every ST multiframe. This framing bit might not be aligned between the handling groups.

Automatic Protection Switching

There are four standards that define APS for SONET and SDH transport. The general set is T1.105 (there are a number of sub-documents). GR-253 addresses Linear APS while GR-1400 and GR-1230 specify Unidirectional Path Switched Ring (UPSR) and Bi-directional Line Switched Ring (BLSR), respectively. For SDH, G.841 defines Linear Protection Switching and Ring Protection Switching.

There are three models of Linear APS. They are 1 + 1, 1:1 and 1:n. Linear APS is used primarily in applications linked to the core network and by definition are point to point. All Linear APS solutions have the drawback of asymmetric delay. Additional buffering at the nodes is required to overcome this. Buffering raises the cost of the equipment

1 + 1 provides two fiber links. Each link carries identical traffic. The receivers at each end monitor the bit streams and choose the “best” one. 1 + 1 is the most expensive but also offers the fastest recovery often without any data loss. It is expensive because two receivers are required at each end point, twice as much fiber is needed, and no additional capacity is gained.

1:n including the special case of 1:1, as its name implies, provides one backup fiber for up to 14 primary fibers. The back-up fiber can carry low priority traffic when not used for back up. There is a time to detect and time to switch traffic from the primary to the back up, and some loss will be experienced. As the detect plus switching times must be kept to 60 mSec or less, conventional phone calls should not be dropped (The network is architected for calls to withstand up to 2.5 seconds of disruption). This method is much less expensive because one fiber provides coverage for multiple primary fibers.

Ring APS comes in two flavors. That is UPSR (Unidirectional Path Switched Ring) and BLSR (Bidirectional Line Switched Ring).

Of the Ring APS methods, UPSR is relatively simple. There are two, counter-rotating fiber links. Each fiber carries the identical traffic. Each “node” monitors both fibers and picks the

“best” one based on several criteria. These criteria include bit error rate, AIS (Alarm Indication Signal) and a couple of others. Some of the positive aspects of UPSR include that the receiver makes all decisions with no interaction with either the local or remote transmitter, no communications channel is needed, and it provides virtually un-interrupted service. The down side includes the need for redundant transceivers as well as introducing asymmetric delay (More information can be found in GR-1400.

BLSR is a bit more complicated. This method uses the K1/K2 bytes as well as other local indications to raise a flag to switch. Once the flag is raised, an independent “controller” then communicates with the local back-up facility (through the back-up SONET/SDH transceiver) via the K1/K2 bytes that communicates to a far end transceiver to prepare for the transfer of the traffic from the failed facility to the back-up facility. Together the switch is synchronized and executed.

Once a switch is executed by any of the methods (linear or ring), its next action is dependent on whether it has been configured as revertive or non-revertive. If it is revertive, the traffic will be automatically switched back to the original facility once it is recognized as good. In non-revertive, the traffic will stay switched until it is manually switched back.

Usually, these switches will be configured as non-revertive. The reason that APS has been implemented in the first place is that the traffic being carried is very large and a service outage is very costly. It is not uncommon for a bad line to appear as fixed for short periods of time while the carrier is determining and fixing the line problems. Automatically switching back could create repeated switches further disrupting service. In 1:1 or 1+1 modes, non-revertive is the norm. After a switch is performed and the faulty facility is repaired, that facility becomes that back-up line.

All APS must meet certain performance thresholds to be standard complaint. The budget for detecting a failure and initiating a switchover must be within 10 mS. Completion of the switchover must be within 50 mS of the initiation. Thus, the maximum time from detection of the failure to complete restoral is 60 mS.

There are several different criteria for making the switching decision. These include:

- AIS (Alarm Indication Signal)
- LOP (Loss Of Pointer)
- Unequipped (indicated in the C2 byte)
- RDI (Remote Defect Indication)
- Bit Error Ratio (Severely Errored Seconds/Errored Seconds)
- Bi-directional or 1:n APS uses the K1 and K2 bytes to also pass information.

Where these various indications are used depends on the type of APS being implemented. Line level indicators are used for Linear Models and Line and Path level for Ring models. K1/K2 are used only when there is a sharing of the back-up facility or when communication with the far end is required for coordination. An examination of the relevant standards will provide the details for your implementation.

Interoperability and NComm's TMS

While there are standards for SONET and SDH APS, interoperability is still an issue. There have been groups that have tried to remedy this situation but to date have not been successful. In discussions with carriers, this reality has forced them to use a single equipment provider within a network being protected. Carriers typically do not like to be tied to a single source and NComm expects this to change. One of the benefits of NComm's Trunk Management Software generally and the APS modules in particular is that their use goes a long way towards achieving interoperability.

Software Architecture

Products requiring SONET/SDH interfaces face the daunting task of providing many low-level functions so that the applications conform to the different SONET/SDH standards.

Hardware implementations of OC-N systems may involve various and mixed devices that access different sub-levels of an OC or STS signal. The Trunk Management Software must supply a set of function calls that permit development of high-level application software independent of the hardware implementation. One way to do this is to use a polymorphic device driver mapping methodology. This methodology permits potentially many device drivers to appear as one virtual device to the upper level application software. This also maintains a commonality in general device driver development.

Trends in Embedded Hardware

The history of bus architecture

It's important to go back in time to the days of the non-intelligent buses such as VESA Local bus, EISA, ISA. These bus structures were in a word, "slow." As CPU speeds increased with Moore's Law, the need for a faster, more intelligent bus emerged. In the early 1990's, Intel invented the Peripheral Component Interconnect (PCI) bus, which had a sweeping effect on the bus structures of the day. The PCI bus began as a 32-bit/ 33 MHz bus capable of 1 Gb/sec of data transfers. But most important was the fact that, unlike its predecessors, it was an intelligent bus. Soon, all servers, workstations and personnel computers were equipped with a PCI bus.

Intel, knowing they had a good thing with the PCI bus, decided to share the technology with the world. This enabled others to build the bridge chips and drove the cost of implementing such a sophisticated bus down dramatically. Now with this low cost, high speed bus, new designs were emerging that solved old dilemmas.

With the PCI bus specification now in the community, one of the dilemmas in the VME world was about to be solved. The VME industry was strapped to a maximum of 21 slots and no way to expand. System designers needed more functionality, especially in the area of I/O. The perfect solution would take the form of a daughter or mezzanine card but not protrude past a single slot mechanical envelope. Such a card would add the needed functionality without expanding the chassis. Now, with inexpensive PCI bridges available, the VME world solved this problem by developing single slot daughter cards we know now as the PCI Mezzanine Card or PMC card. The PMC card combines the electrical characteristics of the PCI bus with the mechanical dimensions of the Common Mezzanine Card or CMC format. The dimensions are similar to one credit card wide and three credit cards long.

VME was the defacto standard for those in need of an industrial, rugged form factor. Despite the adoption of PMCs for additional functionality, the VME lacked a number of features that were quickly becoming very important. Of those, hot swap, more user I/O pins, increased bus bandwidth, and most of all a cost reduction, emerged as the most important criteria for a new design. The solution took the form of a 6U Euro form factor card (just like VME) and was called CompactPCI.

So, with Intel's support, the PCI bus was born and has spawned many new designs, solving existing challenges. The future is bright with new technology and bus structures as well. The PCI bus continues to increase in speed, a new serial bus coined PCI Express promises data transfer speeds of greater than 80 Gb/s, and the CompactPCI and PMC form factors are expanding with a new form factor called ATCA (Advanced Telecom Computing Architecture) to meet greater power and cooling needs.

The remainder of this section will concentrate on describing in more detail the PCI bus and subsequent busses derived from it.

PCI Bus

First introduced in the early 1990s, the PCI bus had a sweeping effect on the current bus structures. The PCI bus, with initial data transfer speeds of 1 Gb/sec, was not only much faster than existing buses, but was designed to be more intelligent. Several cards can coexist on the bus simultaneously. Each PCI card contains a set of registers that allows the host processor to identify and properly configure resources at boot up. These registers also facilitate true Plug and Play capability in a PCI environment.

As shown below in the figure, PCI cards can take on several different forms, from short to long and regular to low profile. The bus structures also have several permutations from 32-bit/ 33 MHz to 64-bit/ 66 MHz. Refer to the following table for PCI bus architecture performance capabilities. A standard PCI bus can be up to four slots. Each slot can be any combination listed on the chart below. Electrical signaling is an important part of the PCI specification. The PCI signal is a "reflective" signal meaning that the signal travels to the

end of the bus, and as the reflection travels back the other direction, the information is considered valid. The signal voltage levels can be 3.3V, 5V, or Universal. The PCI specification is governed by the PCI-Sig group (http://www.pcisig.com).

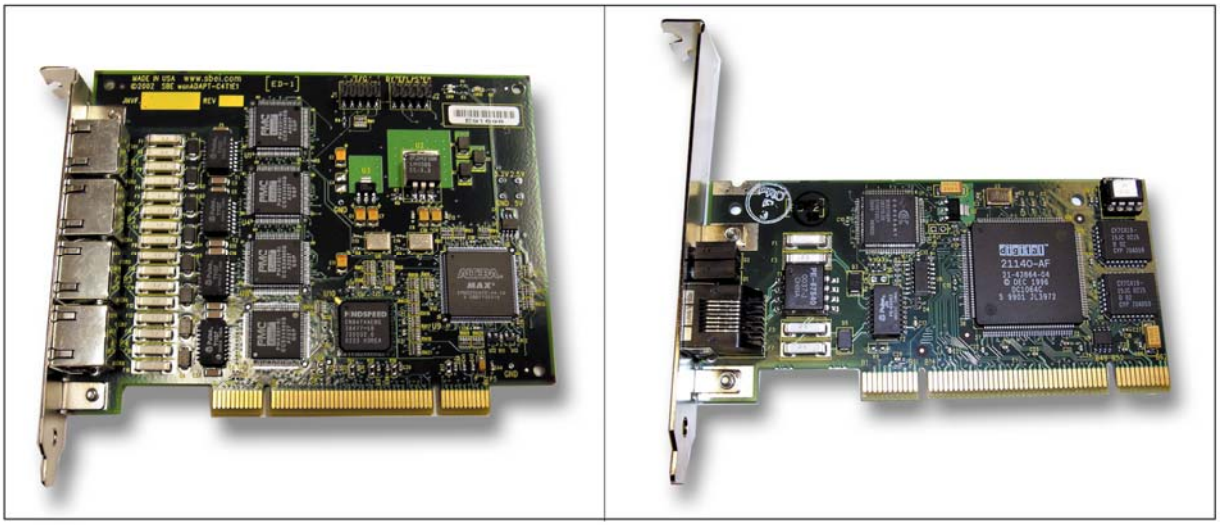


Figure 27. Sample PCI Bus Cards

PCI Bus			
Width	**Speed**	**Bandwidth**	
32 bits	33 MHz	132 MB	1 Gb
32 bits	66 MHz	264 MB	2 Gb
64 bits	33 MHz	264 MB	2 Gb
64 bits	66 MHz	512 MB	4 Gb
64 bits	133 MHz*	1024 MB	8 Gb

* Note: PCI-X

The PCI bus has had an incredibly good run and shown great success in the past. However, even at 64-bit/133 MHz, the PCI is running out of gas for some applications. These applications will be much better served by evolving to PCI Express. However, many

less demanding interfaces such as T1/E1 and T3/E3 will continue to use the standard PCI bus for years to come.

PMC/PTMC

The PCI Mezzanine Card or PMC evolved from the PCI specification and was developed to solve the need to add more functionality to already over burdened VME systems. An IEEE specification (IEEE1386) governs the electrical and mechanical characteristics of the PMC card. The card measures 3” X 5” and is designed as a mezzanine or daughter card intended to fit between two adjacent boards without extending past a single slot. Along with adding more functionality to a system, PMCs can also serve to make a single base platform more flexible. As an example, the SBE HW400c/M can serve as a TDM switch when equipped with a T1 PMC card. Add a DSP resource card and you immediately have a VOIP Gateway.

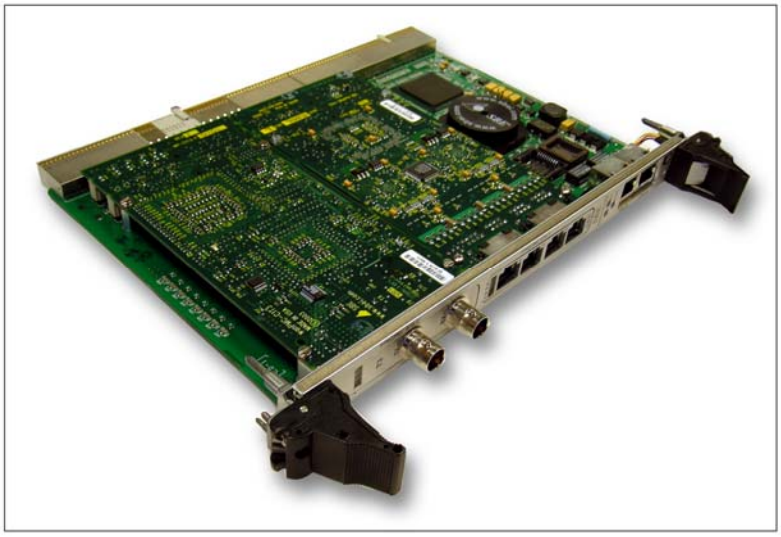


Figure 28. Sample PMC card

Where IEEE1386 defines the electrical and mechanical nature of the PMC card, PICMG 2.3 defines the pin out scheme as it pertains to the CompactPCI platform. PICMG 2.3 defines a 32 or 64-bit interface at 33 MHz along with user definable I/O pins available.

Another PICMG specification, PICMG 2.15 was developed to meet the special needs of the Telecom market. This member of the PMC family is referred to as PCI Telecom Mezzanine Card (PTMC) and retains the mechanical as well as the PCI 32-bit electrical specification. Where it differs is in the pin out assignments. The PTMC was designed to allow for at least two planes of data. The PCI is always there at 32 bits and intended as the control plane. In addition, the specification allows for a combination of the following high bandwidth data plane technologies:

- RMII (10/100 Mb Ethernet)
- Gbit Ethernet
- TDM
- Utopia
- POS/PHY

PTMC Interface Configurations

Interface	0	1	2	3	4	5	6	7	>7
Serial Tx/Rx	RESERVED	x	x	x	x	x	x		RESERVED
RMII			x	x					
RMII PHY Management I/F			x	x					
10/100/1000 Ethernet MDI Ports						2	2		
UTOPIA I/II (8-bit)		x		x	x		x		
UTOPIA II (8/16-bit)					x		x		
POS-PHY					x				
Local CT (20 bit)			x	x					
Extended Local CT (32 bit)						x			
USER IO Pins		64	66	6	0	40	4	64	
32-bit PCI	x	x	x	x	x	x	x	x	
64-bit PCI								x	
JTAG	x	x	x	x	x	x	x	x	
SMB	x	x	x	x	x	x	x	x	

Note: In configurations 2 and 3, two of the USER IO pins are located on Jn3, and are identified as USER1Z and USER2Z.

Much like PCI cards, PMC/PTMC cards have had a good success on VME, CompactPCI and custom motherboard platforms. PMC/PTMC cards will be in demand for many years to come.

COMPACTPCI

CompactPCI is yet another design spawned from the PCI specification and is also governed by the PICMG organization (www.picmg.com). In the mid 1990's the CompactPCI specification was emerging to meet the needs of those disgruntled with what was perceived to be missing features in VME platforms. Hot swap capability, high-speed bus, more user I/O pins, and lower cost were the most important design criteria. The CompactPCI specification retained the 6U Eurocard form factor with 2 mm pin and socket connectors yielding up to 220 pins. The pins are arranged in groups labeled J1 – J5. J1 and J2 are used exclusively for 32-64 bit/33-66 MHz PCI bus transfers enabling up to 4 Gb of data transfer capability. J3 – J5 are used for user I/O or specifically assigned to another bus structure.

The blade architecture along with pin and socket design makes CompactPCI a very rugged, robust architecture suitable for military, industrial, or telecommunication requirements. CompactPCI boards are inserted from the front of the chassis, and I/O can break out either to the front or through the rear.

Unlike PCI, which only supports up to four cards, the CompactPCI can support eight slots in a chassis. More slots can be enabled with the use of PCI bridging technology. Staged power and ground pins are used to provide true hot swap capability.

The PICMG organization has amassed a plethora of specifications intended to enhance the CompactPCI platform. Without a doubt, the two that standout the most are PICMG 2.5 and 2.16, Computer Telephony Specification and Packet Switching Backplane respectively. The Computer Telephony Specification, more often referred to as H.110, takes the user I/O pins on J4 and assigns them to carry up to 32 lanes of bi-directional

TDM data. A CompactPCI board equipped with an H.110 bus can easily be made to function as a TDM switch.

The Packet Switching Backplane, often referred to as 2.16 or PSB, is simply “Ethernet on the backplane”. Much like the H.110 above, 2.16 reassigns user I/O pins on J3 to support two sets of 19 full duplex 10/100/1000 Mb/s Ethernet channels. Two CompactPCI cards equipped with PSB could communicate out of band from the PCI bus via Ethernet. The added bonus is due to the fact there are two sets of Ethernet lines, the two boards now have a failover channel. Add an Ethernet switch into the system and now up to 19 boards can communicate in a non-blocking fashion all over Ethernet.

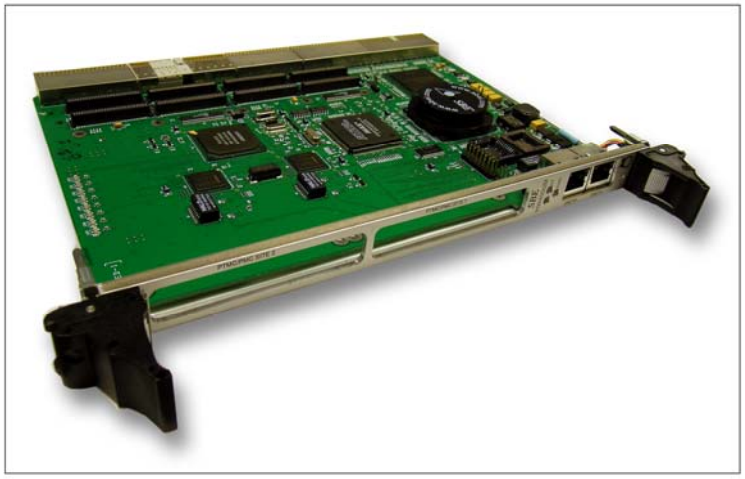


Figure 29. Sample CompactPCI card

The Future: PCI Express, ATCA, AMC…

As CPU speeds and physical access technologies continue to increase, dual 2 GHz servers have become commonplace, and 10 Gb Ethernet is just around the corner. These facts demand that bus structures also keep up. PCI, a parallel style bus structure is essentially exhausted at 64-bit/133 MHz, yielding 8 Gb of bandwidth. With 10 Gb Ethernet on the forefront, however, this clearly won’t be enough bandwidth to process the data. To address this emerging “need for speed,” Intel has developed a serial bus structure,

capable of 80 Gb/sec…that's a 10X increase over traditional PCI. Intel coined this new bus structure PCI Express. To date, there are a small number of servers supporting this new technology, but the future is clear that PCI Express will, like PCI, be deployed in every server on the market.

Embedded form factors are also evolving with the recent PICMG (www.PICMG.com) ratification of the Advanced Telecom Computing Architecture (AdvancedTCA®). This new blade style form factor boasts larger board size to accommodate more components, increased power, and better cooling characteristics. In addition, AdvancedTCA will also adapt to several different high-speed switch fabrics allowing for data bandwidth from 10 Gb to greater than 100 Gb. In the early stage, AdvancedTCA will fully embrace the standard PMC daughter card technology. However, in the future we will see the adoption of yet another form factor, AdvancedTCA Mezzanine Card (AMC). The AMC card will act much like its PMC brother, but will provide more board space and hot swap features.

Growing Demand for VoIP

Voice over Internet Protocol (VoIP) first emerged in the early 1990s as a way to conduct voice conversations over the Internet without incurring telephone charges. Today, the capacity, speed and reach of data networks are expanding much faster than traditional circuit-switched voice networks. Voice over packet solutions (such as VoIP) leverage those data network resources and contemporary packet-switching technologies to generate new revenue opportunities and cost savings for both startup voice service providers (e.g. Vonage), and existing telecom carriers. The result is that telecom companies are frantically investing in VoIP infrastructure equipment more than any other technology. And while proprietary VoIP systems initially helped establish the market, the trend is clearly toward open and interoperable VoIP platforms that easily permit equipment makers and carrier customers to add differentiable value.

Many VoIP gateway vendors are moving in this direction, opting to differentiate through software hosted on VoIP optimized hardware platforms. SBE, for example, has developed

a high-density VoIP gateway engine using Texas Instruments' DSPs with Telogy Software and Wintegra's WinPath network processor to enable integration of voice, data, and fax. This standards-based PTMC module is scalable in density to over 2000 carrier-grade voice channels on a single daughterboard. System designers can now integrate complex gateway blades entirely using off-the-shelf, open-architecture board products in order to provide telecom carriers and service providers with a selection of programmable voice platforms featuring the industry-leading Telogy voice processing software in addition to SBE's line of channelized WAN interface cards.

There are additional benefits to designing with standards-based hardware. Modular VoIP "blades" based around standard, interoperable modules like PMC and AdvancedMC reduce costs by limiting the number of unique blades that telecom OEMs and carriers have to purchase and stock. In a VoIP gateway equipped with transcoding modules, the system could be deployed in a minimal configuration and scaled up later without replacing the whole blade and without taking it off line.

Building VoIP gateways in the context of required interoperability and standards ultimately minimizes network infrastructure costs by facilitating seamless, multi-vendor environments for both media gateways and softswitch/media gateway controllers (see Figure 30). This translates directly to a carrier's ability to generate more revenue per customer by interoperating with leading softswitch, PBX and IP PBX vendors, supporting diverse call control protocols and offering transport-layer-agnostic solutions. As a result, open standard platforms allow seamless support for third-party software applications and services, which is a fundamental driver for VoIP market growth.

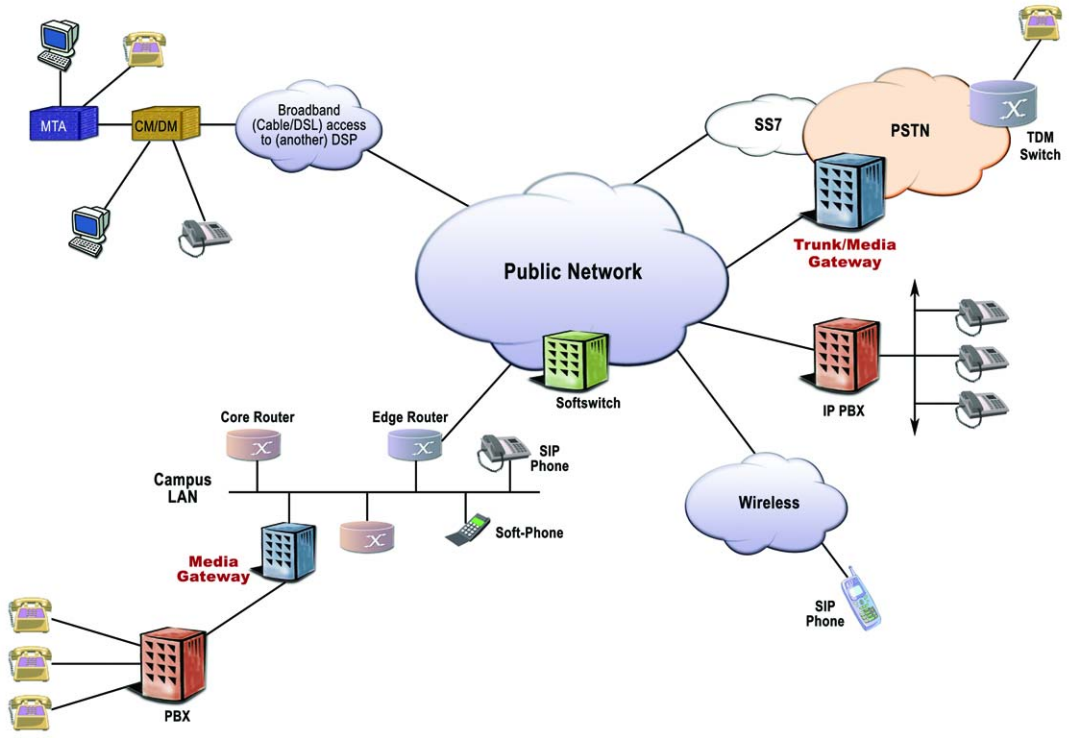


Figure 30. VoIP in the Network

The evolution of telecom equipment

Early on, telecom equipment was very specialized as well as localized. Each company contributed a product based on their core competency. As an example, Company X would build an excellent router and company Y built the physical interface to the outside world. These two disparate appliances would be physically connected to yield a core router.

From there, a form of integration began to take shape. Company X and Y now joined forces to allow one or both to integrate into a single appliance, which was in the past two or more. Of course, the challenge with this arrangement is having enough margins to go around. This spawned the move for Company X to attempt to “do it all.” Except that Company X, although expert in routing, had no competence in physical interfaces. This resulted in sub-par appliances and caused the user of the product much consternation.

Outsourcing solutions that make “cents”

There is no doubt that undergoing a new system design or even a system upgrade can be an intimidating task. So, it is not surprising that there is much talk these days about “outsourcing.” Be it near or off shore, outsourcing almost always implies farming out some piece of design work to another firm. The motivation is simple; reduce the cost of a particular design, minimize the fixed overhead of personnel and accelerate time to market.

The allure of outsourcing is quite understandable, as most firms in the Telco space have lost 50% or better of their staff. To remain viable concerns, these firms must continue to produce competitive products within understaffed environments. Thus, outsourcing is an immediate form of engineer replacement and a way to complete their designs.

The cost of outsourcing must always be taken into consideration and is often understated. Typically, a given project has three cost parameters: direct cost, schedule and quality. Direct cost and schedule are both performance issues and can typically be kept under control via an on-site project manager. However, quality is a much more fickle beast, one that has short and long term effects which can be both good and bad. An on-site project manager may have a difficult time managing quality control at disparate sites.

There exists an alternative to custom outsourcing that can minimize these costs and actually produce a better result than total internal development. This is the selective utilization of off-the-shelf hardware and software specifically designed for seamless integration into a variety of networking applications.

Completing the System Design

So, how does one harness this rich resource of embedded level products, developed specifically by folks who are experts in their field? The likely place to start is with the classic embedded system architecture. Whether you’re building an intelligent brake controller for the auto industry or a Telco grade Class 5 Softswitch, four key components

are necessary for your system: First and foremost is your application, followed by the enclosure, CPU resource, and finally the I/O.

Your application code is clearly the most important piece of the design. This is where your core competency lies and where most of your attention should be focused. Not handling this in-house is likely to reduce your competitive advantage and your ability to maintain that advantage.

The enclosure will most certainly dictate the form factor of the commercially available products used in the system. Selecting a CompactPCI chassis implies the use of CompactPCI cards and gives you the additional option of using PMC cards for extra processing power or I/O. For more cost sensitive products, a PCI platform may be chosen, and PCI cards can be used for the I/O. For future designs requiring much higher bandwidth, an AdvancedTCA platform, with its daughter card technology, AMC, may be your choice.

CPUs have converged into four likely choices: X86, PowerPC, MIPPS and SPARC. Their associated operating systems include: Linux, Windows, and Solaris. Your choice of CPU and OS is largely dependent on history, processing power requirements, power consumption, and real-time requirements. Very likely, your enclosure, CPU selection and OS choice will be made from commercially available products.

That brings us to the I/O. Earlier in the book, we took a deep dive into the most classic of WAN technology, T1/E1, T3/E3, SONET/SDH. These six physical interfaces, along with Gigabit Ethernet are the staples of WAN and Broadband technology. Build a network access device, VoIP Gateway, or router and you will surely be tapping into one or more of these technologies. But just as the CPU and enclosure are likely not in your core competency, and therefore sourced from commercially available products, the I/O is a necessary component that is best left to the experts to produce.

Go to the experts

Today, the solution is to go to the expert for your non-core competency requirements. When Cisco required a Token Ring interface to enable its AGS router to communicate with IBM gear, they didn't hesitate to outsource that to SBE. Same story applies with HP when they required a synchronous serial interface to enable frame relay in their cellular base station. These large companies realized the physical interfaces their end solutions required were not in their core competency, but were absolutely critical in making their product successful. So they went to the leader in WAN devices and allowed SBE to provide the products they required. In addition to embedded WAN and LAN interface products, the SBE product portfolio has evolved to include modular VoIP and iSCSI solutions, designed to enable optimal performance and rapid deployment across a wide range of next generation communications and storage systems. Your project can also benefit from going to the experts. And it couldn't be easier as there is a plethora of standards-based hardware, designed with specific functionality and ease of use in mind, available today.

Testing Issues and Gotcha's

Now that you have built your product, you need to make sure it works properly. You have hardware to test as well as software and are not sure which is working yet. Make sure your software can talk to you hardware. If this does not work, you can't even start testing the software.

Gotcha

Don't try to get your software working until you can read from and write to the framer

The first step is to make sure you can read and write the registers in your framer. This can typically be done without having your system software running, but using a debugger or emulator. Make sure all bits can be written and read back. The test patterns 0x00, 0xAA, 0x55 and 0xff are common patterns to use.

Although most communications devices are byte-aligned, newer devices have 16-bit and/or 32-bit interfaces. In these cases, your processor must be able to talk to devices that have different byte alignments. Some 16-bit or 32-bit devices can only be accessed with a 16-bit or 32-bit bus cycle. Your software and the device driver will have to be written to consider the device width. For example, you can get into trouble if you have a 16-bit device as shown below:

1) You want to read a status register.
2) The status register is 16 bits long and the register clears the bits after you read them.
3) You read only 1 byte (8 bits) of data.
4) The processor will run a 16-bit cycle to the device.
5) So, even though you only wanted 8 bits, the hardware will read 16 bits and clear them all, even the ones that you did not intend to read. You have lost information that can result in your system being in an incorrect state.

Another area that 16-bit and 32-bit devices can get you in trouble is if the device has the ability to read and write bytes and has a bus interface that is 16 or 32 bits wide. Your hardware must provide byte accessibility. However, if the hardware only provides access at 16-bit or 32-bit modularity, then when a one-byte register is written, other registers will changed that the software engineer did not intended to change.

Gotcha

If your framer supports byte accesses, make sure your hardware does also.

Once you can talk to the device correctly, you need to get a signal from your board to the outside world. You will typically have a Line Interface Unit (LIU) or optical module (OM) between your framer and the physical cable attached to your system. The LIU or OM is where the signal from the framer is converted from a string of bits to the appropriate analog signal for the physical medium. You might think that it's a digital interface and that is true, but all digital signals are actually analog. The signal from the framer must propagate through this interface and make it on the cable. There are transformers, protection circuits, clock recovery chips as well as other items that must be traversed before reaching the cable.

Gotcha

Don't use a loopback cable for initial testing

So, how do you know the signal made it on the cable correctly? You connect the signal to a test set. You might think that you can use a loopback cable, but if you can not be sure the signal is making it to the cable, how can you be sure that the signal will be received correctly and your freshly built, untested hardware can detect it? The bottom line is you need a test set. Without one, you will be guessing because some design problems affect both the transmitter and receiver and will go undetected without a test set. If you cannot afford a test set - the right tool to do the job - why are you doing the project at all?

Now that you have a test set connected to your board, you need to tell your framer to send a signal that is easy for a test set to see. Usually, this is an Alarm Indication Signal (AIS). AIS will typically override other settings in the framer, so that you do not need to get all the

bits in possibly hundreds of registers correct. Once you are sending AIS to the test set, verify on the test set that you are receiving the AIS. If you are not, get your hardware designer to verify that the clocks and the data from the framer to the interface circuitry are correct.

If you are not getting AIS at the test set, and the hardware checks out, verify that you have the cables attached to the test set properly. One easy way to test this is to look at the power level being received on the signal. If the signal is –40db from nominal, then you are getting cross talk between your transmit cables and receive cables, or noise out of your system. You are not getting a good signal. You need to get to a signal level that is near nominal levels.

Once you get the AIS to the test set, you need to verify that the frequency of the signal is correct. Telecom signal frequencies are VERY accurate. If the clock frequency that your test set is receiving is off, then you have a clocking problem in your hardware or a configuration problem in your software.

Gotcha

Transmit clocks that don't meet the specifications indicate hardware or configuration problems.

The table below shows the typical timing tolerances you should be seeing:

Interface Type	Clock Accuracy
T1[5]	1.544Mhz +- 50Hz (32ppm)
E1[6]	2.048Mhz +- 102Hz (50ppm)
T3[7]	44.736Mhz +- 895Hz (20ppm)

5 ANSI T1.403 – 1999 Section 5.2.3

6 ITU G.703 – 10/98, Section 9.1.

7 ANSI T1.404 – 1994, Section 5.2

Interface Type	Clock Accuracy
SONET[8]	OC1: 51.84Mhz +-238Hz (4.6ppm) OC3: 155.52Mhz +-715Hz (4.6ppm) OC12: 622.08Mhz +- 2861Hz (4.6ppm) OC48: 2.48832Ghz +- 11,446Hz (4.6ppm)

Again, if your clocks are off by more than this, you have a clocking issue that needs to be solved before proceeding.

Once you get the AIS to your test set, then your transmit hardware path is likely operating correctly. You still have to verify all the functionalities of the interface in the transmit direction.

Gotcha

Verify that the Loss of Signal detector indicates that a signal is present

However, it is now time to go on to the receive direction and verify that you can get traffic from your test set into your board. Start by sending AIS from your test set to the board. All interfaces have a loss-of-signal (LOS) detector and, if the board is receiving AIS, should cause the LOS detector to indicate that a signal is present. On some interfaces, AIS is an unframed signal (T1/E1) while on others it is a framed signal (T3/SONET) so make sure that you are not looking at the loss-of-frame (LOF) indication.

After you get a good signal into the board, you want to verify that you can detect the framing pattern contained in the signal. Change your test set to start sending a valid signal that is, typically, a pseudo-random test pattern. Check your framer device so that you can

8 ANSI T1.105.09 – 1996, Section 6.1

determine if you are getting a valid, framed signal. If you are getting a valid frame signal, your receiver is getting traffic. The next step is to check out both the transmitter and receiver at the same time.

Place your framer into line or remote loopback as shown in the following diagram:

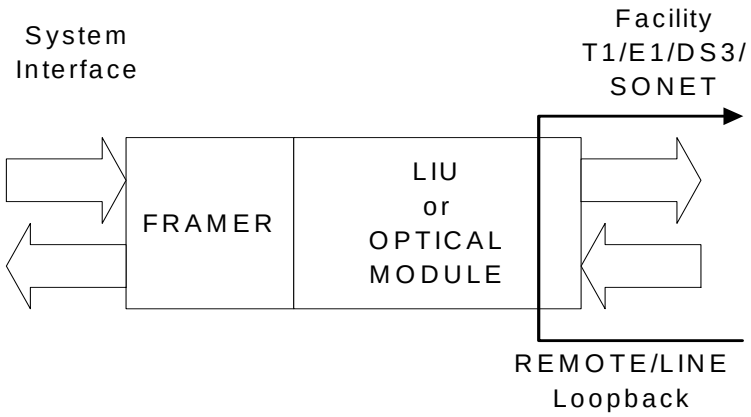


Figure 31. Loopback setup

Using your test set, send a pseudo-random test pattern. Verify that the test set can see the test pattern and the test pattern operates error free. Yes, that is error free. You are in a laboratory environment and you should operate with zero bit errors. If you are having problems seeing a signal, make sure your test set is setup to generate the transmit clock. Whenever you are running a test, you need to be able to trace the clock back to an oscillator somewhere. The oscillator could be on your board, in the test set or available from a system clock source. One source of measurement errors, as well as system problems, is having a facility hooked up where there is no clock source. For example, if your system is

Gotcha

Make sure you have a clock source.

in loop timing[9] AND your test set is in loop timing; the interface will sort of work, but will not function properly. In our experience, this is a very common source of problems.

9 Loop timing is when the receive clock is used as the transmit clock.

Estimating the Development Time for a WAN Access Project

As you know, estimating development times for any project of this magnitude is never an exact science. One major assumption, that most projects make, is that the hardware and software components can be started at roughly the same time. Based on our experience, which now spans three decades, for typical telecom/datacom equipment development, the WAN access card or subsystem project can be expected to look something like the chart below.

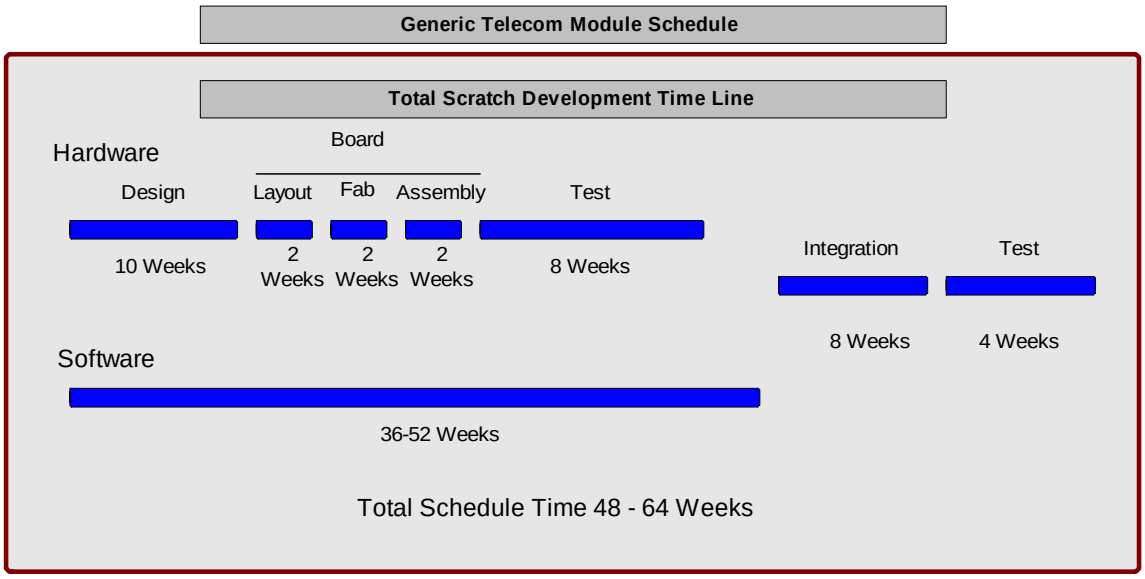


Figure 32. Schedule for Telecom Module

Obviously, any of these components can take longer based on staffing and project magnitude, but if you are starting a project “from scratch”, these are reasonable estimates.

What if you didn’t have to start from scratch on the software side? Suppose the framer driver and the trunk management software components were off-the-shelf? If so, then you

could concentrate all your development effort on your application – the value-added part. Using NComm TMS™, or Trunk Management Software, you can do exactly that and your schedule for the WAN access device would change to look closer to the table below.

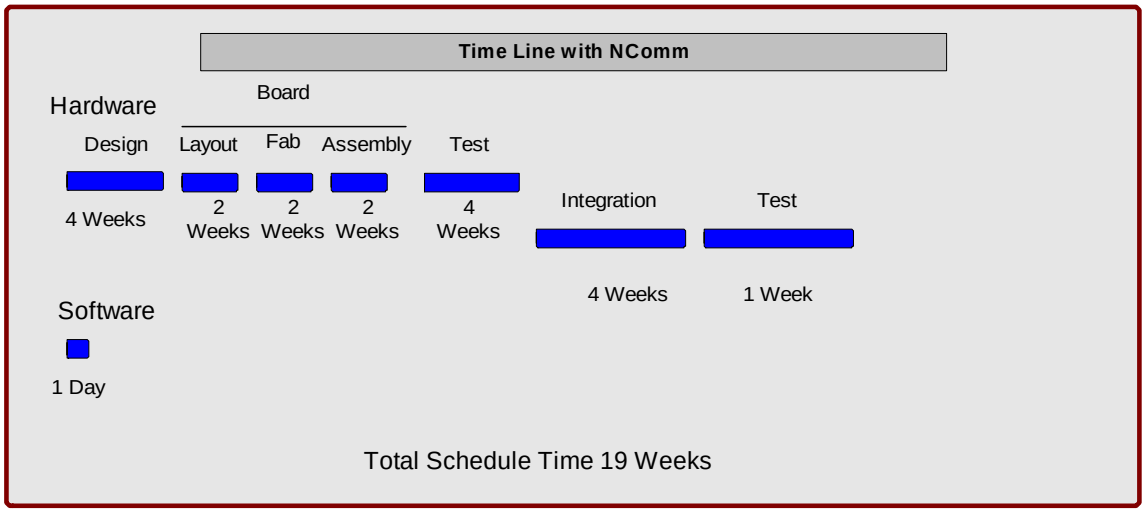


Figure 33. Schedule with TMS

NComm TMS™ is a completely functional suite of T1, E1, T3, E3, SONET (VT-n, STS-n, OC-n) and SDH software providing configuration, alarm, maintenance/performance monitoring, loop back activation, Robbed-Bit Signaling (T1) and Channel-Associated Signaling (E1) functionality. The TMS resides between your telecom application and product above and the framer device driver below. It does not manage the data payload, but does manage the control, monitoring and reporting required to keep the trunk operating as effectively as possible and allows it to function in a standard compliant manner. NComm also has a T1/E1 Line Interface Unit (LIU) package, which when combined with the T3 TMS provides full functionality for an M13 multiplexer application.

Best Options for Development

There are three major reasons why you should consider using NComm TMS™ – time, money, and quality.

1. You collapse your T1, E1, T3, E3, SONET or SDH development schedule from the better part of a year to just a few days, thus gaining time-to-market. First time development of standards compliant programming in-house takes about 2 staff-years and 9-12 months of elapsed time.
2. You spend less money and conserve scarce development resources. Engineering expenses could have been applied to adding value to your product, not developing standards compliant and pre-tested transmission software. Further, such in-house development adds another branch of development activity. This will delay your product release and add risk to whether the software is fully functional.
3. By deploying software that has been used in many applications, widely deployed, tested and retested, and fully debugged, you gain the immediate quality that can only come from years of field deployment. Quality you gain on day one. Savings in support costs you gain every year.

Engineers who have developed trunk management software in the past are quickest to realize the value proposition of not developing it in-house again.

NComm's portfolio of products includes T1, E1, T3, E3, SONET and SDH technologies as well as Primary Rate ISDN. As companies are becoming more global in nature, they are developing product platforms to run on different types of wide area networks. NComm offerings co-exist with each other so that all the combinations may be enabled at either build or run time, increasing the manufacturer's flexibility. NComm TMS™ is framer independent and drivers have already been developed for a variety of devices.

Likewise, TMS™ is designed to be portable to any processor and real time operating system. Pre-defined ports to industry popular real-time operating systems VxWorks (Wind

River), pSOS (Wind River), OSE, Nucleus (Accelerated Technology) and some versions of Linux are standard with each NComm TMS™ suite.

Using NComm's TMS for SNMP Management

NComm provides a set of documents, called MIB Mapping Documents, which will describe how to fill in the functionality of the stub file that is created by the SNMP development environment. These MIB mapping documents will describe which API calls to make to implement the set and get functions that will be required for your device. Since the MIB mapping document is independent from the SNMP protocol, your job implementing the SNMP Agent should be simple regardless of the SNMP development environment chosen. Thus, SNMP management of your device will be greatly simplified by selecting NComm's TMS products for implementing your WAN interface.

Glossary

The terminology found in this document is based on the definitions found in various standards and other ANSI documents. The most commonly used terms are noted below.

Alarm Indication Signal (AIS). A signal transmitted in lieu of the normal signal to maintain transmission continuity and to indicate to the receiving equipment that there is a transmission interruption located either at the equipment originating the AIS signal or upstream of that equipment.

Alternate Mark Inversion (AMI). A line code that employs a ternary signal to convey binary digits, in which successive binary ones are represented by signal elements that are normally of alternating positive and negative polarity and of equal amplitude, and in which binary zeros are represented by signal elements that have zero amplitude. North American implementations use signal elements representing binary ones that are non-zero for only half the unit interval (50% duty cycle).

Asynchronous Transfer Mode (ATM). A multiplexing/switching technique in which information is organized in fixed-length cells with each cell consisting of an identification header field and an information field; the transfer mode is asynchronous in the sense that the recurrence of cells depends on the required or instantaneous bit rate.

B3ZS (Bipolar with 3-zero substitution). A method of encoding a string of 3 zeros by inserting bipolar violations. This is usually used in T3.

B8ZS (Bipolar with 8-zero substitution). An AMI line code with the substitution of a unique code to replace occurrences of eight consecutive zero signal elements. 000VB0VB replaces each block of eight successive zeros, where B represents an inserted

non-zero signal element conforming to the AMI rule, and V represents an inserted non-zero signal element that is a bipolar violation.

Bipolar Violation. A non-zero signal element in an AMI signal that has the same polarity as the previous non-zero signal element.

BIP-8. An error-checking scheme used by E3. This does a byte-by-byte even parity calculation over the previous frame and places the value in the current frame's EM location.

Bit Interleaved Parity N (BIP-N). A method of error monitoring. If even parity is used, an N-bit code is generated by the transmitting equipment over a specified portion of the signal such as a manner that the first bit of the code provides even parity over the first bits of all N-bit sequences in the covered portion of the signal, the second bit provides even parity over the second bits of all N-bit sequences within the specified portion and so on.

Bit Oriented Code (BOC). A message sent over the FDL of an ESF formatted T1 that controls maintenance operations on the T1.

Blue Alarm. An AIS signal.

Bursty Errored Seconds (BES). See Severely Errored Sections (SES).

Channel Associated Signal (CAS). A method of signaling that assigns signaling bits that correspond to their timeslot.

Channelized, Channel, Channel Timeslot. A frame is said to be channelized if the payload timeslots are assigned in a fixed pattern to signal elements from more than one source, each operating at a slower digital rate. T1: the 192 payload bits represent 24, 8-bit channel time slots, making up 24 individual 64kbits/s (DS0) bit streams; each DS0 is referred to as a channel. The eight contiguous digit timeslots associated with a DS0 channel are referred to as a channel time slot. T3: Typically

a T3 will consist of 7 T2 signals with each T2 containing 4 T1 interfaces. For E3, the payload is up to 4 E2s and each E2 can have 4 E1s.

Concatenated Synchronous Transport Signal level N (STS-Nc). A signal constructed by concatenating the envelope capacities of N STS-1s to carry an STS-Nc SPE that transports a super-rate signal. These STS-1s shall be transported as a single entity.

Cyclic Redundancy Check (CRC). A method of detecting the existence of errors in the transmission of a digital signal using polynomial division.

D4 Frame. The fourth generation digital channel bank.

DS1 (Digital Signal 1, T1). A digital signal transmitted at the nominal rate of 1.544 Mbits/s.

E1. A digital signal transmitted at the nominal rate of 2.048 Mbits/sec

Elastic Store/Slip Buffer. Used to adjust for differences in timing between the T1/E1 interface and the system timing.

Errored Second (ES). A one second interval with an error. See AT&T standard TR 54016 (T1), ANSI standard T1.231 or ITU-T standard G.826 (E1/E3)

Excessive Zeros (EXZ). The occurrence of more too many contiguous zeros. For an AMI coded signal, the occurrence of more than 15 contiguous zeros. For a B8ZS coded signal, when more than 7 contiguous zeros occur.

Extended Super Frame (ESF). A DS1/T1 framing format of 24 frames. In this format, 2 Kbps are used for framing pattern sequence, 4 Kbps are used for the Facility Data Link, and the remaining 2 Kbps are used for CRC. A one second interval with an error. See TR 54016.

Facility Data Link (FDL). An embedded overhead channel within the ESF format for T1.

FEAC. Far End Alarm and Control Channel that is used in T3.

Frame. T1: A set of 192 timeslots for the information payload, preceded by a one-digit timeslot containing the framing (F) bit, for a total of 193 timeslots. The payload is often DS0-channelized into 24 channel timeslots. E1: A set of 256 bits organized into 32 timeslots numbered 1 to 32. Timeslot 1 contains the framing pattern, CRC-4, Si bits, Sa Bits, and A bit. When CAS signaling is used, timeslot 17 is used to carry the signaling bits for each channel. T3: A set of 192 timeslots for the information payload, preceded by a one-digit timeslot containing the framing (F) bit, for a total of 193 timeslots. The payload is often DS0-channelized into 24 channel timeslots. E3: A set of 32 timeslots.

Framer Loopback. An internal (within the framer) loopback that tests the path up to where framing is introduced. Used for diagnostics.

HDB3. A zero substitution code used in E1 signaling.

High-Level Data Link Control (HDLC). A very common bit-oriented data link protocol (OSI layer 2), standardized by ISO.

In-Band. Using or involving the information digit timeslots of a frame; i.e., bit assignments of a frame exclusive of the framing bit.

Line. In SONET/SDH, a transmission medium, together with the associated equipment, required to provide the means of transporting information between two consecutive Network Elements (NEs), one of which originates the signal and the other terminates it.

Line Build-Out (LBO). An electrical network used to increase the electrical length of a cable section used in T1.

Line Coding Violation (LCV). The occurrence of either a Bipolar Violation or Excessive Zeros.

Line Loopback. A loopback in which the signal returned toward the source of the loopback command consists of the full signal with (1) bit sequence integrity maintained, (2) no change in framing, and (3) no removal of bipolar violations.

Local Loopback. An internal (within the framer) diagnostic loopback in which the signal returned towards the source is framed.

Loop Down Code. Code sent to disable loopback.

Loop Up Code. Code sent to set up loopback.

Loopback. A state of a transmission facility in which the received signal is returned towards the sender.

Loss Of Frame (LOF) or Out Of Frame (OOF). A framing error occurred; for SONET/SDH the network element is unable to frame align on an incoming signal.

Loss Of Signal (LOS). When no pulses are detected of either positive or negative polarity.

M13 Frame. A T3 framing standard that supports Asynchronous T2 and T1 signals.

Multi-Frame. A method used in E1 to provide CAS signaling.

Out Of Frame (OOF). A framing error occurred.

Path – In SONET/SDH, A logical connection between the point at which a standard frame format for the signal at the given rate is assembled and the point at which the standard frame format for the signal is disassembled.

Path Coding Violation (PCV). See Bipolar Violation.

Path Overhead (POH). Overhead assigned to and transported with the payload until the payload is demultiplexed. It is used for functions that are necessary to transport the payload.

Payload. The information bits of a frame.

Payload Loopback. For T1, a loopback in which the signal returned toward the source of the loopback command consists of the payload of the received signal (with bit sequence integrity retained) and newly generated ESF framing (not necessarily maintaining the integrity of the channel timeslots, frames, or superframes of the received signal.). The newly generated ESF data link contains a valid performance report message with a value of one in every LB-labeled bit position for the duration of the loopback indicating the signal is the result of a payload loopback.

Revertive – In APS, the traffic on a facility will switch back to the original facility once it is recognized as good.

Section – For SONET/SDH, the portion of a transmission facility, including terminating points, between a line terminating equipment (LTE) and Section Terminating Equipment (STE) OR between two Section Terminating Equipments.

Severely Errored Seconds (SES). This is a performance measure. See TR-54016 or G.826 for detailed information.

Signal Bits. Special bits on the T1/E1 used for placing calls.

SLC-96 (Subscriber loop carrier). Another T1 framing format.

Stuffing. A method used in communications to multiplex low-level signals into higher-level signals so that the clock rate and data are preserved across the interface such as in T3 to multiplex T2's into the T3 and multiplex T1's into the T2.

Super Frame (SF). A DS1/T1 framing format of 12 frames.

Super Frame vs. Extended Super Frame: While both formats contain the same number of channel time slots, the SF format is a 12-frame structure while ESF contains 24 frames. Both use the 8th bit of each channel time slot in every 6th frame for

signaling, thus providing the SF format with A/B signaling bits and the ESF format with A/B/C/D signaling bits every multi-frame. In addition, the ESF format uses the F bits to provide frame alignment, CRC-6 check bits, and a 4 kbit/s data link. The SF format divides the F bits into Ft and Fs bits. The Ft bits are terminal framing bits that identify frame boundaries and the Fs bits are signaling framing bits that identify signaling frames.

Synchronous Transport Signal Level 1 (STS-1). The basic logical building block signal with a rate of 51.840 Mbits/s.

Synchronous Transport Signal Level N (STS-N). This signal is obtained by byte interleaving N STS-1 signals together. The rate of the STS-N is N times 51.480 Mbits/s.

T1 vs. E1: The main differences in T1 and E1 are the operating frequency, the number of time slots, the pulse shape, the characteristic line impedance, and the signaling method. The T1 system operates at 1.544 MHz with a total of 24 time slots. The T1 pulse shape contains over and under shoot and is driven on a line impedance of 100Ω. Finally, digital messages to signal on and off hook or other conditions are sent using robbed bit signaling. The E1 system operates at 2.048 MHz with a total of 32 time slots. The E1 pulse shape is a perfectly rectangular pulse shape and is driven on a line impedance of 120Ω or 75Ω. Finally, digital messages to signal on and off hook or other conditions are sent using channel associated signaling.

Yellow Alarm. A Remote Alarm Indication signal. The indication from the far end equipment that it is having difficulties receiving the near end signal.

Zero Destuffing. Used in HDLC packets; remove the zero following 5 consecutive ones.

About NCOMM

NComm, based in Salem, NH, provides turnkey embedded software solutions and hardware platforms that are used by equipment vendors to add WAN interfaces to their products. Developed by NComm's team of engineering and business professionals, our products are designed using the experience obtained by decades of experience in communications software & hardware design and bringing complex products to market.

NComm Trunk Management Software is the WAN de facto standard, embedded by equipment vendors from 3COM to ADC to Sonus Networks and is the most widely used and tested software for WAN overhead management. NComm delivers the underpinning, drop-in software technology necessary to build interoperable, standards-compliant WAN access devices including: framer configuration, alarming & fault management, PMON, line testing, and signaling. NComm's mission is to reduce their client's time-to-market through turnkey T1, E1, T3, E3, SONET, SDH, APS and Primary Rate ISDN telecommunications source code

For more information, call us at (603) 893-6186, or visit www.ncomm.com

About LSI Corporation

LSI Corporation is a leading provider of innovative silicon, systems and software technologies that enable products which seamlessly bring people, information and digital content together. We offer a broad portfolio of capabilities and services including custom and standard product ICs, adapters, systems and software that are trusted by the world's best known brands to power leading solutions in the Storage and Networking markets. For additional information, please visit lsi.com.

This page is intentionally left blank