Video Transfer Application Transport Protocol Design Over ATM Networks

by

Edward V. Sayadian

Project and Report submitted to the Faculty of the

Virginia Polytechnic Institute and State University

In partial fulfillment of the requirement for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

Dr. Fred Ricci, Chairman

Dr. Steve Heppe

Dr. Joesph Knight

April 11, 1994

Falls Church, Virginia

Video Transfer Application Transport Protocol Design Over ATM Networks

by

Edward V. Sayadian

Committee Chairman: Dr. Fred Ricci
Electrical Engineering

(ABSTRACT)

This report analyzes the transport layer protocol functional requirements needed to support a video file transfer application over the proposed Broadband Integrated Services Digital Network (ISDN) network. The B-ISDN network will be capable of supporting very high speed transmissions (up to 100s of Mbps to individual users). The Asynchronous Transfer Mode (ATM) protocol is intended to be the transmission and switching standard upon which the next generation high speed integrated network will be based. The transport layer protocol supports the end-to-end communication channel and provides the application access to the B-ISDN network.

The video transfer application will essentially act as a file transfer service, where the file will be a digitized compressed form of the video. This file must then be transferred in a reasonable time with a very low probability of error (a requirement of the high compression ratios). High compression ratios are desired to minimize file length, transmission duration, and buffer requirements. The unique transfer requirements for the video transfer application are described in Section 3.0.

In order to achieve the application requirements, the transport protocol must minimize overhead, maximize the fixed application transfer data rate, and make full use of the ATM functions. Section 2.0 of this report provides a overview of B-ISDN and insight into the ATM protocol functions and processing times.

As a result of the transport layer analysis, a dedicated video transfer application transport protocol can be designed to operate without additional overhead fields other than what is provided by ATM. To minimize transmission duration, the file transfer data rate is fixed at 60.16 Mbps, this rate is then subdivided into sub-channels in order to utilize the ATM modulo 7 sequencing number header field. The transport protocol will use a Go-Back-7 window error control algorithm assigned to each sub-channel. With the low error rates associated with fiber transmission (or non-fiber transmission system with similar BER values), the probability of exhausting the window, and hence decreasing the transmission rate, is insignificant with respect to the overall transmission duration. Section 4.0 of this report, analyzes each transport layer function and describes its implementation tailored to supporting the video transfer application. Numerical analysis and simulations support the conclusions derived in this section.

Current transport protocols, primarily designed for circuit switched or datagram environments, are designed for relatively low speed, high error rate transmission systems. As a result, current transport protocols add a significant amount of overhead, hence increasing overall transmission duration and processing complexity. From a functional standpoint, it is concluded that the unique transport protocol design specified in Section 4.0 is superior to current/modified transport protocols in terms of resource utilization, processing simplicity, and total transmission duration.

Table of Content

1.0 Introduction 2

2.0 The ATM Based B-ISDN Network 3

2.1 Physical Architecture 3

2.2 Functional Architecture 4

2.3 Network Interfacing, SDH and ATM 7

2.3.1 Physical Layer (SDH) 7

2.3.2 ATM Layer and AAL 18

2.3.3 Call Routing and Addressing Via VPI and VCI Fields 25

3.0 VTA Application Data Transfer Requirements 28

3.1 Data Compression 29

3.2 File Transfer Data Rate Requirements 33

3.3 VTA Buffer Requirements 34

4.0 VTA Transport Protocol Analysis 35

4.1 Transport Protocol Requirements 35

4.1.1 Error Control 35

4.1.2 Multiplexing 47

4.1.2.1 Demultiplexing 47

4.1.2.2 Remultiplexing 59

4.1.3 Call Set-up 61

4.1.4 Other Transport Protocol Functions 66

Table of Content (continued)

4.2 Current Transport Protocols 67

5.0 Conclusion 72

Appendix A Link Analysis 73
Appendix B Markov State Probability Analysis 85
Appendix C References 91
Appendix D Acronyms 92

Video Transfer Application (VTA) Transport Protocol Design for ATM Networks

Wide area video on demand systems have recently received extensive analysis. With the continuing development of Broadband Integrated Services Digital Naetworks (B-ISDN) technology, many potential applications/services are being proposed which can utilize the high data rates offered by the network. This report will examine a video on demand system, referred to as the video transfer application (VTA), which will be able to transfer videos (movies) from a video provider facility (facilities) to residential users. The providers will store large amounts of compressed video data in various data base buffers and download videos upon residential user requests. The provider tracks the requests per user per month and then bills the user much like the phone company does today. This system is proposed to supplant the video rental scenario by allowing residential users to order videos electronically without having to visit the video provider facilities. Furthermore, this system can provide 100% availability of all videos independent of demand. The means in which the providers and user will communicate will be through the B-ISDN which will utilize the asynchronous transfer mode (ATM) protocol stack. This is based on the assumption that a ATM cell relay based network will be the foundation of the next generation high speed integrated public network. The VTA will utilize the ATM-based network to facilitate its communications and data transfer needs. The majority of the ATM network has been defined by the Consultative Committee on International Telegraphy and Telephony (CCITT). The design of the transport layer protocols, whose functions are to interface applications to the ATM-based network, are still largely undefined. This report will analyze the current design state of the proposed B-ISDN network and determine the VTA data transfer requirements in order to define the transport protocol requirements and functions. This information will serve as a basis to design a VTA specific transport protocol and compare its functions to existing transport protocols. The existing protocols will be compared to the VTA specific protocol in order to identify which, if any, can be modified to efficiently support the VTA requirements. The conclusion will state the minimum operational design functions of the VTA specific transport protocol and the most efficient implementation strategy.

1.0 Introduction

The VTA application will require a very high channel data rate in order to transfer the video data in a reasonable time duration. Building a separate network to support the VTA is economically unfeasible. Using current low speed local loop links to residential users, limits the video downloading procedure to real time transmission and puts considerable processing constraints over the network. The most efficient and user convenient scheme seems to be implementing the application over the proposed B-ISDN. The speeds and services offered by the B-ISDN are on the order of 100's of Mbps to the user. Utilizing these data rates, it is possible to transfer a video within minutes while simultaneously allowing other residential user applications to operate. Once the video is transferred and stored in the user end buffer, the user can view the video at his/her leisure. The design of the VTA transport protocol requires the examination of the VTA data transfer requirements. The ATM network functions must also be examined in order to design a transport protocol that will fully utilize these functions. This report is presented in three parts; part 1 will be a overview of the B-ISDN network and ATM protocol stack. In this section, the ATM protocol (lower layer) functions will be defined. It is important to define the data structure and network abilities in order to design a transport protocol that will fully utilize the ATM functionality (not duplicating functions) while maintaining support of the VTA requirements. Part 2 will examine the VTA data transfer requirements. This will include defining the VTA data compression standard and the overall VTA transmission data rate. The compression technique, in turn, defines the maximum acceptable error rate that VTA can tolerate while providing a VCR quality picture. Part 3 will examine the transport protocol. This section defines the transport layer functions needed to interface the VTA application to the lower network layers (ATM protocol). Each of the standard transport protocol functions is examined and designed to maximize VTA video file transfer operation. Existing transport protocols are then compared to determine if any existing protocols can be used directly or modified slightly to support the VTA system.

2.0 The ATM Based B-ISDN Network

2.1 Physical Architecture

B-ISDN is a cell-relay based network proposed to supplant the current public telephone network. The network is proposed to support all application data in an integrated fashion over digital links. Users can subscribe to services through the network (i.e. phone service). Recent breakthroughs in information transfer mode techniques have lead to the emergence of the ATM standard. ATM is now recognized as the standard upon which B-ISDN should be built, and in 1990 the first set of CCITT recommendations for B-ISDN was approved. The VTA will be a service implemented at the residential user and video provider facilities over the B-ISDN. The provider service will store videos, in compressed state, distributively among a set of video provider facilities and will serve a defined area of users. Each user and video provider facility will be connected to the network through an ATM interface module connected to a local loop optical fiber. The bandwidth of the local loop fiber can be ordered according to specified interface rates given by the Synchronous Digital Hierarchy (SDH) standard (current physical interface standard defined by CCITT). The interface service rate is chosen by the user based on user needs (both current and future). The local loop bandwidth or data rate can be used dynamically by the user applications depending on the number of applications running simultaneously and the required data rate for each application. This flexibility can only be realized by implementing the user-network interface functions at the user premises [3]. Figure 2-1 depicts a generic network topology where each residential user and video provider facility is connected to the network.

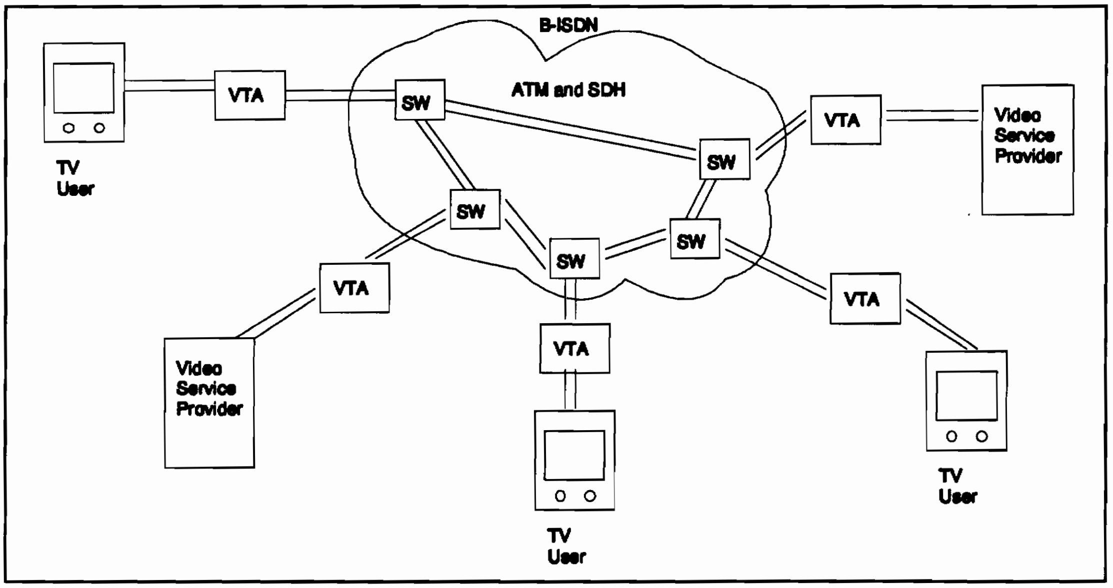

Figure 2-1 Generic Network Topology

Public switching ATM nodes will provide the connection routing through the network. Geographical limits and population densities will limit the coverage area of a set of video provider facilities. The required end-to-end BER probability limits the connection length (maximum length will depend on implemented fiber parameters) hence placing a limit on the geographical bounds of the service. The number of simultaneous users a provider facilities is capable of supporting will limit the overall number of users connected to a provider set. Once in the network, the connections can be routed to any other network user or interface. The proposed VTA service will use the B-ISDN much like a telephone unit uses the current telephone network. The difference being that the B-ISDN can support many very high data rate connections simultaneously while maintaining high throughput. These features are the result of synchronous optical transmission technologies, the ATM transfer protocol (enhanced routing scheme using virtual channel identifiers (VPI)/ virtual path identifier (VCI) and the virtual elimination of link to link overhead), high speed-light weight transport protocols, and applications that take advantage of the network and interface capabilities.

2.2 B-ISDN functional Architecture

The functional architecture of the network is layered, where each layer is assigned functions as appropriate. This simplifies the design, development, and operation of the network, and allows smooth network evolution. The network will provide the VTA and other applications with a transparent transmission link upon which any number of channels may be supported simultaneously as long as the user local loop can support the congregate data rate. Figure 2-2 shows a high view of the functional layers between a video provider and residential user.

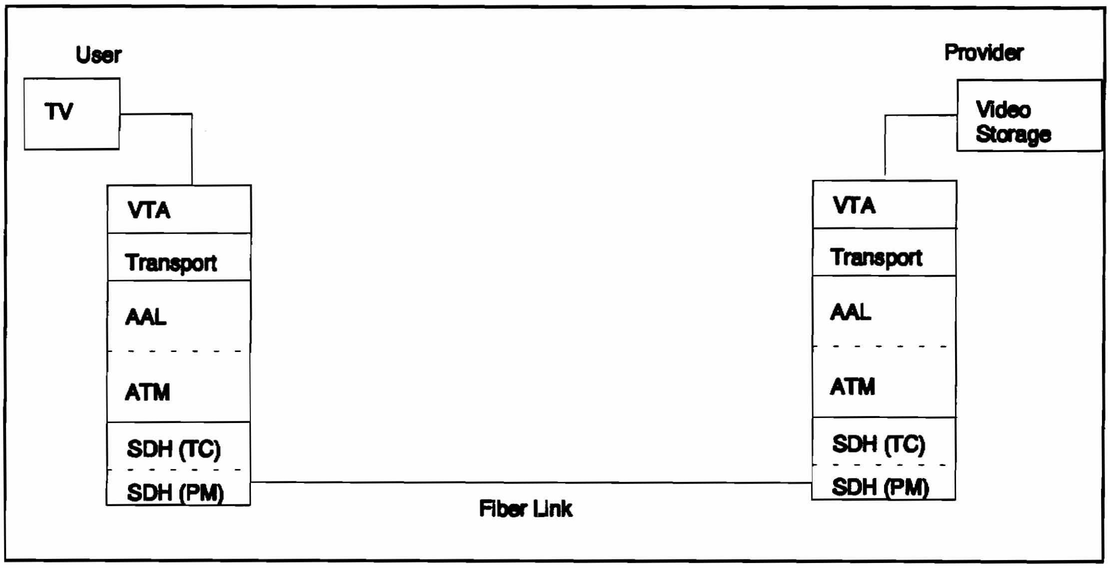

Figure 2-2 Functional Layers

The VTA layer implements the functions unique to the video transfer service, such as video request processing, data decompression, and video buffering. The primary function of the ATM adaption layer (AAL) and ATM layer is to map channel information into the ATM protocol cell format (shown in Figure 2-14). The SDH layer provides the physical means of transporting the ATM cells. The transport layer then must function to interface the VTA to the AAL and ATM layers. The transport layer must provide for the end-to-end functions that are not supported by the lower layers and deliver data to the lower layers in the correct format. Once within the network, all channel information is transported via ATM cells. This is a change from early network transmission which transmits information over time slot multiplexed links (on a byte level), introducing huge multiplexing/demultiplexing tasks upon the switching nodes when channels of various data rates are present. Using ATM cells however, each channel is broken down into ATM cells at the user-network interface, hence, the switching scheme works on a cell basis and requires no hierarchical multiplexing/demultiplexing stages. This, coupled with relieving the switching nodes of overhead tasks concerned with the end-to-end link (error detecting, ACK-NAK reporting, etc.) greatly increase the switching speed throughout the network. Figure 2-3 shows the functions associated with each layer in the protocol stack [5]. Each layers functional capabilities and constraints are examined in detail in the following sections.

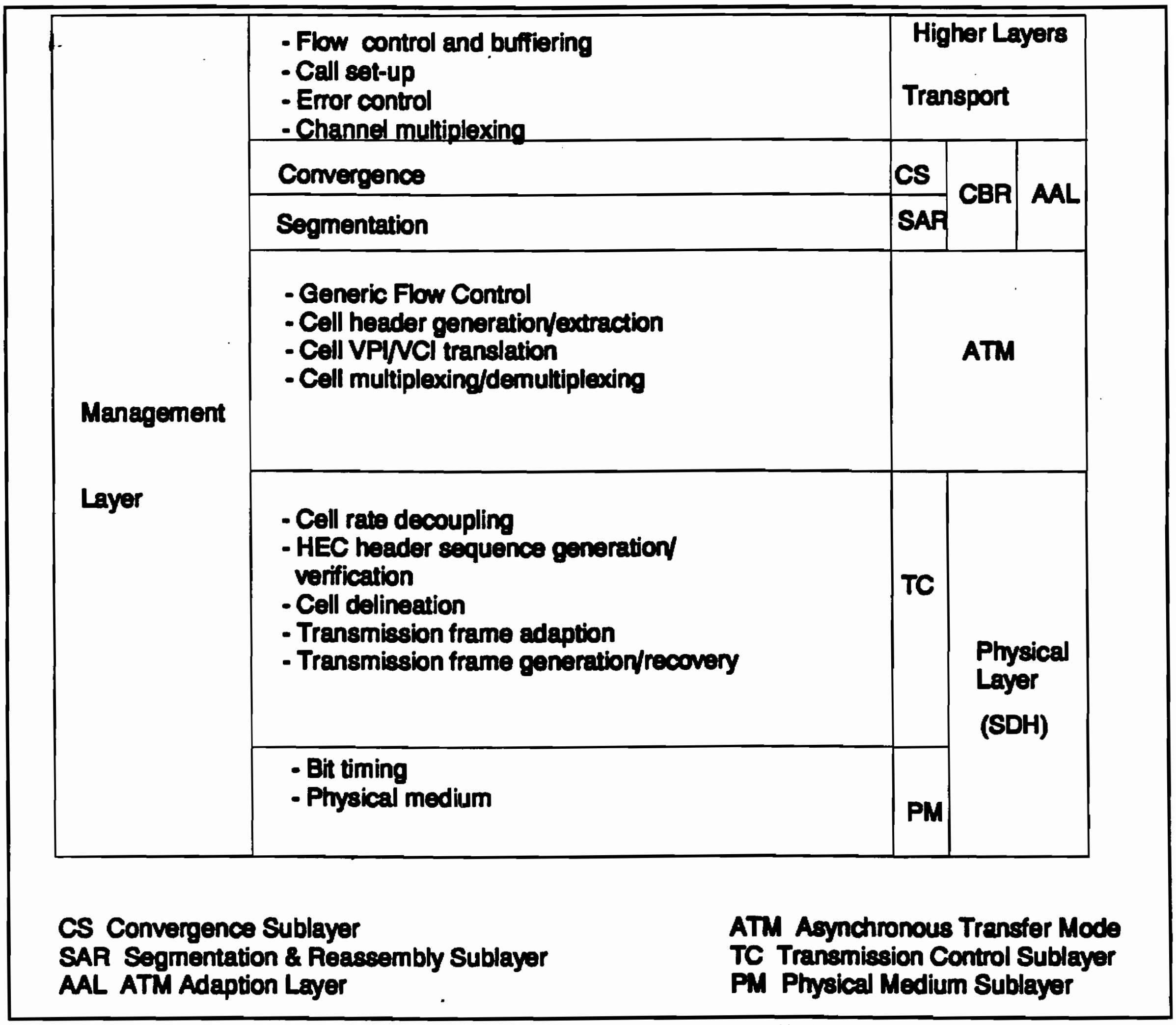

Figure 2-3 Functional Layer Responsibilities

2.3 Network Interfacing, SDH and ATM

The interfacing scheme to the network is global to all applications. The network interface layers consist of the AAL, ATM, and physical layers. These layers function to convert the transport virtual channel data stream into the ATM protocol format (depicted in Figure 2-14) and provide for a means of digital transmission. Examining the capabilities and constraint of the interface layers will allow the VTA transport protocol design to efficiently use these capabilities while staying within the constraints.

Before the ATM protocol stack is examined in detail, it is noted that there are still issues yet to be resolved before the ATM based B-ISDN network can assume the role of the universal transmission system for all applications. Table 2.1 lists these issues.

Table 2.1 ATM Issues

Major issues still unresolved with ATM:	Comments
- Congestion control over entire network	Network issue , must examine all applications to resolve
- Traffic framework to manage various application data with different service requirements	Network issue , must examine all applications to resolve
- Transport protocols to take advantage of high speed network	Subject of Section 4.0
- Switch fabrics with multi-cast capabilities	Switching design issue - switching durations must be on a cell basis with speeds of ~ 1 Gbps

2.3.1 Physical Layer (SDH)

The primary function of the physical layer is to provide the electronic connection between network users. The physical layer merges user cells with the transmission overhead to generate a continuous bit stream across the physical medium. The CCITT has adopted the SDH network node interface standards (Recommendations G.707, G.708, G.709) as the transmission interface (physical Layer) to the B-ISDN. The SDH is a synchronous interface with the following features:

- Transports ATM cells or synchronous data channels in frames. The protocol used to define the frames is the synchronous optical network (SONET) standard.
- Frames are divided into two entities, overhead and information payload. The information payload carries virtual channel information in the form of ATM cells; or synchronous data channels, which it bundles into high and low order paths.
- Frame overhead is divided into section and path layers for transmitting operation and maintenance (OAM) information, thus allowing enhanced network management functions
- Frame overhead uses pointers for synchronization; the pointers isolate the various virtual channels from the transmission frames and allows wander and jitter to be accommodated.

The use of high order and low paths basically multiplexes many lower order synchronous data rate channels with the same destination into one higher composite data rate channel and treats that data rate throughout the network as if it were one channel instead of many smaller channels. SDH defines two composite path data rates (high and low order) for various lower channel rates.

By implementing ATM over the SDH standard as opposed to synchronous data rates, the throughput can be increased since the ATM layer will handle the "multiplexing" by defining cells and sending them as needed to support the specific application data rate.

Using the SDH frame standard, the physical layer is responsible for collecting ATM cells, assembling the SONET frame, and sending it out over the link in a synchronous digital fashion. Since the B-ISDN operates in the ATM cell mode for the VTA, the only responsibility of the SDH interface will be to map ATM cells (all applications will be converted to the ATM cell structure prior to the physical layer) into the frames upon transmission and removed them upon reception. The SDH interface standard is separated into two sub-layers:

Physical Layer Medium (PM): Functions include bit transmission including generation and reception of waveforms, insertion and extraction of symbol timing information, and electrical-optical and optical-electrical transformations.

Transmission Convergence (TC): Functions include header error checking (HEC) generation and verification, frame and cell delineation, time coding, integration of idle cell to present links with continuous bit flow.

The SDH interface defines several service rates (bandwidth of the fiber links):

- synchronous transport signal (STS-3) 155.52 Mbps
- STS-6 311.04 Mbps
- STS-12 622.08 Mbps
- STS-24 1244.16 Mbps
- STS-36 1866.24 Mbps

The standard SONET frame protocol for a STS-3 service is described below [1].

SONET (STS-3) provides a physical payload envelope through a frame structure (SONET). The protocol uses 720 bits for overhead and leaves 18720 bits for the information field (payload). A frame is sent every 125 μsec. Therefore,

$$\text{Overhead Rate} = (\text{number of bits of overhead per frame}) \times \frac{\text{frames}}{\text{second}}$$

$$= (80 \times 9) \times \frac{1}{125\ \mu\text{sec}} = 5.76\ \text{Mbps}$$

$$\text{Data Rate at ATM Layer} = \text{service rate} - \text{overhead rate}$$

$$= 155.52\ \text{Mbps} - 5.76\ \text{Mbps} = 149.76\ \text{Mbps}$$

The TC layer frame overhead which includes:

- Header error check (HEC) generation/verification (polynomial of degree 31 (32 bits of the frame));
- Cell framing indication (used to indicate all start locations in adjacent frames);
- Cell scrambling/descrambling (randomization of cell payload to avoid delineation patterns);
- Cell delineation;
- Path signal identification;
- Frequency justification/pointer-processing;
- Multiplexing (within network);
- Transmission frame generation/recovery;
- OAM function.

For this report it is assumed that residential users will be connected via a full duplex 155.52 Mbps service and the video providers will be serviced by several full duplex 622.08 Mbps services. Note, higher data rate service will be common in the future, however, since all the rates are multiples of the standard STS-3 rate, the combination of several STS-12 services can be substituted by a STS-36 rate for example. The overhead discussed below is also linearly proportional (1:1) with the STS standard. For instance a STS-3 service frame has 9 columns of overhead and a STS-12 service frame has 36 columns of overhead.

Both sub-layers of the SDH interface are examined to estimate the processing delay. The bit rate into and out of the interface (and link) are continuous (since the SDH is a synchronous interface), therefore the clock and synchronization functions can be maintained. The PM sub-layer receiver block diagram is shown in Figure 2-4. The PM sub-layer function is to basically convert the optical signal into electrical binary data and maintain bit synchronization.

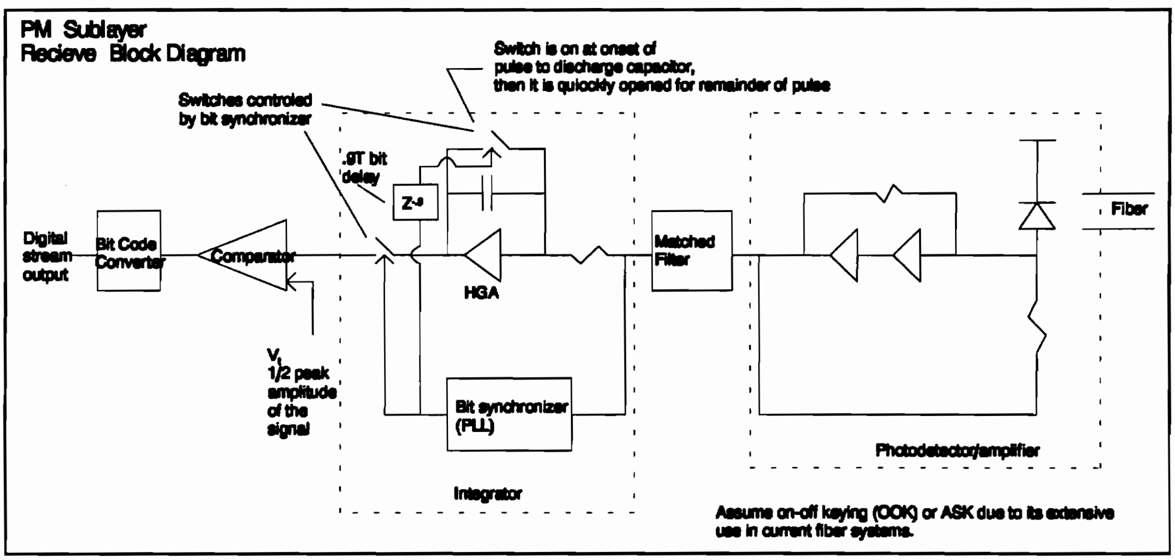

Figure 2-4 PM Sub-layer Receiver Block Diagram

The transmitter uses a laser diode because of the relatively high power output (+5 dBm) including coupling loss, and large modulation bandwidth, about 1 GHz. On-off keying (OOK) modulation is obtained by simply switching the bais current on and off. Figure 2-5 shows the functional design of the PM sub-layer transmitter. The PM sub-layer does not contribute any processing delay or provide any functions other than interfacing the TC sub-layer to the fiber.

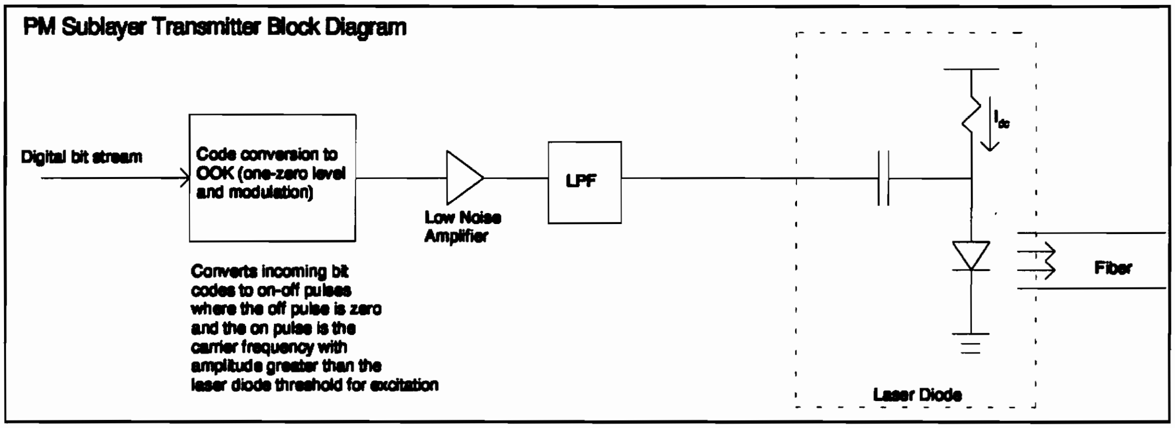

Figure 2-5 PM Sub-layer Transmitter Block Diagram

The TC sub-layer supports the SONET frame assembly (at transmitter) and disassembly (at receiver). The standard SONET frame protocol format is shown in Figure 2-6 [8].

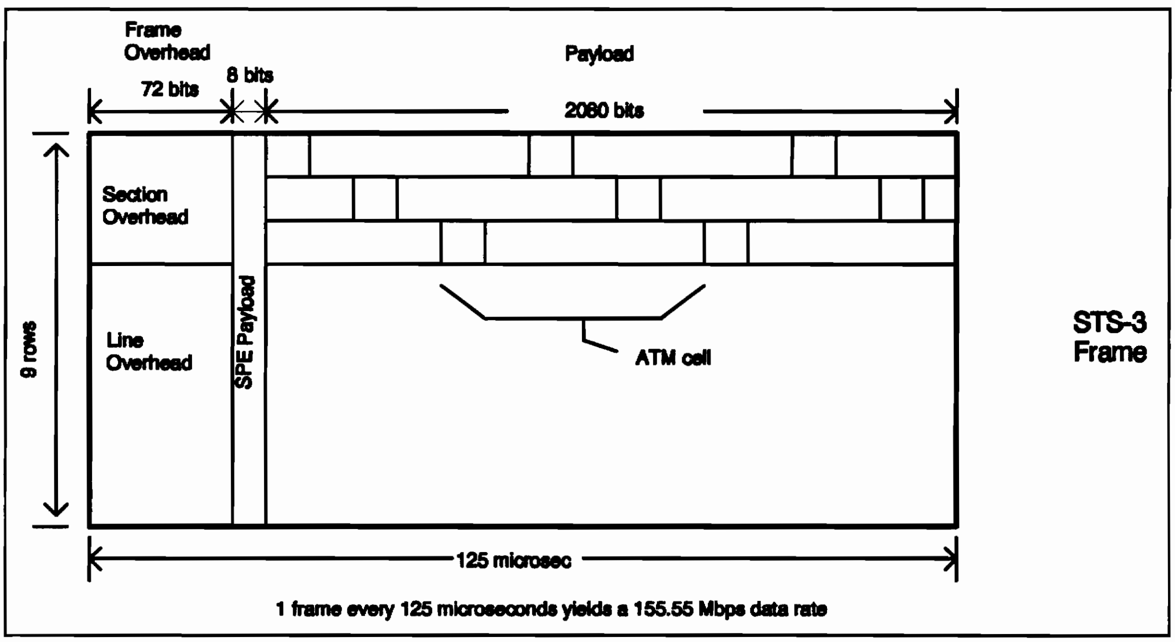

Figure 2-6 SONET Frame Format

The frame will come in and go out of the TC layer row by row. Hence the overhead will not be received in one piece. The TC sub-layer, however, will buffer the data logically as shown in Figure 2-6. The overhead functions associated with the SONET frame are described below:

The SONET frame overhead for a standard STS-3 service interface has a byte oriented structure. The frame is made up of 2430 bytes. The bytes are often viewed functionally as depicted in Figure 2-6. Rows 1-3 of the first 9 columns (bytes) of the frame are used for section overhead and rows 4-9 of the first 9 columns are used for line overhead. The SPE payload is used to support synchronous data rate channels and serves no function when transmitting in the ATM cell format. The functions for the various bytes in the overhead are given in Table 2.2 [8].

Table 2.2 SONET Fields

Bytes		Names and Functions
Rows 1-3	Rows 4-9	
6	-	(A_1, A_2) for frame alignment (frame delimiter)
1	3	(B_1,B_2) HEC for frame overhead
1	-	C_1 Frame identifier (frame sequence number (SN))
3	9	(D_{1-3} , D_{4-12}) Data comm channels for remote monitoring and control, and interoffice data comm respectively
1	1	(E_1 ,E_2) section maintenance communication channels
-	9	H pointer rows (payload pointers)
-	2	K_1 , K_2 automatic protection switching signal
-	6	Z not yet defined
14	24	Bytes for national use
27	54	Total

Rows A of the overhead uses 48 bits or 6 bytes for framing, i.e. some type of delimiter. Hence the TC layer upon receiving the bit stream will look for this delimiter to indicate the start of the frame. Once the frame is detected, a signal is passed to the management layer for clocking purposes (so that the management layer is in frame sync w/ the incoming stream). As a result of the delimiter, the TC layer must (upon transmission) insure no such bit pattern is contained in the payload. Hence scrambling/descrambling is introduced in these circumstances.

Rows B_1 and B_2 will contain the framing overhead HEC generated from a standard cyclic redundancy check (CRC) polynomial of degree 31 [1]. Upon transmission of the frame, the overhead sections are assembled and the HEC code is then calculated. The process is implemented as a dividing circuit consisting of exclusive OR gates and a string of 1-bit storage devices, as shown in Figure 2-7 [5].

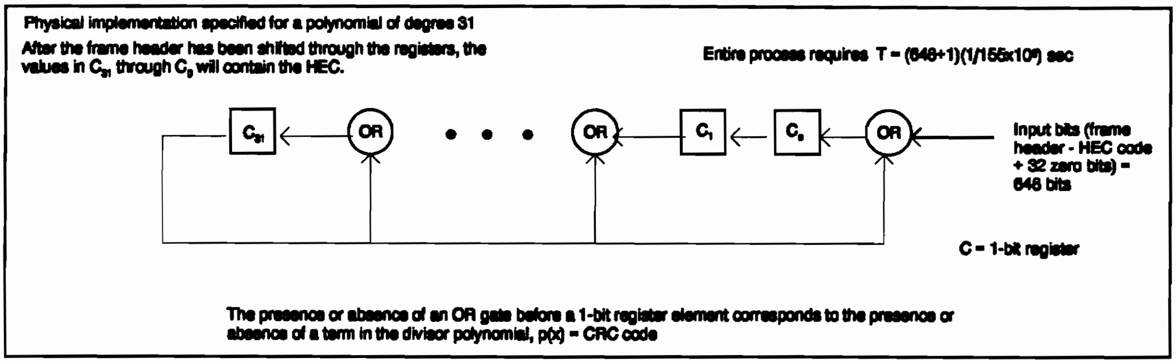

Figure 2-7 HEC Generator

At the discrete instances of the bit clock (clock works on bit rate, e.g. 1/155.52 Mbps for STS-3 service), the values of the 1-bit registers are replaced by their input values, causing a 1-bit shift along the entire register. The register contains 32 1-bit registers (length of the HEC code). Exclusive- Or-gates are placed according to the CRC divisor (this is fixed for all frames). The frame overhead plus 32 zeros are shifted sequentially (by row) through the register, initially all registers are clear. After the entire overhead and extra zeros are shifted into the register, the values of the 1-bit registers will be the HEC code. Upon the next clock count, this code is then uploaded into the frame overhead and inserted into the header. For a STS-3 frame, this process requires a bit synchronous clock (period = $1/155.55 \times 10^6$), 32 1-bit registers, and a number of exclusive OR-gates corresponding to the CRC polynomial. The receiver uses the same logic to check the HEC code in order to determine if any errors occurred during transmission. The physical implementation is shown in Figure 2-8 [5].

Upon reception, each received SONET frame is buffered. The SONET frame overhead and HEC code are then downloaded, in serial, into the error checking processor. The frame overhead (not including the HEC code) is sequentially shifted through the register. If there

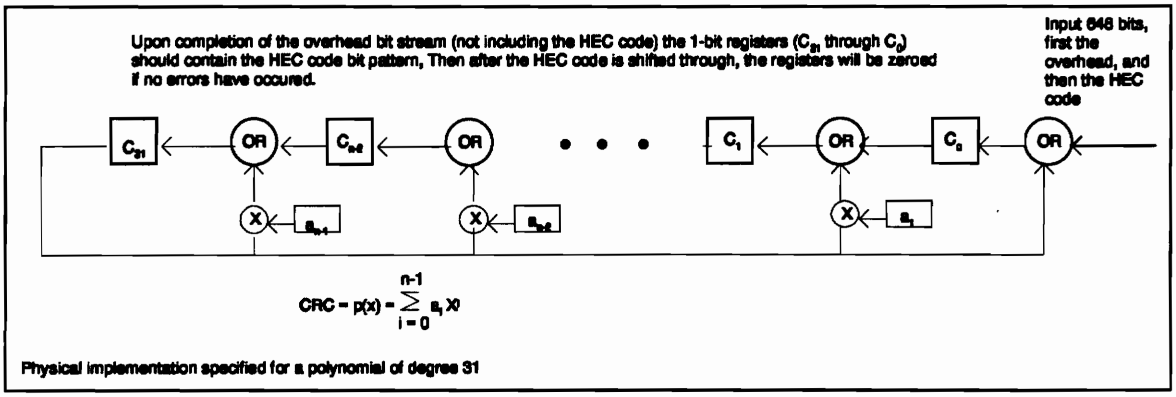

Figure 2-8 HEC Checking Block Diagram

have been no errors during transmission, the 1-bit shift registers should contain the bit pattern of the HEC code. The HEC code is then shifted through, which effectively zeros out the 1-bit registers. Therefore, a error-free cell transmission will result in all the 1-bit registers containing zero values. For a STS-3 frame, the receiver process requires a bit synchronized clock, 32 1-bit registers, 32 exclusive OR-gates, and a number of 1-bit buffers and AND-gates which correspond to the CRC-polynomial (fix for all frames).

Row C_1 determines the frame SN (modulo 2^8 -1 = 255) [8]. At the transmitter, the TC sub-layer will utilize the frame clock (125 μsec) to preform this counting function. The sequence number (SN) is then inserted into the frame header. At the receiving side, a copy of the SN is downloaded into a SN header field subprocessor and sent to the management layer to keep track of error free frames, if the management layer notices a skip in the received SN count, it will report this to the active applications as well as the network (via the management channel).

Rows D and E are used for management control, these are internal to the interface and provide for performance reporting and inter-office communications. Row K is for switching protection.

Row H is used for payload pointers. This function points to the beginning of the first ATM cell in the payload. The pointer function was primarily inserted into the SONET frame to be used for synchronous data transmission. For example, if the SONET frame were to transmit a

DS-1 signal, the pointer would define the number of bytes between the H row and the appropriate synchronous payload envelope (SPE); the SPE is a header associated with the synchronous channel. Once the first byte position is know, the DS-1 channel can be removed from the frame as shown in Figure 2-9 [4].

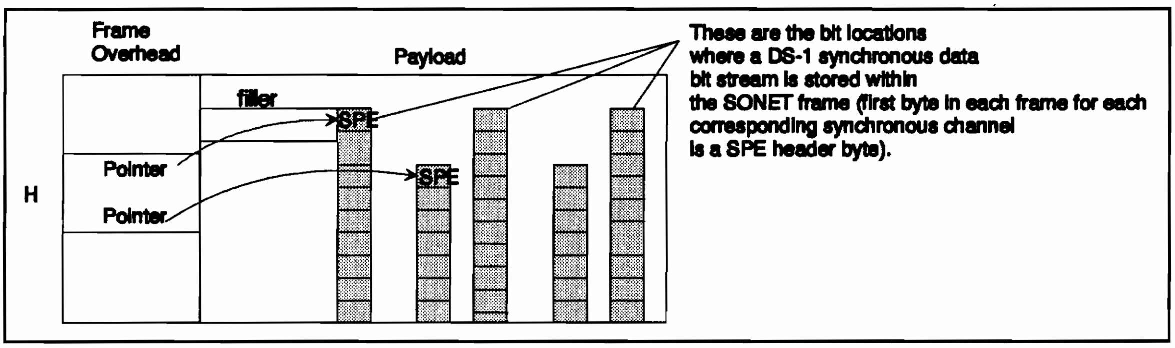

Figure 2-9 Pointer Function

For the ATM cell transmission case (used for the VTA), the pointer identifies the number of bytes between row H and the first ATM cell. Once the first ATM cell boundary has been identified, CCITT Recommendation I.432 defines how the HEC field of the cell (not the frame) can be used to define the cell boundaries of the remaining cells. The entire payload is serially shifted into a delineation processor (defines all boundaries). The processor counts off the number of bytes to the first ATM cell, then it uses the header error checking implementation described earlier to divide the cell header with the standard CRC-8 polynomial defined for ATM cells. The remainder (values in the 8 1-bit registers) should be zero. If this is the case, the next 48 bytes are skipped and the cell is defined. The processor then hunts for the next 5 byte sequence which produces all zeros in the 1-bit registers, in this manner the cells boundaries are defined. Figure 2-10 shows the delineation functional design.

In this manner, only error-free cells are passed up to the ATM layer for processing, thus this function serves not only to delineate the cells, but also as the error checking function at the ATM layer.

At both the TC layer transmitter and receiver, SONET frames will be sent or received (in serial) every 125 μsec. At the transmitter, a buffer will collect the ATM cells during the 125

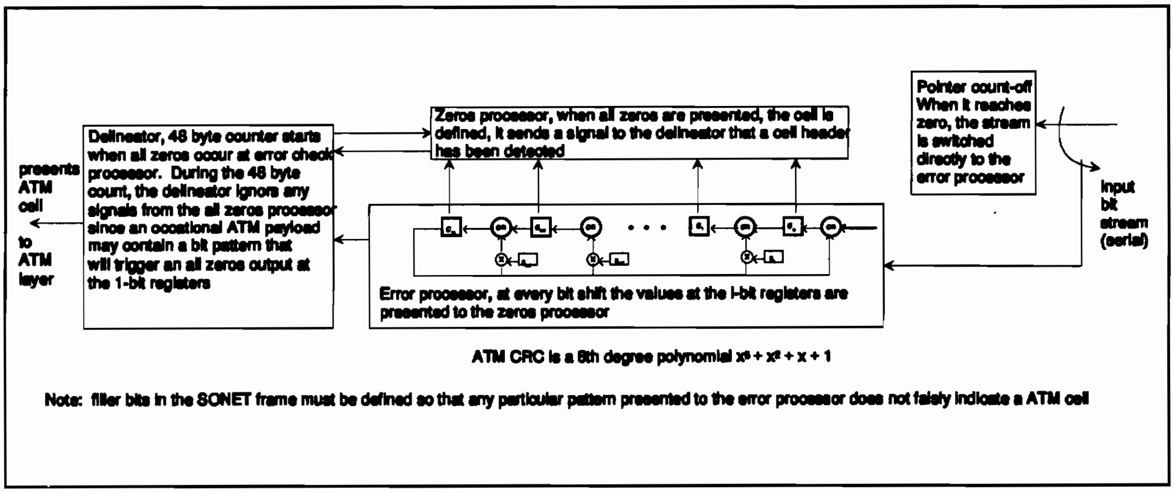

Figure 2-10 Cell Delineator Block Diagram

μsec clock count. The framing header needs to be assembled along with the HEC code during this time period. On the next frame clock count (125 μsec) the buffer (payload and header in tact) will, in parallel, download the entire frame to an output buffer. This output buffer will send, in serial, the frame out over the link at the specified interface rate (155.55 Mbps for STS-3). Therefore, at the end of the second clock count the output buffer will be empty and the next frame (already assembled) can be downloaded, in parallel, to the output buffer. The transmitter block diagram is shown in Figure 2-11 [12].

At the receiver, an input buffer collects the SONET frames. A delimiter checking module scans the input bit stream as it enters the TC sub-layer and signals to the input buffer when a SONET frame boundary has been detected. The input buffer then collects the next 19440 bits, which defines the SONET frame. The delimiter module, upon a SONET frame detection, establishes the frame sync clock (125 μsec). Frame sync is maintained since frames are detected every 125 μsec (SDH is a synchronous interface). Once an entire frame is collected at the input buffer, the buffer downloads the frame, in parallel, to a processing buffer. The input buffer then begins to collect the next frame. The processing buffer, in the mean time, has one frame clock count before the input buffer will download the next SONET frame. The processing buffer then has 125μsec to process the header and send, in serial, the frame payload (information field) to the ATM cell delineation processor. The receiver block

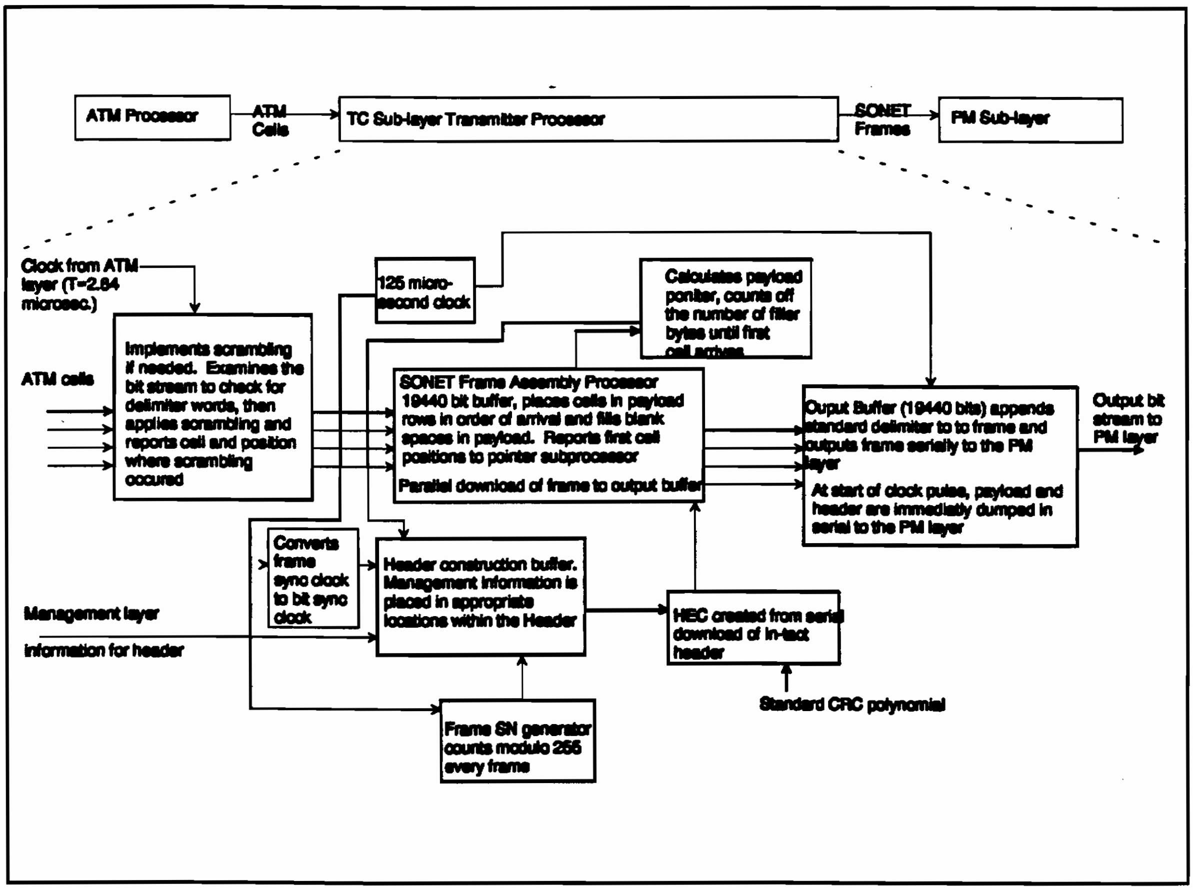

Figure 2-11 TC Sub-layer Transmitter Block Diagram

diagram is shown in Figure 2-12 [12].

At the TC processor buffer, the header is extracted and processed while the payload is uploaded in serial to the cell descrambling unit. The descrambling header field must be processed prior to the first bit of the payload entering the descrambling processor. The entire frame is 19440 bits, of which, only 720 of these bits are dedicated to the header. The total inter-frame arrival time at TC layer is 125 μsec. Therefore the required payload duration (to serially upload the entire payload) operating on the same bit clock as the PM layer is,

= (payload length) x (frame duration) / (frame length)

= (19440 - 720)x(125 μsec)/19440 = 120.37 μsec

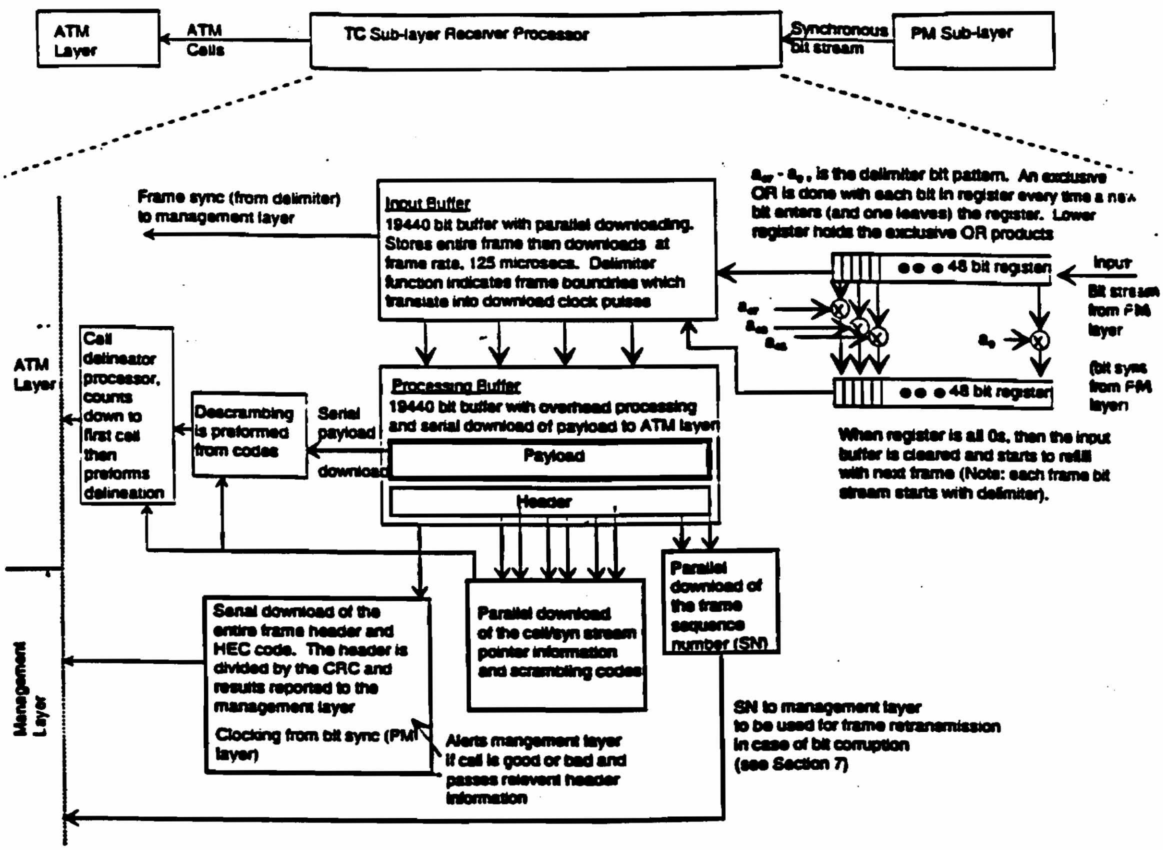

Figure 2-12 TC Sub-layer Receiver Block Diagram

The header descrambling field must then be processed in,

$$720 \text{ x } (125 \text{ µsec})/19440 = 4.62 \text{ µsec}$$

Hence, the payload will experience a delay of 4.29 µsec at the TC sub-layer before being sent to the descrambling processor.

2.3.2 ATM Layer and AAL

The AAL layer operates on top of the ATM layer. Figure 2-13 illustrates the protocol formats and header lengths for both layers [6].

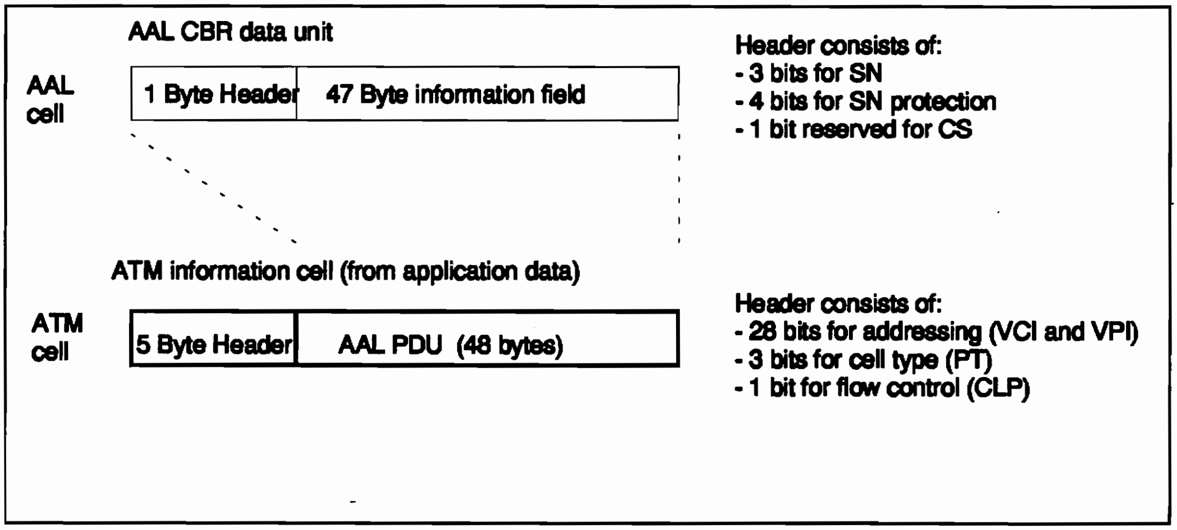

Figure 2-13 AAL and ATM Protocol Formats

The ATM layer receives/passes ATM cells from/to the AAL/TC sub-layer. The ATM layer appends the cell address fields to the incoming ATM-payload data units (PDUs) along with the payload type, and generic flow control. Figure 2-14 shows the ATM cell format [6].

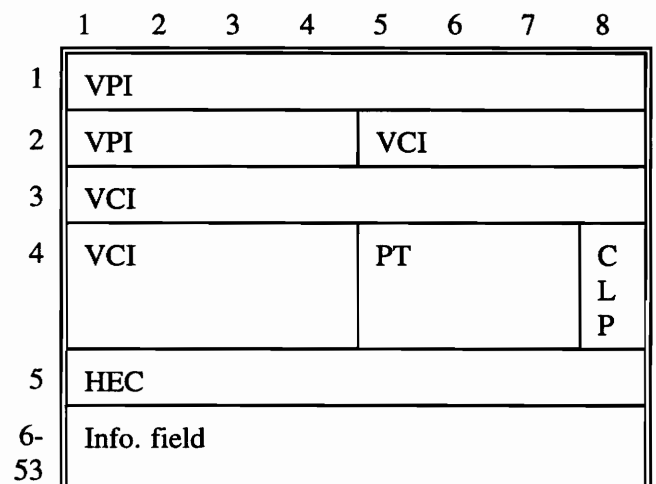

Figure 2-14 ATM Cell Format

CLP - cell loss priority, 1 = discard in case of congestion, 0 = do not discard in case of congestion. Set to 0 for the VTA application (application will not exceed its predetermined data rate, therefore cells from this application will not be discarded if congestion occurs).
HEC - cell delineation, header error check (single bit error correction) $x^8 + x^2 + x + 1$ [5].
PT - payload type, payload type coding is shown in Table 2.3 [1].

Table 2.3 Payload Coding

PT coding	Meaning
000	User data cell, congestion not experienced, SDU type=0
001	User data cell, congestion not experienced, SDU type=1
010	User data cell, congestion experienced, SDU type=0
011	User data cell, congestion experienced, SDU type=1
100	Segment OAM flow-related cell
101	End-to-end OAM flow-related cell
110	Resource management cell
111	Reserved

The ATM protocol also includes a HEC code over the ATM header fields. The header is 5 bytes and the ATM PDU is 48 bytes, hence each cell is 53 bytes in length.

ATM and AAL layers provide cell multiplexing and demultiplexing, SN generation and detection, addressing and routing using the VPI and VCI fields, header error detection, and cell flow control to ensure that connections stay within the limits negotiated at call establishment. The ATM and AAL layers support all sequence integrity for each transport virtual channel. It does not provide for error control. The method in which the VPI and VCI are used for addressing and routing is explained later in this section.

At the receiver side, ATM cells are passed from the TC sub-layer (delineator processor) if no errors have been detected. The TC sub-layer signals the ATM input buffer that a ATM cell has been delineated and the next 53 bytes constitute the cell. The ATM input buffer then collects the cell in serial (once cell delineation has been detected), and then up-loads the cell, in parallel, to the ATM processing buffer. The ATM processing buffer processes the header fields as illustrated in Figure 2-15 [2], and then sends the cell information field (in parallel) to the appropriate AAL processing buffer or management layer processing buffer. The AAL processor receives the cell information field and processes the SN and SNP fields. The AAL processor then uploads the AAL information field to the appropriate transport layer input port. The upload multiplexing is controlled by the VCI subprocessor.

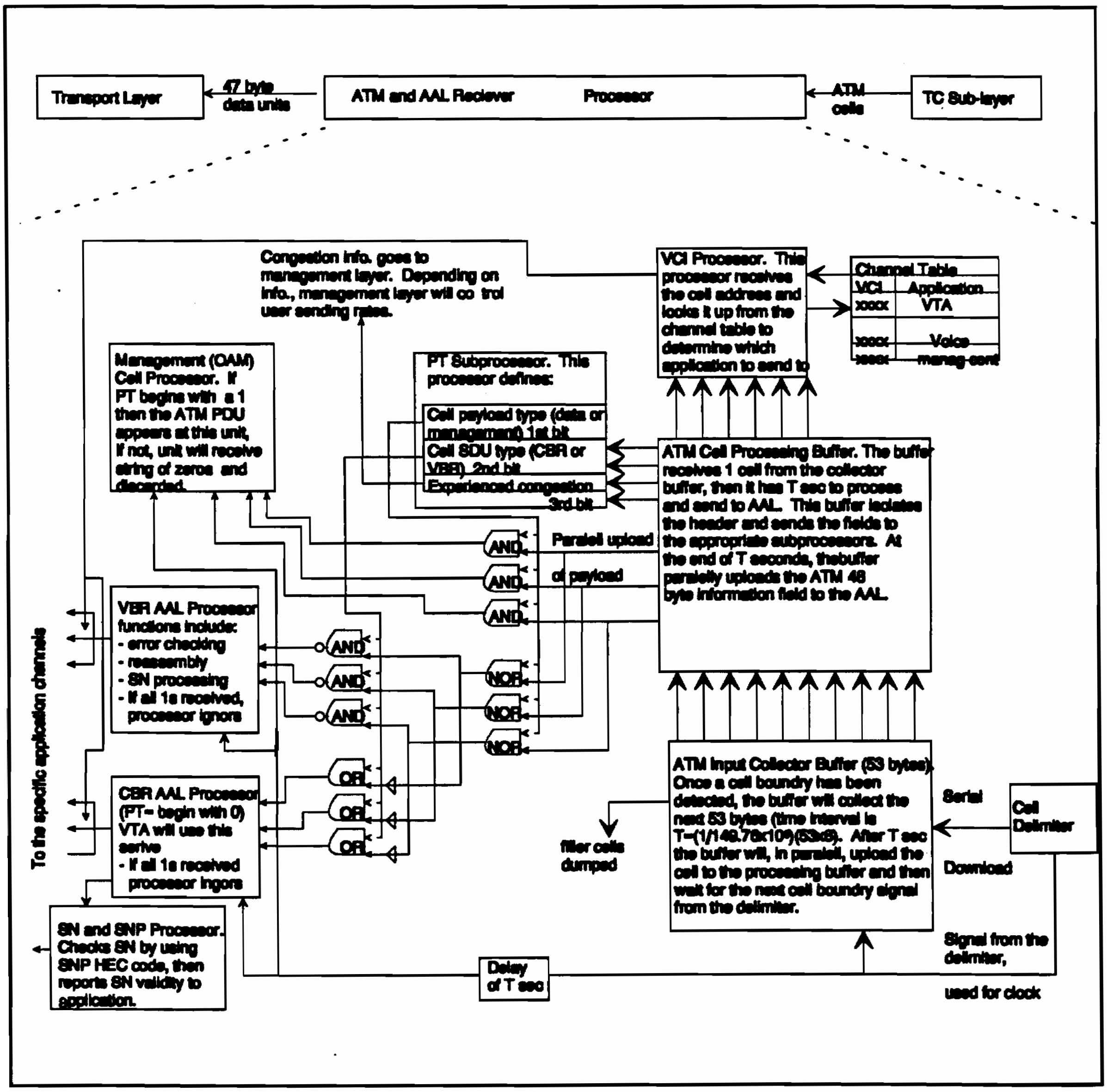

Figure 2-15 ATM and AAL Receiver Block Diagram

The transmitter AAL and ATM functional block diagram is shown in Figure 2-16 [2]. The AAL processor multiplexes the transport layer input channels on a 47 byte basis. Each of the 47 byte data units serves as the information field for the AAL cell. The AAL processor downloads data units every 2.84 μsec.

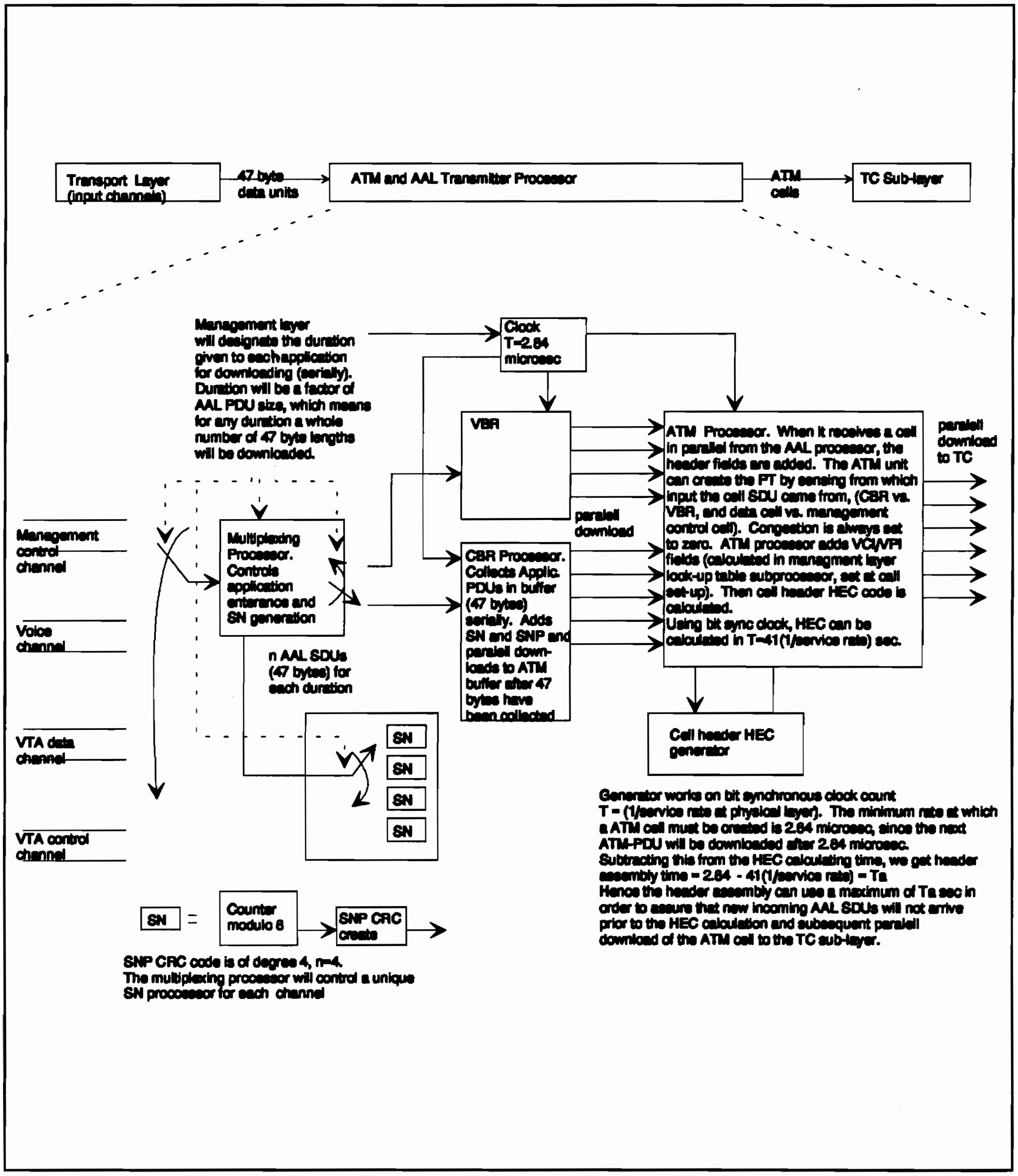

Figure 2-16 AAL and ATM Transmitter Block Diagram

$$\text{AAL Multiplexing Time Slot Duration} = \frac{\text{data unit length (bits)}}{\text{data rate available at the AAL layer}}$$

$$\text{Rate Available at the AAL} = (\text{service rate} - \text{SONET header rate}) \times \frac{\text{data unit length}}{\text{ATM cell length}}$$

$$\therefore \quad \text{Multiplexing Time Slot Duration} = \frac{47 \times 8}{138.38 \text{ Mbps}} = 2.84 \ \mu\text{sec}$$

During this 2.84 μsec duration, the SN and SNP header fields are calculated and inserted into the appropriate field locations. At the end of the 2.84 μsec duration, the AAL cell is downloaded in parallel to the ATM processor. The ATM processor calculates the ATM header fields and assembles the ATM cell. This is also done in 2.84 μsec. At the next clock count (2.84 μsec), the ATM cell is downloaded in parallel to the TC sub-layer. At connection set-up, applications will inform the management layer of the proposed sending rate and connection type (CBR vs. VBR). From this information, the management layer will inform the AAL multiplexing switch of the duration and frequency it should allocate to each application input channel and to which AAL service it should be routed to. The AAL layer offers two types of services. VBR- variable bit rate service and a CBR- constant bit rate service. The VTA will use the CBR service since it is basically a file transfer service at a fixed rate. AAL multiplexing durations should be discrete in which a multiple of 47 bytes of application data for each input channel can be downloaded. The management layer informs the multiplexing unit of how long to stay at each application input channel. The SN generator correspondingly switches between SN sub-generators dedicated to each application channel. The duration in terms of n (47 byte download durations) tells how many times the dedicated SN sub-generator will count during its operational duration. The durations must also account for sudden increase and decreases in data rate for some applications. This is explained later in this section. The management layer is responsible for controlling all channel multiplexing timing in the transmission module. It will keep a table of all current running applications as shown in Table 2.4.

Table 2.4 Management Layer Channel Table

Application	VPI/VCI (application informs Management Layer of destination at call set-up)	PT (application informs Management Layer of PT at call set-up)	Data Rate (application informs Management Layer of data rate at call set-up)	Multiplex Duration (calculated from proposed data rate)
VTA data	xxxxxxx	000	60.16 Mbps	56.8 μsec/frame
VTA control	xxxxxxx	000	64 Kbps	2.84 μsec/47 frames
Voice	xxxxxxx	001	64 Kbps	2.84 μsec/47 frames
Management control	xxxxxxx	100	64 Kbps	2.84 μsec/47 frames

Note - Multiplexing duration is calculated for the base data rate. For CBR this rate will remain constant. For VBR connections, the management layer must inform the switching unit to pole the input channels during the non-dedicated durations in case of rate increases. During rate decrease the channel sends nothing.

The base multiplexing durations are calculated with respect to the available rate into the AAL, which is, for a STS-3 service,

= [(interface service rate) - (SONET overhead)](47/53) = 132.83 Mbps (for STS-3)

This is the upper limit of available bandwidth to the application. Therefore the sum of all application data rates must be less than 132.83 Mbps. The AAL layer can only accept 47 bytes of data at a time, hence for a STS-3 service (155.52 Mbps), the maximum rate a which the AAL layer will process the 47 byte PDU unit is,

132.83 Mbps/(47 x 8) = .353 Mega AAL-PDUs/sec

Therefore, one AAL-PDU will be processed every 2.84 μsec (44 per 125μsec SONET frame). This is the minimum duration allocated to a application channel per SONET frame. Note that for very low rates, the management layer will allocate durations on a several frame basis.

The management layer then calculates the number of durations per SONET frame for each channel using the following equation.

(channel rate) x (frame duration)/(AAL-PDU length) = number of AAL-PDUs/frame

For example,

VTA channels:

= (60.16 Mbps) x (125 μsec)/(47 x 8) = 20 AAL-PDUs/frame

hence the duration per frame for the VTA will be 2.84 μsec x 20 = 56.8 μsec.

64 kbps channels:

= (64 kbps) x (125 μsec)/(47 x 8) = .021276596 AAL-PDUs/frame

= 1 AAL-PDU/47 frames

hence the duration every 47th frame will be 2.84 μsec.

In a similar manner, durations for all channel rates can be calculated on a per frame basis. From these values, the management layer can coordinate the application multiplexing into the AAL layer.

Any extra bandwidth at the AAL layer will be used to support rate increases for VBR channels. For the STS-3 service, there are 44 time slots per 125 μsec SONET frame. If only 40 are used, the management layer will use the 4 extra slots to pole the variable bit rate (VBR) input channel ports to determine if extra (increased data rate) bits have accumulated. Since the VTA will use a constant bit rate (CBR) service, this function will not be utilized by the VTA system.

2.3.3 Call Routing and Addressing Using VPI and VCI Fields

Two levels of identifiers are used to route cells in ATM networks, VPI and VCI. Assume that a virtual channel consisting of a 6-hop path is to be established between two users, A and B. After establishing the path, the network assigns VCI values to be used at each link along the path and sets up routing tables at nodes N1 to N7, as shown Figure 2-17 [1].

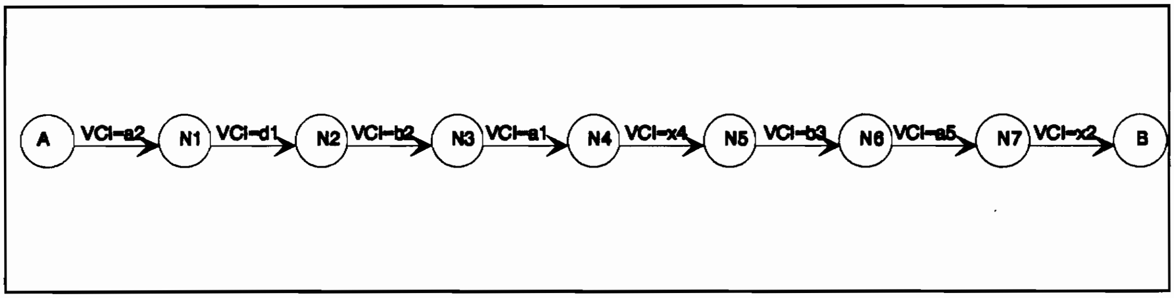

Figure 2-17 Routing Using VCIs

All cells are routed via the VCI routing tables. At the end users (ATM interface), the VCI will be user to route the ATM to/from the appropriate application. Hence the application address is converted into a "ATM address" via the VCI.

The concept of virtual paths is introduced in ATM networks to reduce the size of the routing tables and processing complexity of the network. A virtual path is a logical direct link between two nodes in the network that are connected via two or more sequential links (note, for a single hop, virtual path and virtual channel concepts are identical). Connections using the link between the two end nodes of the virtual path are bundled together and transported with an identifier common to all virtual channels, referred to as a VPI. The cells belonging to different virtual channels are switched using this common VPI along the nodes of the virtual path, which will reduce the routing table size. For example, if the path from A to B consists of two virtual paths, as shown in Figure 2-18 [1].

Then the network assigns a VPI value of a2 from N1 to N3 and x4 from N4 to N7. Note that the VPI values are switched at each node, while the VCI values remain the constant in the virtual path. The common virtual paths are switched while the virtual channels are not.

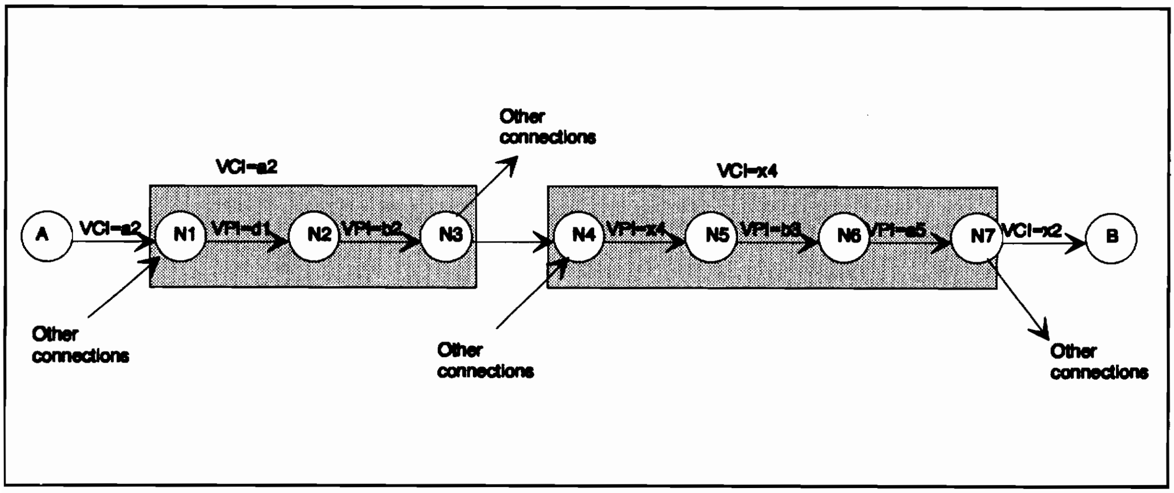

Figure 2-18 Routing Using VPIs

3.0 VTA Application Data Transfer Requirements

The VTA system is proposed to basically distribute stored video data information to paying customers. Each video, in its compressed state, will be stored at provider locations. There will be a set of video provider facilities (each storing a subset of the entire video library) servicing a dedicated geographical area of users. This is done to ease the buffer requirements and simultaneous transfer requirement at the provider. When a video is requested by the user, the request will be routed to a common video provider data base, where the request will be logged. The common provider data base will determine which video provider facility contains the requested video, and inform the user of the address. The user then initiates a cell set-up scenario with the appropriate video provider facility. Once the connection is established, the provider will download the video in a constant bit rate fashion (as is common with file transfer applications). Upon reception at the user, the video will be buffered for later viewing. Once the viewing command at the user application is initiated, the data will be sent from the buffer to the viewing equipment (TV or monitor) at the real time data rate. The application will be responsible for converting the digital data into signals comprehensible by the viewing equipment, and preforming the decompression. This signal conversion will depend on the type of equipment (analog for most current TVs and digital for future TVs). The viewing equipment decompresses the data as it is being viewed. Hence the VTA is a high speed form of a file transfer service, where the files, in this case, contain compressed video data.

The main VTA transmission requirements effecting the transport protocol design, are the overall application transmission data rate and the error sensitivity of the compression method. To define these requirements, the following must be considered:

- A standard data compression technique (this will yield the maximum acceptable cell error or cell loss ratio);

- The file transfer data rate (this will determine the total transmission duration of the video file).

Besides these VTA dependent requirements, the transport protocol can treat the application as if it were a common file transfer service.

3.1 Data Compression Technique

Video is presented to users as a series of frames in which motion of the scene is reflected in small changes in sequentially displayed frames. Frames are displayed at the user equipment (TV) at some constant data rate, 30 frames/sec for VCR quality service. Current digital representation of video signals requires the transformation of a continuous image field into the discrete domain. This transformation is a mapping from the continuous image samples into a finite set of discrete amplitudes that span the intensity range of the image, similar to the PCM method for voice. Each quantized amplitude is defined uniquely by a digital code word referred to as a pixel. For full color, 24 bits/pixel (8 each for red, green, and blue) are used. Currently, VCR recorders deliver an image resolution of 352x240 pixels per frame [1]. Therefore, the raw data rate (bandwidth) for VCR quality service is,

30 frames/sec x (352x240) pixels/frame x 24 bits/pixel = 60.8256 Mbps

This report assumes VCR quality service for the VTA. High density TV (HDTV) would probably be costly in terms of resource requirements to be economically feasible.

This is a very high data rate and will require a lengthy transmission duration. Suppose that the video is 2 hours long (7200sec), this yields 4.3794432×10^{11} bits for each video transfer. The transmission duration can be reduced by compressing the video data; sending the compressed form through the transmission system; and decompressing it, once received at the user end. Video sequences contain a significant amount of repetition, both within a frame and between frames. It is very likely that the pixel values corresponding to neighbor locations within a frame are close to each other, this is referred to as spatial redundancy. Similarly, subsequent frames have large amounts of data common between them (picture does not change much from frame to frame), this is referred to as temporal redundancy.

- Spatial redundancies lead to intra-frame coding (within frame)
- Temporal redundancies lead to inter-frame coding (between frames)

Video compression standards have been presented for use with VOD applications. Current VOD systems have proposed using a video technology standard from the Motion Picture Experts Group (MPEG) [1]. This compression standard will serve as the basic compression technique for the VTA (provide high compression ratios while maintaining VCR quality, based on 352x240 pixel/frame). The MPEG video compression algorithm relies on 2 basic techniques: block base motion compression (MC) for reduction in temporal redundancy and discrete cosine transform (DCT) for spatial redundancy. The video data is hierarchically organized into 8x8 pixel blocks for spatial reduction and 16x16 pixel blocks for macro-blocks for motion compression. A group of macro-blocks (22) forms a slice (352x16 pixels), and 15 slices forms the frame (one every 1/30 seconds).

Discrete cosine transform - image is divided into a number of NxN disjointed pixel blocks (8x8 for MPEG). These are orthogonally transformed to produce a set of transformed coefficients $\{F(u,v): 0 \leq u, v < N\}$ where $F(0,0)$ corresponds to the mean of the pixel values on a block. The $F(u,v)$ coefficients are the off-set values for each pixel with respect to the mean value. The coefficients are then thresholded and quantized. Finally the quantized coefficients are scanned into a one dimensional sequence whose nonzero amplitudes and run lengths of zeros are variable length coded before being passed on.

Motion Compensation [1] - prediction error, defined as the difference between the current block (16x16 for MPEG) and the motion compensated block from the previous frame, is coded using DCT intra-frame coding method. The MPEG produces three types of output blocks, referred to as frames: intra-frames (I), predicted frames (P), and interpolated frames (B). I frames are points for random access. They refresh the frame sequence, with only DCT compression, preventing errors from propagating across frames. The first frame in a sequence is a I frame. P frames are coded with reference to a pervious frame and are generally used as a reference for subsequent P frames. These, predictively motion-compensated frames include motion vectors that represent the difference between the spatial location of the macro-block and that of its predictor. A draw back is that new information (with respect to the DCT transform) cannot be adequately compensated since the previous frame does not include related information. However, this information will most likely available in future frames. The MPEG scheme utilizes this by allowing backward in time prediction and interpolation by using previous as well as future frames. These frames are referred to as B frames. Figure 3-1

[1] shows the block design for the compression algorithm. Figure 3-1 shows 4x4 DCT pixel blocks for simplicity.

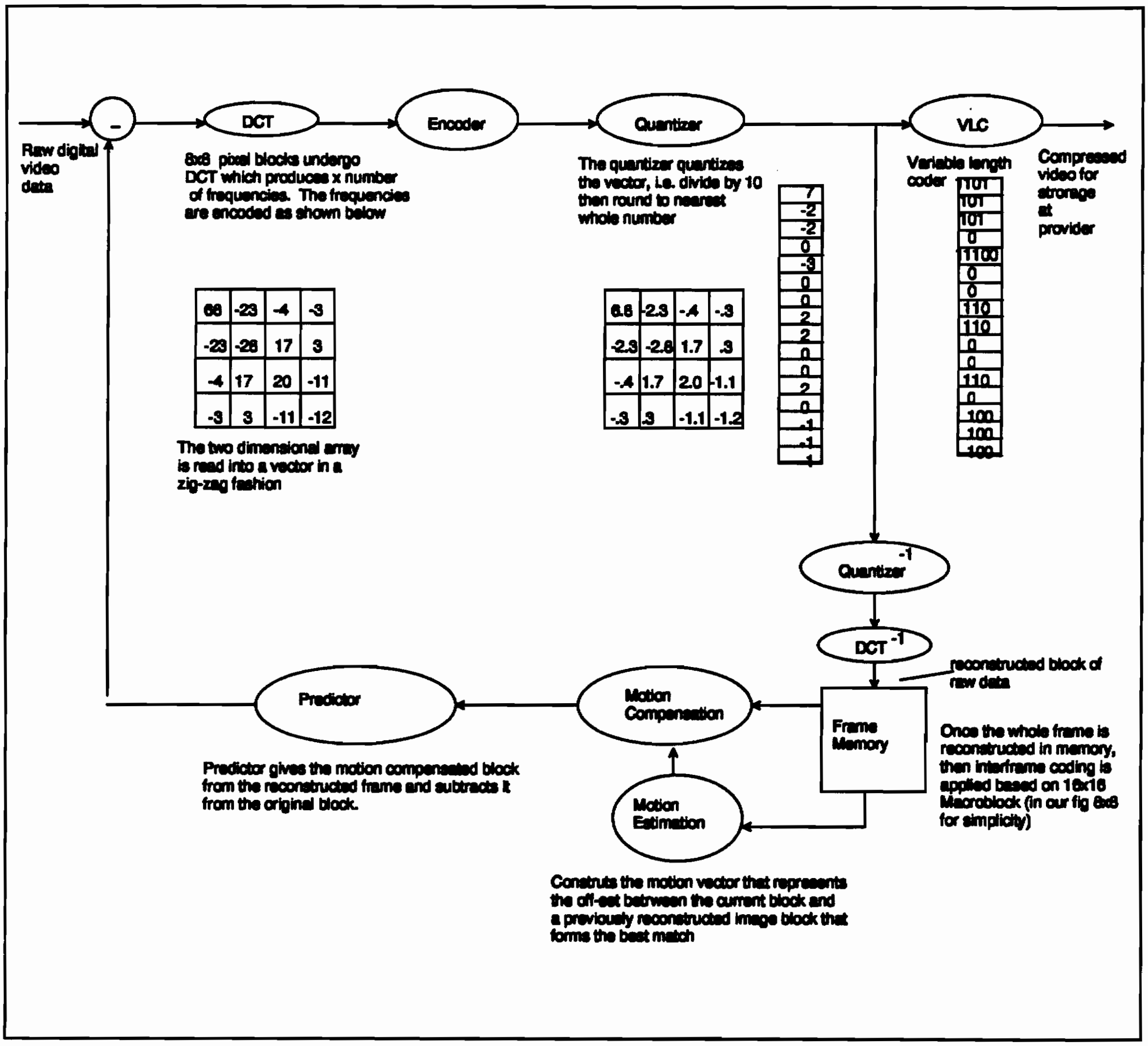

Figure 3-1 MPEG Compression Block Diagram

The I frames are compressed by taking the DCT of blocks of image data, quantizing the DCT c-coefficients, and variable length coding (VLC) of the quantizer coefficients. P frames allow higher compression by relating the DCT coefficients to the previous block, where as B frames allow the highest compression rates but both requires previous and subsequent frame reference for prediction.

At the receiver, the bit stream is first decoded through an inverse variable length coding device. The output of the inverse VLC then passes through an inverse quantizer and then through an inverse DCT that yields the DCT coefficients. From these coefficients a block of data is reconstructed and stored in the frame memory. In the inter-frame mode, motion vectors are extracted from the VLC decoder, which are used to provide the location of the predicted blocks. Figure 3-2 [1] presents a simple block diagram of the decompression algorithm.

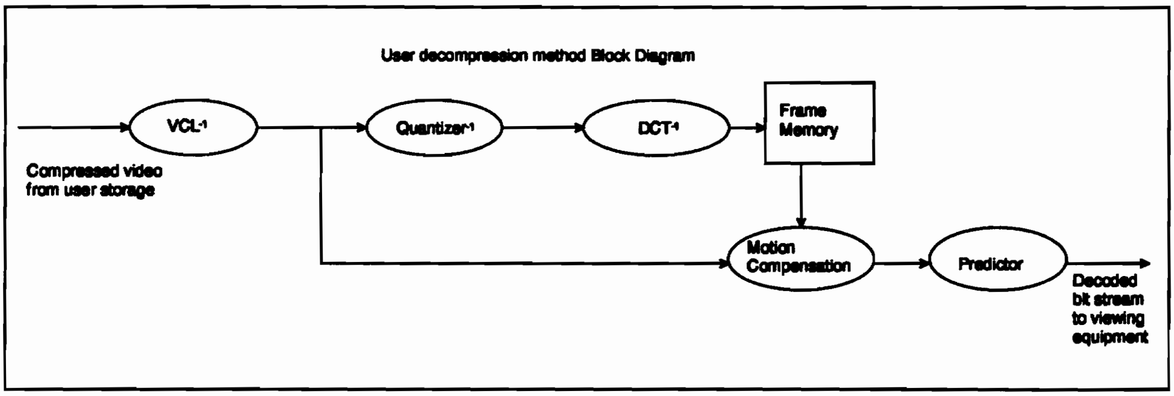

Figure 3-2 MPEG Decompression Block Diagram

The MPEG standard can reduce raw bandwidth rates by twenty to fifty fold depending on the application and desired picture quality. For VCR quality service, operating on a 352x240 pixel/frame, 24 bits/pixel, 30 frame/sec full color, full motion raw data rate, a compressed data rate of around 1.5 Mbps can be expected [1]. Hence, the VTA will need to send,

(1.5 Mbps) x (video length (sec)) bits for each request

If an average video length of 2 hours is assumed, this yields an average data transfer of 1.08×10^{10} bits per request. To reduce the loss of quality, once the video has been compressed, the provider can decompress the video "off-line" and view it. If any obvious degradations of quality exist, the provide can reduce the amount of compression applied to that segment of video data. Once the video is of acceptable quality, the video can be stored in the provider video storage memory for user access (requests).

With video data, a video frame (picture) contains a large amount of data and is transmitted over a number of ATM cells. Loss of a cell in the middle of a frame not only effects the particular block of data, but also due to compression techniques, will effect the block of data spatially adjacent to it and in neighboring frame blocks. Hence, depending on the type of block that was lost, either I, P, or B, this may result in, respectively, a short flash picture distortion, a distortion of a number of lines, or a soft cloud distortion that spreads over several frames and diminishes gradually.

The MPEG standard defines a maximum cell loss ratio (CLR) of 10^{-8} [1]. Hence the end-to-end connection must be capable of supporting this value in order to achieve the desired compression ratio while maintaining VCR picture quality. Appendix A presents the link budget equations and analysis for a n-fiber end-to-end connection to meet this requirement. From the analysis, maximum connection length (given set fiber parameters) can be determined.

3.2 File Transfer Data Rate Requirement

Each user is assumed to be connected to the network via a full duplex 155.52 Mbps link. This yields an application rate of 132.8 Mbps. The determination of optimal VTA data transfer rate was determined based on the following criteria:

- The VTA rate must be significantly less than the full application rate in order to allow other application sufficient bandwidth for simultaneous operation.

- The VTA rate was designed to simplify the management layer processing with respect to assigning AAL multiplexing durations (see Table 2.4). The multiplexing duration for the VTA application should be an integer multiple of the minimum duration (one 2.84 μsec slot). Therefore, the data rate must obey the following relationship,

  data rate = n x (47x8)/(125 μsec)   such that n = integer less than 44

- The VTA rate can not drive the subchannel breakout parameter such that it exceeds the total number of connections that the management layer can support (~50 [1]). The subchannel breakout parameter is a result of the transport protocol analysis in Section 4.0. From the analysis, the following relationship is established,

  data rate ≤ (max # of channel management layer can support) x (min subchannel breakout data rate)
  data rate ≤ 50 x (1.33 Mbps) = 66.5 Mbps

- The VTA rate should be as high as possible, given the above constraints, in order to minimize transfer time duration.

From the above criteria, a VTA transfer data rate of 60.16 Mbps was chosen. This data rate yields,

- 72.64 Mbps for other application;
- AAL multiplexing duration of 20, 2.84 μsec slots, during each 125 μsec SONET frame;
- Maximum of 45 subchannels (described in Section 4.0)
- Reasonable transmission durations.

Considering a 2 hr video, a compression rate of 1.5 Mbps, and a transfer rate of 60.16 Mbps; the total average transfer duration (excluding data unit retransmissions due to errors) will be,

$$\frac{(7200) \times (1.5\ \text{Mbps})}{60.16\ \text{Mbps}} \approx 3\ \text{min}$$

3.3 VTA Buffer Requirements

The VTA will need sufficient buffer space to store the video data. Assuming a maximum video length of 3 hr, the VTA will need to store 16.2 Gb of compressed information. New storage techniques are currently in the development phase, such as CD and magneto-optical (MO) disk, and could be used for the VTA. If this is economically unfeasible in the future, the video could be downloaded in parts, however, this would require additional connection processing (disconnect connection, then re-set-up connection when buffer clears). This issue is not examined in detail in this report.

4.0 VTA Transport Protocol Analysis

4.1 Transport Protocol Requirements

From the ATM layer analysis, the network functions were identified. The primary functions include:

- The ATM protocol can support any channel rate less than or equal to the available rate at the AAL layer (132.8 Mbps);
- Modulo 7 SN generator;
- Error detection on SONET, ATM, and AAL headers (not on application data for CBR service); damaged cells will not appear in transport receiver buffer;
- Management layer coordinates application multiplexing timing and appends addressing information in the form of VPIs/VCIs, payload types (also for cell identification), and generic flow control to each application cell accordingly.

From the VTA requirements analysis, the transport layer must be able to support,

- 60.16 Mbps channel rate;
- CLR = 10^{-8};
- Maximum channel utilization to achieve minimum transfer durations.

The following sections, will examine the transport functions and design each function according to the VTA requirements, while utilizing the ATM protocol stack functions.

4.1.1 Error Control

The ATM layer provides CRC coding for SONET header protection, ATM cell header protection, and AAL header protection (CBR AAL service only provides header protection). The CRCs provide nearly 100% error detection for associated headers. The transport layer-to-transport layer CLR must be analyze to determine if additional error protection is needed over the application data to meet the VTA data transfer requirements (for MPEG compression technique and VCR quality, a CLR = 10^{-8} is recommended). To determine if additional error protection is needed at the transport layer, the following must be examined:

- Probability of a transport data unit containing bit errors;
- Acceptable CLR for the video service (this is the acceptable probability of error for the service after error detection);

- Additional cost (extra transport layer header bits) of implementing error detection coding over the application infomation.

The probability that a application data unit (47 bytes) contains errors is dependent on the transmission link BER values. The link BER is defined as the probability that a single bit is received erroneously over a single link. Hence the end-to-end BER (including receiver noise - receiver noise is factored in at each link - see link analysis, Appendix A) is derived from Figure 4-1.

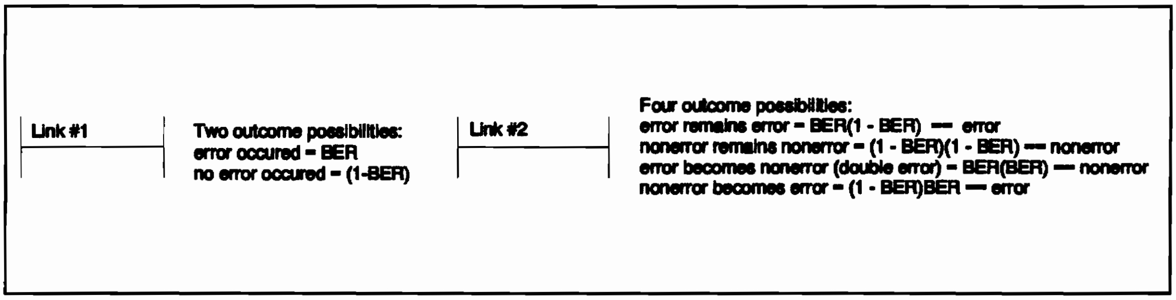

Figure 4-1 BER Analysis

The error analysis continues as shown in Figure 4-1 for n links. Note, above analysis assumes binary transmission (only two states corresponding to 0 and 1) and that all links have the same BER. The probability of an error reversing itself is a maximum when n=2 and is BER x BER, this value can be considered zero, hence the probability that a bit, received at the user end, is not in error is $(1 - BER)^n$, where n is the number of links.

47 bytes or 376 bits make up one transport layer data unit. The probability that this data unit is received error free is,

$$((1 - BER)^n)^{376}$$

Therefore, the probability (or CLR) that a application data unit is received <u>with</u> errors is,

$$CLR = 1 - ((1 - BER)^n)^{376}$$

The recommendation for the MPEG compression technique is $CLR = 10^{-8}$

Note, if the links have different values of BER per link, the CLR equation can be written as

$$CLR = 1 - [\prod (1 - BER_i)]^{376} \text{ multiplied over } i=1 \text{ to } n$$

Table 4.1 has the calculated end-to-end BER values for various connection parameters.

Table 4.1 End-to-End BER Probability for Various Link Parameters

	1 link	2 links	3 links	4 links	5 links
link BER= $1x10^{-9}$	$3.8x10^{-7}$	$7.5x10^{-7}$	$1.1x10^{-6}$	$1.5x10^{-6}$	$1.9x10^{-6}$
link BER= $1x10^{-10}$	$3.8x10^{-8}$	$7.5x10^{-8}$	$1.1x10^{-7}$	$1.5x10^{-7}$	$1.9x10^{-7}$
link BER= $1x10^{-11}$	$3.8x10^{-9}$	$7.5x10^{-9}$	$1.1x10^{-8}$	$1.5x10^{-8}$	$1.9x10^{-8}$
link BER= $1x10^{-12}$	$3.8x10^{-10}$	$7.5x10^{-10}$	$1.1x10^{-9}$	$1.5x10^{-9}$	$1.9x10^{-9}$

Note: in the table above it is assumed that all links in a connection have the same BER.

The bit error rate into and out of the switches can be ignored. Test results from reference [13] show that mutilpexers and demultiplexer with frame detection, can operate at 2.488 Gbps with a bit error rate of $1x10^{-14}$. This tem can be ignored when compared to the expected BER of the fiber links.

As long as CLR<= 10^{-8} for the transport layer-to-transport layer connection, no extra error detection, other than what is provided by the ATM protocol stack, is needed. A tremendous advantage is observed in processing simplicity and data throughput, if further error detection at the transport layers can be ignored. We assume that this end-to-end CLR can be economically achieved through future fiber technology, Appendix A develops the link parameters per connections to meet this requirement. The results of the analysis indicate that state-of-the-art fiber technology can support the CLR requirement.

The analysis above assumes that all errors in the header fields are detected and either corrected or retransmitted.

Errors detected at the ATM layers results in the discarding of ATM cells. Cells are counted based on the Modulo 8 SN counter function in the AAL. In order to utilize this function, the transport layer data unit acknowledgement scheme must operate on a maximum window size of 7. Therefore, no more than 7 data units can be sent by the transmitting transport layer before it receives ACKs for the data units. This leads to a major problem with using high data rate applications over ATM. Recall the ATM protocol stack sends data units (which become cells in the ATM layers) in SONET frames. Hence, the sender is restricted to sending, at most, 7 cells per frame, (then wait for a ACK or NAK before sending more cells). For the VTA application, we are proposing to support a user rate of 60.16 Mbps or 20 cells per SONET frame. If this rate is sent over ATM, the SN function will become meaningless due to the fact that 20 cells per frame will arrive without a ACK or NAK in between. Therefore the number of cells greatly exceeds the limit placed upon it by the SN function at the AAL. Increasing the SN count will require the entire ATM protocol structure to be redefined, thus this is not a option. An alternate option is to design the transport protocol to decompose the VTA application data rate (60.16 Mbps) into small sub-channels which will be able to use the modulo 8 SN. In order to implement a subchannel breakout scheme which will maximize the 60.16 Mbps channel rate while maintaining the SN constraint, the following channel parameters must be defined,

- Retransmission rate and end-to-end delay;
- Proposed subchannel data rate;
- Processing time to detect error at the ATM and AAL layers;
- Extra processing and signalling needed to support the sub-channel breakout;

There are three types of error control techniques currently used in transport protocols [5],

- Stop-and-wait ARQ
- Go-back-N ARQ (maximum window size $2^n - 1$)
- Selective-reject ARQ (maximum window size $2^n - 1$)

Any of these techniques could be used, however, the choice of which will drive throughput efficiencies (channel utilization) and processing and buffer requirements at the transport layer.

The average transport layer-to-transport layer round trip delay (data from provider to user and ACK/NAK from user to provider) on a cell basis must be calculated to determine error control throughput. This relationship is derived from the ATM protocol stack operations and the physical layer operations described in Section 2. Note processing and propagation delay can not be ignored due to the low transmission delays inherent in ATM networks.

<u>Round Trip Delay</u> = data cell processing time at transmitter (provider end) + data cell transmission time + propagation and queuing delay through the network + data cell processing time at the receiver (user end) + ACK/NAK processing time at the transmitter (user end) + ACK/NAK transmission time + propagation and queuing delay through the network + ACK/NAK cell processing time at the receiver (provider)

Each term is derived in turn,

Data cell processing time at the transmitter (provider end) - The processing delay from the application to the TC sub-layer is derived from Figure 2-16. When the management layer signals the VTA application to transmit (during its allocated transmission duration), the bits are passed in serial to the CBR processor where they are buffered until the buffer is completely full. The duration for one 47 byte data unit is 2.84 μsec. The AAL header is also appended to the data unit during the 2.84 μsec duration. The AAL-PDU is then downloaded in parallel to the ATM processor where the ATM header is constructed and appended to the AAL-PDU. This duration is also 2.84 μsec. At the end of the 2.84 μsec duration, the TC-PDU is downloaded to the TC sub-layer in parallel. Hence the processing delay for a VTA data unit (or any transport protocol data unit for that matter) from the transport layer to the TC sub-layer is,

$$2 \times (2.84\ \mu\text{sec}) = 5.68\ \mu\text{sec}$$

The processing delay from the TC sub-layer to the fiber is derived from Figure 2-11. At the TC sub-layer, the TC-PDU is buffered in the scrambling buffer and checked for any words corresponding to the SONET frame delimiter. The TC-PDU is stored in this buffer for 2.84 μsec, then downloaded in parallel to the SONET frame assembly processor. The SONET frame assembly processor maps the ATM cell into the SONET frame. The processing delay at this processor will depend on whether the ATM cell is the first, second...., or 44th cell mapped into the frame. We will used the average waiting time for a cell for this analysis (mean waiting time), hence the delay time in the assembly frame processor is 125μsec/2. The entire frame is then downloaded in parallel to the output buffer where it is immediately transmitted to the PM layer (in serial). In the PM sub-layer, there is no processing delay. The processing delay from the TC sub-layer to the fiber is,

$$2.84\ \mu sec + 62.5\ \mu sec = 65.34\ \mu sec$$

The total processing delay from the transport layer to the fiber is,

$$5.86\ \mu sec + 65.34\ \mu sec = 71.2\ \mu sec$$

Data cell transmission time is calculated with respect to the VTA subchannel rate. The fiber data rate is viewed in terms of a virtual channel dedicated to the particular VTA subchannel rate, since each VTA subchannel will only be able to transmit up to the virtual channel rate (note, the optimum virtual channel rate is to be derived from this analysis). The transmission time (T) is then,

T = AAL-PDU (length in bits)/(VTA subchannel rate (bps)) = 376/(VTA subchannel rate)

Propagation and queuing delay can be calculated on a per fiber basis, where,

Propagation delay = (distance of fiber in meters)/(velocity of propagation in meters/sec)

Since the end-to-end connection will be made up of several fiber links, the propagation delay is,

T= Σ(distance of fiber in meters)/(velocity of propagation in meters/sec) summed over the number of links

The queuing delay experienced at each switching node along the connection can only be determined by extensive analysis of the switching fabric and the traffic input/output load characteristics. For properly designed switches, queuing delays (in non-congested state) are on average between 1 and 4 times the cell duration [11]. We will calculate the delay based on the average operating condition. For properly designed switches a average value of 2 is common. The queuing delay is then,

queuing delay = 2(cell length in bits)/(service rate)

for a 155.55 Mbps fiber, the delay is 5.45 μsec.

Therefore, the total propagation and queuing delay is,

$$T = \Sigma \{ \text{(distance of link)}_i / \text{(velocity of propagation)}_i \} \text{ summed over number of links}$$
$$+ \Sigma \{ 5.45 \ \mu\text{sec} \} \text{ summed over number of links minus one}$$

The data cell processing time at receiver (user end) from the fiber to the ATM layer is calculated from Figure 2-12. As the SONET frame arrives (containing the transport data unit under consideration) it is demodulated and passed to the TC input buffer where it is collected and uploaded in parallel to the TC processing buffer, once the last bit of the SONET frame is received. This delay is accounted for in the transmission delay parameter. At the TC processing buffer, the SONET header is extracted while the frame payload is passed in serial to the descrambler and cell delineator. The overall delay experienced at this processor also depends on the data unit location within the SONET frame, the average delay is 62.5 μsec. Figure 2-15 is referenced to calculate the ATM to transport processing delay. Each ATM cell (cell has been identified in the cell delineator processing unit) is buffered at the ATM input buffer which collects the serial stream of bits comprising the cell. Once the last bit has been received, the entire cell is uploaded in parallel to the ATM processing buffer. The processing buffer extracts the header fields and processes the information (e.g. application routing and management information). The processing buffer contributes a delay of 2.84 μsec for each ATM cell (this is the minimum duration between ATM cell delineations, since this is the sending duration, the processor must operate at this minimum rate to ensure buffer overflow will not occur). The AAL data unit is then uploaded in parallel to the CBR AAL processor where the AAL header is processed. This processor also contributes a 2.84 μsec delay. The AAL data content is then uploaded to the appropriate transport layer/application input port. The total processing delay is,

$$62.5 \ \mu\text{sec} + 2.84 \ \mu\text{sec} + 2.84 \ \mu\text{sec} = 68.18 \ \mu\text{sec}$$

The total delay from the provider transport layer to the user transport layer is,

$$T_{data} = 71.2 \ \mu\text{sec} + 376/(\text{VTA subchannel rate}) + \Sigma \{ \text{(distance of link)}_i / \text{(velocity of propagation)}_i + 5.45 \ \mu\text{sec} \} + 68.18 \ \mu\text{sec}$$

The delay of the ACK/NAK cell from the user to the provider transport layer is the same expression, since the ACK/NAK signaling will take place on the full duplex subchannel. Hence,

$T_{ACK/NAK}$ = 71.2 μsec + 376/(VTA subchannel rate or out-of-channel signaling rate) + Σ { (distance of link)$_i$ /(velocity of propagation)$_i$ + 5.45 μsec } + 68.18 μsec

From the above analysis of transport layer to transport layer delay, relationships can be derived to compare the 3 types of error control techniques based on VTA subchannel rate and fiber characteristics.

The error control performance is based on utilization of the connection data rate (VTA subchannel rate). The most efficient schemes will be when the utilization approaches 1. For the three error control techniques, the utilization equations are presented below.

Stop and wait:	U= (1 - P)/(1 + 2a)		
Selective-reject:	U = 1 - P	N > 2a + 1	
	U = N(1 - P)/(2a + 1)	N < 2a + 1	where N=7
Go-back-N	U = (1 - P)/(1 + 2aP)	N > 2a + 1	
	U = N(1 - P)/{ (2a + 1)(1 - P + NP) }	N < 2a + 1	where N=7

where P = probability that the ATM cell carrying the a transport data unit is in error, which is,

$P = CLR = 1 - [\prod (1 - BER_i)]^{48}$ multiplied over i=1 to the number of fiber links comprising the connection

Note: P is the probability of error detection (error detection is only applied over ATM and AAL headers (6 bytes = 48 bits)) where as the CLR is the probability of an error not being detected (error in the transport layer data unit).

and where a = the ratio of,

(transport-to-transport propagation + queuing + processing delay in one direction) / (data cell transmission rate)

a = {71.2 μsec + Σ [(distance of fiber link)$_i$ /(velocity of propagation)$_i$ + 5.45 μsec] + 68.18 μsec} / {(376)/(VTA subchannel rate)}

From the equations above, the three error control techniques can be compared for a given connection. Table 4.2 presents the values of P, calculated for various BER per fiber values for a connection consisting of 1 to 5 links.

Table 4.2 Probability of Nondetected Error for Various Link Parameters

	1 link	2 links	3 links	4 links	5 links
link BER= $1x10^{-9}$	$4.8x10^{-8}$	$9.6x10^{-8}$	$1.4x10^{-7}$	$1.9x10^{-7}$	$2.4x10^{-7}$
link BER= $1x10^{-10}$	$4.8x10^{-9}$	$9.6x10^{-9}$	$1.4x10^{-8}$	$1.9x10^{-8}$	$2.4x10^{-8}$
link BER= $1x10^{-11}$	$4.8x10^{-10}$	$9.6x10^{-10}$	$1.4x10^{-9}$	$1.9x10^{-9}$	$2.4x10^{-9}$
link BER= $1x10^{-12}$	$4.8x10^{-11}$	$9.6x10^{-11}$	$1.4x10^{-10}$	$1.9x10^{-10}$	$2.4x10^{-10}$

Note: in the table above it is assumed that all links in a connection have the same BER.

For the comparison analysis, an average probability of 10^{-9} is assumed. Furthermore, it is assumed that the connection between provider and user is comprised of 3 links, each 15 km in length and each with a propagation velocity of $2.2x10^8$ m/s. These values are good approximations with respect to average fiber propagation (refractive index ~ 4.87) parameters and network topology characteristics. Using these link parameters as a basis of comparison, the term "a" can be solved for in terms of VTA subchannel rate,

$$a= 2 \times \{71.2\ \mu sec + \Sigma\ [\ (15\ km)_i\ /\ (2.2 \times 10^8)\ m/s_i+5.45\ \mu sec] + 68.18\ \mu sec\}/\ (376)\}$$
$$\times\ (\text{VTA subchannel rate}) \qquad \text{where the sum is from } i=1 \text{ to } 3$$

Note: the end-to-end delay is multiplied by 2 to calculate the round trip delay.

$$a = 2 \times (9.43684725x10^{-7}) \times (\text{VTA subchannel rate})$$

Furthermore, it was assumed that $P= 10^{-9}$

Table 4.3 presents the results of the comparison of error control techniques for the baseline parameter assumptions.

Table 4.3 Error Control Comparison Based on Utilization

	subchannel rate = 1 Mbps	subchannel rate = 1.5 Mbps	subchannel rate = 1.7 Mbps	subchannel rate = 2 Mbps	sub channel rate = 3 Mbps
Stop & Wait	.21	.15	.13	.117	.0811
Go Back N	1	1	.95	.82	.57
Selective Repeat	1	1	.95	.82	.57

From the table, it is evident that for the VTA application, the most efficient error control scheme based on utilization is either Go Back N or Selective Repeat with a subchannel rate ~ 1.7 Mbps. The Go Back N was chosen based on a comparison of implementation processing and complexity. If the overall VTA rate of 60.16 Mbps is divided into 35 equal subchannels at the transport layer, each subchannel will have a data rate of 1.72 Mbps yielding a subchannel rate utilization of ~ 1. Each user-provider connection will not have the same connection parameters, however. To fully utilize the overall VTA rate of 60.16 Mbps, each connection must dynamically determine the most efficient subchannel breakout for the particular user-provider connection. Examining the equation for the "a" term, it is evident that the propagation and queuing delay is the only variable based on different user-provider connections. The delay variation will depend on the switching parameters and expected traffic load during the connection and fiber propagation velocity and length. The optimal subchannel rate calculation per user-provider connection should be calculated individually for each connection to insure maximum subchannel utilization over the specific connection. Hence the transport protocol must be flexible enough to allow for variations in the subchannel breakout parameter for user specific connection parameters at call set-up. A range of subchannel breakout parameters can be derived by examining the connection parameter extremes. First it is noted that a connection will always go through at least one switching node (users and providers will never be directly connected). Reasonable limits are provided below, set a minimum propagation and queuing delay at $\rightarrow$ 2 links (5 km each), hence,

$a = 2 \times (139.38\ \mu sec + 2(5\ km / 2.2x10^8\ m/s) + 5.45\ \mu sec) / 376 = 1x10^{-6} \times$ (VTA subchannel rate)

set a maximum propagation and queuing delay at $\rightarrow$ 4 links (15 km each), hence,

$a = 2 \times (139.38\ \mu sec + 4(15\ km / 2.2x10^8\ m/s) + 5.45\ \mu sec) / 376 = 2.28x10^{-6} \times$ (VTA subchannel rate)

The Go Back N equation is then solved for the maximum rate that yields a utilization of ~ 1. Since the probability of error detection per cell is very small, 10^{-9}, the utilization relationships can be rewritten as,

$U = 1$	$N > 2a + 1$	where N=7
$U = N/(2a + 1)$	$N < 2a + 1$	where N=7

Hence U ~ 1 when $N = 2a + 1$ or when a=3. Solving for "a" for the limits results in,

For minimum limit: $(1x10^{-6}) \times$ (VTA subchannel rate) = 3
VTA subchannel rate = 3 Mbps $\rightarrow$ 20 subchannels at a rate of 3.008 Mbps

For maximum limit: $(2.28x10^{-6}) \times$ (VTA subchannel rate) = 3
VTA subchannel rate = 1.316 Mbps $\rightarrow$ 45 subchannels at a rate of 1.337 Mbps

Therefore, the subchannel breakout range the transport layer protocol must support is 20 to 45 subchannels. The equation for calculating the subchannel breakout number and data rate are,

Number of subchannels:

$$= \frac{2(139.38\ \mu\text{sec} + \sum_{i=1}^{n}(\frac{\text{fiber length}_i}{\text{propagation velocity}_i}) + \sum_{i=1}^{n-1}(5.45\ \mu\text{sec})60.16\ \text{Mbps}}{376\times3}$$

Data rate:

$$= \frac{60.16\ \text{Mbps}}{\text{number of subchannels}}$$

Hence to optimize the throughput efficiency at a application rate of 60.16 Mbps, the transport protocol at the provider must demultiplex the data stream from the application into x subchannels (where x depends on the connection fiber and switching node parameters). Each subchannel will be treated as a separate channel through the provider ATM protocol stack, the network, and the user ATM protocol stack. The user transport protocol then remultiplexes the x subchannels back into one channel for the application. Since each subchannel is treated as a separate channel in the ATM protocol stack, each subchannel will have a unique VCI and SN generator. The SN is used for error control via Go Back N, and the VCI is used to identify the subchannel to which a particular cell belongs. Both of these functions are provided for in the ATM/AAL layers, hence no extra transport overhead is needed to support these functions. The transport layer must however coordinate the establishment of the subchannels (set-up each subchannel and respective VCI), the multiplexing and demultiplexing of the subchannels, and error control for each subchannel. The implementation of these functions can be examined in two parts.

- Call Set-Up - the transport layer establishes the number of subchannels, coordinates with the management layer to allocate the appropriate multiplexing durations for each subchannel into the AAL layer (per SONET frame), coordinates with the management layer to establish unique VCI addresses for each subchannel (the management layer establishes the network routing based on the VCIs), and coordinates the subchannel multiplexing/demultiplexing timing with the far end transport protocol.

- Data transfer - the transport layer demultiplexes/remultiplexes the subchannels out-of/into the application and provides the Go Back N error control on a subchannel basis. It also coordinate with the management layer when rerouting is necessary due to network failures.

4.1.2 Multiplexing

The multiplexing function provides the means for dividing the provider application data stream into x subchannels and reassembling the subchannels at the user end. The number of subchannels has already been determined at call set-up and the subchannels have been established. Both demultiplexing and remultiplexing are examined in turn.

4.1.2.1 Demultiplexing

The demultiplexing function is implemented at the provider transport protocol. The provider application sends a video file transfer at 60.16 Mbps, this file is then demultiplexed into x subchannels. Each subchannel is then interfaced to the AAL layer. The AAL layer provides a SN function (modulo 7) on a 47 byte data unit basis, that is every 47 bytes of each subchannel is given a SN. At the user end, these 47 byte data units will be presented to the transport protocol for remultiplexing back into the original file sequence. To simplify processing, the transport demultiplexing and remultiplexing function also implemented on a 47 byte basis. In this manner each 47 byte data unit will contain a 47 byte chunk of the original file and maintain proper bit sequence within the data unit. Figure 4-2 depicts the demultiplexing scheme (on a 47 byte basis).

Each subchannel has a unique VCI identifier which has been determined at call set-up. The AAL layer downloads 47 byte data units from the input ports one at a time (one from each input port) in a round robin fashion. At each subchannel input port, the rate at which 47 byte data units (from the application file) are received can not exceed the rate at which they are passed to the AAL layer. On the other hand, in order to achieve high throughput, the rate of input data units (from the application) should not be less than the rate they are passed to the AAL layer. Hence, the switching mechanism at the transport layer demultiplexing function should be synchronous with the switching mechanism of the AAL layer (as one data unit is downloaded to the AAL layer, another is demultiplexed into the input port). If the switching rates are equal, every time a subchannel input port passes a data unit to the AAL layer, it will receive a data unit from the application file. Furthermore, each subchannel port will receive data units separated by x data units with respect to the application file sequence. Table 4.4

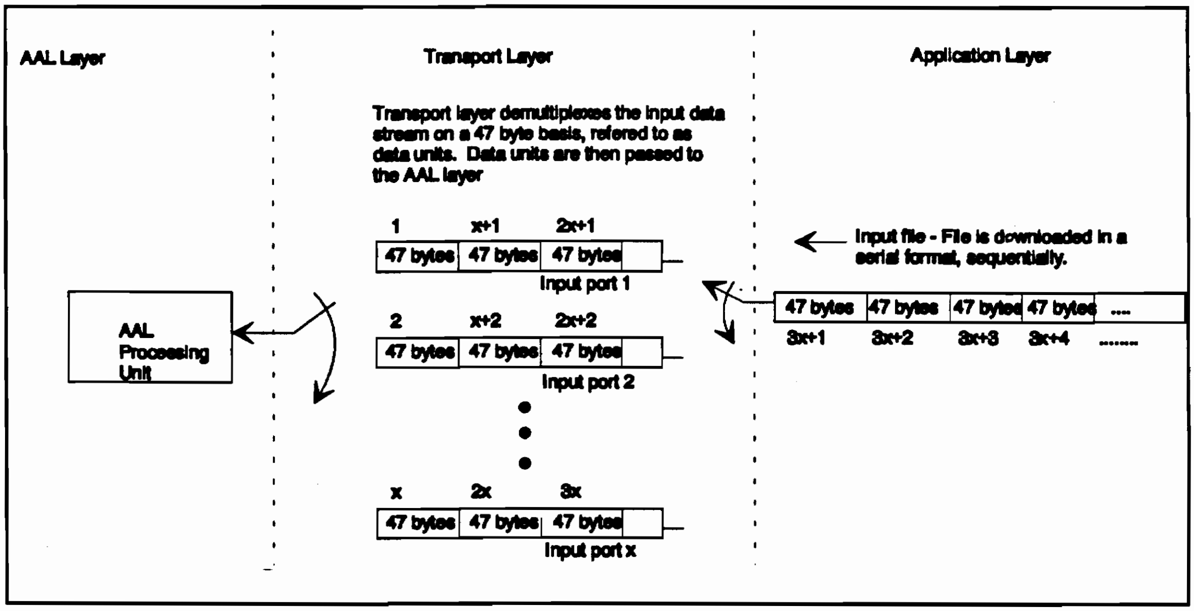

Figure 4-2 Demultiplexing Scheme

shows the data unit allocations for x number of transport input ports.

Table 4.4 Logical Data Unit to Input Port Assignments

Input port #	Data units (sequence number with respect to the input file)			
1	1	x+1	2x+1	...
2	2	x+2	2x+2	...
3	3	x+3	2x+3	...
.	.	.	.	.
x	x	2x	3x	...

The receiver input ports corresponding to each subchannel can also expect to receive the data unit as shown in Table 4.4. In this manner, the input video file can be demultiplexed into the transport subchannels and maintain high subchannel utilization while avoiding buffer overflows. The receiving transport protocol will also know the data unit sequence (from the subchannel VCI) to remultiplex the subchannels into the application video file by knowing that its transport input port will contain the data units from the table above, corresponding to the video file sequence. If the transport layers maintain this scheme (specified data units are passed through the designated subchannel ports as depicted in the Table 4.4) there will be no need to add extra SN overhead infomation, at the transport layer, to resequence data units into the original video file. The only parameter needed is x (subchannel breakout paramter), which is obtained at call set-up.

As a comparison, if we add a header field in the transport layer for sequencing, this would add (assuming that one input port would never be ahead or fall behind by more than 7 data units with respect to the other input ports) a minimum of,

$$\text{number of header bits} \geq \text{Int}[\ \log_2(7 \times 45)] = 9$$
where 45 is the maximum number of subchannels

Hence the 47 byte data unit would, at a minimum, contain 367 bits of infomation as opposed to 376, resulting in a decrease in infomation throughput and increase in processing delay and complexity.

To implement the error control scheme at the transport layer, each of the input ports must consist of 7 47 byte buffers and the error control algorithm. Furthermore, each input buffer will need to know the SN of the currently buffered data units (transport layer coordinates with the AAL). The error control time out clocks are commonly set as follows [5]:

- Time out clock - set to 2 x average round trip delay
- Time out clock at receiver - set to 2 x average round trip delay

The scheme described above would be ideal if data unit retransmissions never occurred and ACKs always arrived in sufficient time. This, however, is never the case. Recall, for the Go Back N error control scheme, that a data unit must be stored until the receiver acknowledges that it has received the data unit. Hence the rate that the subchannel input ports send data units to the AAL layer decreases when errors occur and retransmissions are needed. It was previously detemined that, to increase throughput, SN at the transport protocol would not be implemented, hence the demultiplexing scheme must insure data units are downloaded to the appropriate input ports as depicted in Table 4.4. For instance, data unit x+1 (corresponding to the x+1th 47 byte unit of the video file) must be demultiplexed into port 1. If it is not, then the receiver has no way of knowing that this is the x+1th data unit (w/o extra SN at the transport layer). If the x+1th data unit is sent to port 2, then the receiver will think that it is the x+2th data unit, and remultiplex it accordingly. To account for this scenario, the following two functions are proposed,

- The entire demultiplexing (file transfer) could halt and wait for the subchannel to retransmit data units until the port clears its buffer window.
- The data unit could be buffered outside of the window buffer.

The demultiplexing scheme can not go to the next available port and download the data unit.

The first approach will require no extra buffering capacity, however, throughput will be decreased not only for the subchannel unable to accept the data unit, but also for all subchannels. For this scheme, the entire demultiplexing rate is delayed when any of the subchannel input ports windows has become exhausted (ACK or NAK has been lost or delayed). In order to get a handle on the significance of this operational implementation, the probability that a window at the subchannel port becomes exhausted must be determined. If the probability of this occurrence is insignificant, this operational scheme could be implemented without effecting throughput efficiency and total transmission time. By examining the demultiplexing and error control operation, the probability can be calculated .

At the commencement of data transfer, the first data unit is downloaded to input port 1, and is sent to the AAL layer. Data unit 1 must be kept in the input port buffer until an ACK is received for it or any subsequent data unit. This means that any time an ACK is not sent before the subsequent data unit is downloaded from the application, the input buffer must store an additional data unit. Once 7 data units are stored in the input buffer, it can no longer accept data units from the application (transport demultiplexing function) until an ACK is received and it can discard the ACKed data units from the buffer.

From the previous analysis, average round trip delay with respect to the connection parameters was calculated.

$$T = 2(139.38 + \sum_{i=1}^{n} [\frac{(\text{fiber length})_i}{(\text{velocity})_i}] + \sum_{i=1}^{n-1} (\text{queuing delay}) + \frac{376}{\text{VTA rate}})\ \mu\text{sec}$$

where the VTA subchannel rate is,

$$R = 60.16\ \text{Mbps}[\frac{(139.38\ \text{Mbps} + \sum_{i=1}^{n} (\frac{(\text{fiber length})_i}{(\text{velocity})_i}) + \sum_{i=1}^{n-1} (5.45\ \text{Mbps}))}{376 \times 3}]$$

The VTA subchannel calculates a fixed rate, where average values of queuing delay per node were used. The actual queuing delay experienced by specific data units will vary depending on traffic loading conditions at each switch. This is the only parameter in the round trip equation that will vary from the initial delay calculation (done at call set-up).

Each subchannel input port can be modeled as a one dimensional Markov chain where the state corresponds to the number of buffered 47 byte data units. The Markov chain is modeled in discrete time, where the time unit is equal to the inter-arrival time of the data units from the application. A state change is only possible at these discrete time count.

By calculating the number of subchannels such that the subchannel utilization is ~ 1, we assure that the average round trip delay is always equal to 3 times the data unit inter-arrival time for each subchannel breakout parameter. Figure 4-3 depicts the delay relationship for data unit transmission with average round trip delay.

From Figure 4-3, it is evident that if errors never occur and the data units and corresponding ACKS are always delivered at or below the average end-to-end delay, the error control scheme will always have a window size of 4. Therefore, there will always be 3 data units in the subchannel input port. In the Markov model, this relates to always being in state three. Furthermore, each data unit transmission (with respect to a single input port) is independent of the one before it (unless the input port error control window is exhausted corresponding to the

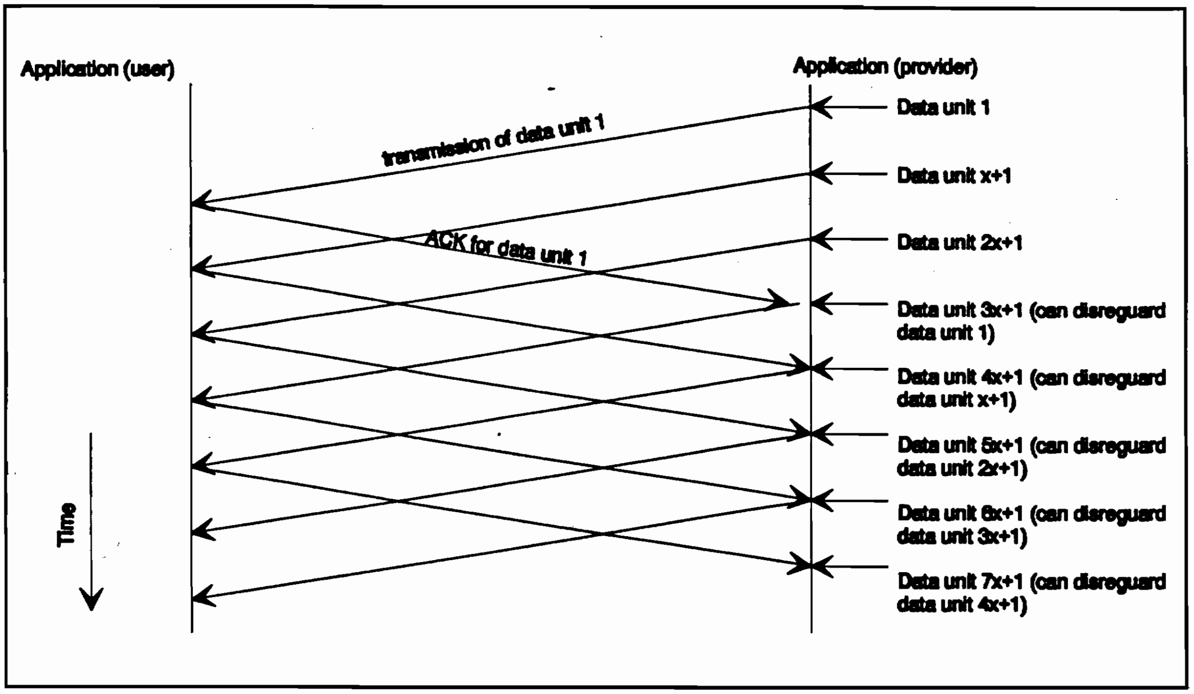

Figure 4-3 Data Unit Average Round Trip Delay

Markov chain being in state 7). Each data unit is transmitted via a separate SONET frame (the minimum subchannel breakout is 20 corresponding to one data unit per subchannel per SONET frame) and from the Go Back N scheme, data units can be transmitted without prior data unit acknowledgments (up to 7 ahead). Figure 4-4 depicts the one dimensional Markov chain used to model the input port buffer states for each subchannel.

The probability of moving up one state will correspond to a error in data unit reception (error control returns a NAK) or a delay in round trip time. The probability distribution that an error occurs in the data unit and the error control returns a NAK is unifom for all data units with an average probability of 10^{-9}. The probability distribution for round trip delay is dependent upon queuing delay (all other delays are fixed for a particular connection) where the queuing delay is discrete based on cell transmission times (for a STS-3 link , cell transmission time = 2.725 μsec). The mean is 2(2.725 μsec). The probability distribution can be approximated by the following plot shown in Figure 4-5 [11].

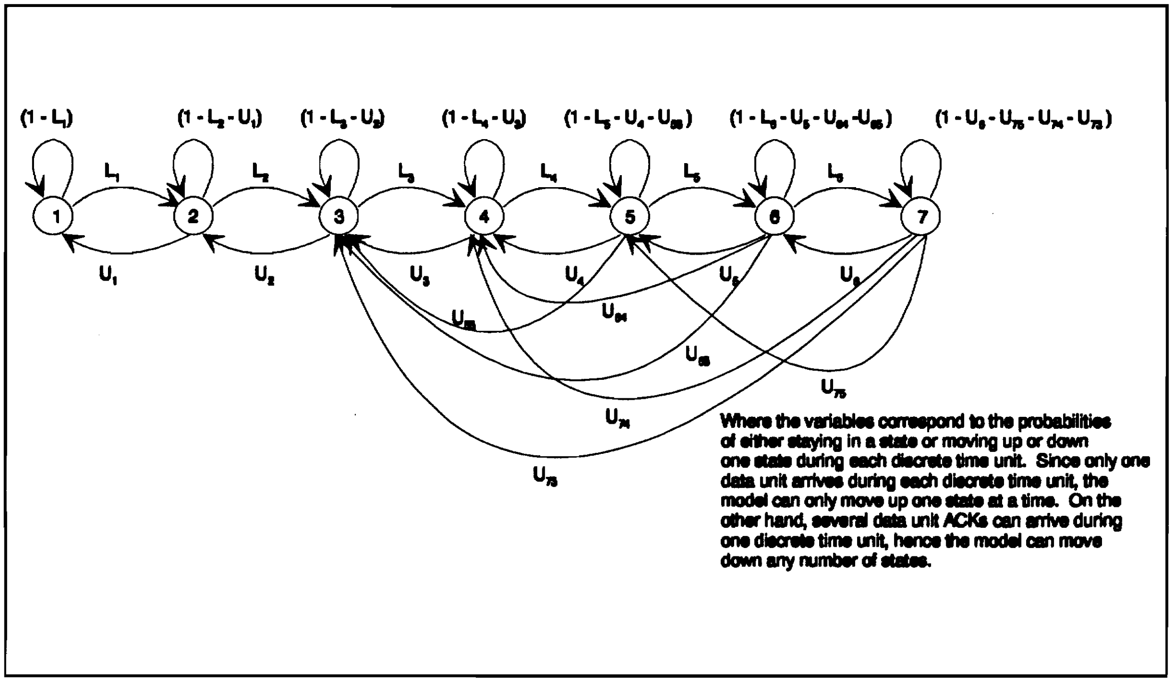

Figure 4-4 Markov Chain Model for a Input Port

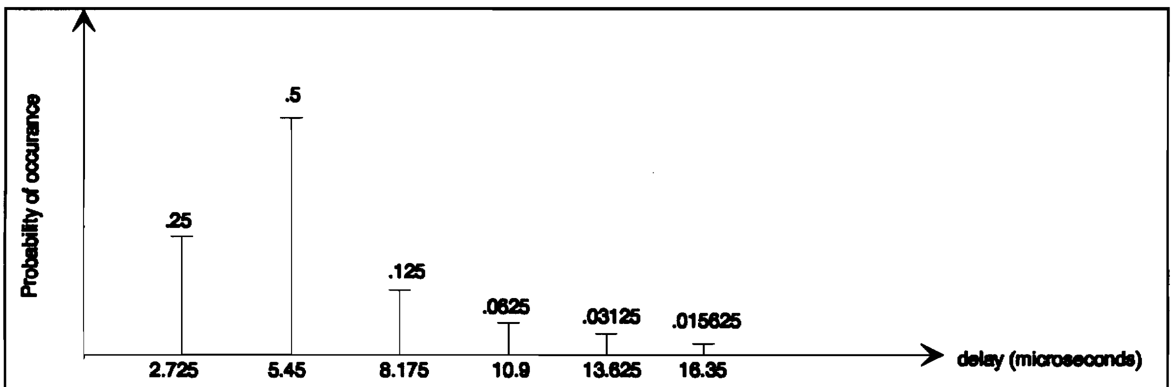

Figure 4-5 Queuing Delay Probability Distribution

Examining the distribution, it can be seen that there is a 25 % probability that a single switch will delay the arrival data unit or ACK just enough to push the Markov chain to state 4. A data unit will pass through a number of switches based on connection parameters. The probability of increased delay with respect to the average round trip delay, increases as the number of switches in the round trip path increases. The worst case scenario for each subchannel breakout parameter is when the maximum number of switches comprises the connection. An estimate of this number for each subchannel breakout parameter can be derived by assuming that the connection is comprised of all 5 km fiber lengths. If the connection is made up of all 5 km sections, this will yield the maximum number of switches per subchannel breakout parameter. This maximum was calculated for each subchannel breakout paramter. Based on the maximum number of switches and the inter-arrival time for each subchannel breakout parameter, a simulation was run using MATLAB to determine the probability that the round trip delay will exceed 3, 4, 5, and 6 times the inter-arrival times for each subchannel breakout parameter. These probabilities correspond to probabilities of the Markov chain being in state 3, 4, 5, 6, and 7. Table 4.5 presents the results of the simulation. Appendix B describes the calculations used to derive these probabilities and contains the MATLAB algorithm used. The maximum number of switches to inter-arrival time ratio, given in column 6 of Table 4.5, is a useful measure to gauge the extent in which queuing delay variations effects the Markov probabilities. The higher this ratio, the greater the number of round-trip queues there are compared to the inter-arrival time. Hence, the higher the probability will be of moving up states in the Markov chain. The Markov chain probabilities for the subchannel breakout parameter with the highest ratio are shown in Figure 4-6.

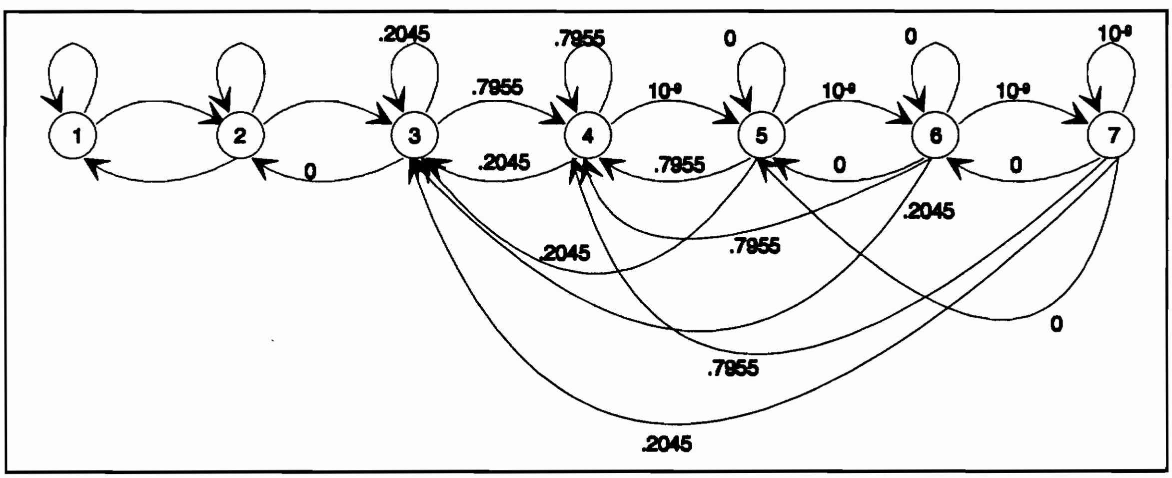

Figure 4-6 Markov Chain State Probabilities for a Subchannel Breakout Parameter of 45

Table 4.5 Connection Parameters and State Shift Probabilities for Various Subchannel Breakout Parameters

Number of Subchls	Data Rate	Avg. round trip delay	Inter-data unit time	Max # of switches	Max # of switches to arrival time	prob. of same state	prob. of 1shift up	prob. of 2 shifts up	prob. of 3 shifts up	prob. of 4 shifts up
20	3008000	0.000375	0.000125	1	8000	0.6263	0.3737	0	0	0
21	2864762	0.000394	0.0001313	1	7619.048	0.6251	0.3749	0	0	0
22	2734545	0.000413	0.0001375	1	7272.727	0.6237	0.3763	0	0	0
23	2615652	0.000431	0.0001438	2	13913.04	0.4809	0.5191	0	0	0
24	2506667	0.00045	0.00015	2	13333.33	0.4798	0.5202	0	0	0
25	2406400	0.000469	0.0001563	2	12800	0.4791	0.5209	0	0	0
26	2313846	0.000488	0.0001625	3	18461.54	0.4224	0.5776	0	0	0
27	2228148	0.000506	0.0001688	3	17777.78	0.4214	0.5786	0	0	0
28	2148571	0.000525	0.000175	3	17142.86	0.4205	0.5795	0	0	0
29	2074483	0.000544	0.0001813	4	22068.97	0.3521	0.6479	0	0	0
30	2005333	0.000563	0.0001875	4	21333.33	0.3513	0.6487	0	0	0
31	1940645	0.000581	0.0001938	4	20645.16	0.3502	0.6498	0	0	0
32	1880000	0.0006	0.0002	5	25000	0.2606	0.7394	0	0	0
33	1823030	0.000619	0.0002063	5	24242.42	0.2592	0.7408	0	0	0
34	1769412	0.000638	0.0002125	5	23529.41	0.2579	0.7421	0	0	0
35	1718857	0.000656	0.0002188	6	27428.57	0.3001	0.6999	0	0	0
36	1671111	0.000675	0.000225	6	26666.67	0.2975	0.7025	0	0	0
37	1625946	0.000694	0.0002313	6	25945.95	0.2923	0.7077	0	0	0
38	1583158	0.000713	0.0002375	7	29473.68	0.2201	0.7799	0	0	0
39	1542564	0.000731	0.0002438	7	28717.95	0.2149	0.7851	0	0	0
40	1504000	0.00075	0.00025	7	28000	0.2171	0.7829	0	0	0
41	1467317	0.000769	0.0002563	8	31219.51	0.2281	0.7719	0	0	0
42	1432381	0.000788	0.0002625	8	30476.19	0.227	0.773	0	0	0
43	1399070	0.000806	0.0002688	8	29767.44	0.2242	0.7758	0	0	0
44	1367273	0.000825	0.000275	8	29090.91	0.2231	0.7769	0	0	0
45	1336889	0.000844	0.0002813	9	32000	0.2045	0.7955	0**	0	0

Note: Probabilities do not include error probability, only the probability due to queuing delay

**Note: From calculations, probability is approximatly 1.0E-33 This probability decreases as the number of subchannels decreases.

Note, the Markov chain will never be in states 1 or 2 due to the round trip delay. The probability that the chain will be in state 7 can only occur if a data unit is delayed by 3 inter-arrival times plus the probability that a data unit or NAK is received in error. If a cell is in error, it will have to be retransmitted, requiring an extra 3 inter-arrival time periods pushing the Markov state to 7. Furthermore, by setting the window time out values at twice the average round trip time, a lost data unit or ACK will also push the Markov state to 7. The probability of a lost data unit or ACK (resulting in a time-out at the receiver or transmitter) is assumed to be included in the probability of error value, 10^{-9}. This assumption is based on an ideal transmission network. Furthermore, since the time-outs are 2 times the average round trip delay, a lost data unit or ACK will effect the state model in the same manner as an error detection. The probabilities corresponding to Figure 4-4 are given in Table 4.6 for each subchannel breakout parameter.

Given a 2 hr video the file consists of

$$\frac{(1.5 \times 10^6)\ 7200}{47 \times 8} = 2.87 \times 10^7$$

data units. The average time length that the demultiplexing scheme at the transport layer will have to pause and wait its buffer to clear can be calculated. From Table 4.6, it can be seen that the probability of a single input port reaching state 7 is 10^{-9} for all subchannel breakout parameters. The probability of staying in state 7 (second attempt to send the data unit) is also 10^{-9} for each subchannel breakout parameter. Hence the probability of reaching state 7 and staying there for more than 1 discrete time unit is $10^{-9}x10^{-9} = 10^{-18} \sim 0$. Table 4.7 gives the average durations for which the demultiplexing scheme will pause for a 2 hr video file transfer for each subchannel breakout parameter as well as the total transmission duration.

From these values it can be seen that during a file transfer (assume a two hour video) the Markov chain will go to state 7 an average of once during the file transfer. Given this information, it can be concluded that adding extra buffering to the input port is not needed. Therefore, the demultiplexing function is designed to stop and wait if a particular input port is

Table 4.6 Markov State Probabilities for Various Subchannel Breakout Parameters

	$1-L_3-U_2$	L_3	U_2	$1-L_4-U_3$	L_4	U_3	$1-L_5-U_4-U_{53}$	L_5	U_4	$1-L_6-U_5-U_{64}-U^{63}$	L_6	U_5	$1-U_6-U_{75}-U_{74}-U_{73}$	U_6	U_{53}	U_{64}	U_{75}	U_{74}	U_{63}	U_{73}
20	.6263	.3737	0	.3737	10^{-9}	.6263	0	10^{-9}	.3737	0	10^{-9}	0	10^{-9}	0	.6263	.3737	0	.3737	.6263	.6263
21	.6251	.3749	0	.3749	10^{-9}	.6251	0	10^{-9}	.3749	0	10^{-9}	0	10^{-9}	0	.6251	.3749	0	.3749	.6251	.6251
22	.6237	.3763	0	.3763	10^{-9}	.6237	0	10^{-9}	.3763	0	10^{-9}	0	10^{-9}	0	.6237	.3763	0	.3763	.6237	.6237
23	.4809	.5191	0	.5191	10^{-9}	.4809	0	10^{-9}	.5191	0	10^{-9}	0	10^{-9}	0	.4809	.5191	0	.5191	.4809	.4809
24	.4798	.5202	0	.5202	10^{-9}	.4798	0	10^{-9}	.5202	0	10^{-9}	0	10^{-9}	0	.4798	.5202	0	.5202	.4798	.4798
25	.4791	.5209	0	.5209	10^{-9}	.4791	0	10^{-9}	.5209	0	10^{-9}	0	10^{-9}	0	.4791	.5209	0	.5209	.4791	.4791
26	.4224	.5776	0	.5776	10^{-9}	.4224	0	10^{-9}	.5776	0	10^{-9}	0	10^{-9}	0	.4224	.5776	0	.5776	.4224	.4224
27	.4214	.5786	0	.5786	10^{-9}	.4214	0	10^{-9}	.5786	0	10^{-9}	0	10^{-9}	0	.4214	.5786	0	.5786	.4214	.4214
28	.4205	.5795	0	.5795	10^{-9}	.4205	0	10^{-9}	.5795	0	10^{-9}	0	10^{-9}	0	.4205	.5795	0	.5795	.4205	.4205
29	.3521	.6479	0	.6479	10^{-9}	.3521	0	10^{-9}	.6479	0	10^{-9}	0	10^{-9}	0	.3521	.6479	0	.6479	.3521	.3521
30	.3513	.6487	0	.6487	10^{-9}	.3513	0	10^{-9}	.6487	0	10^{-9}	0	10^{-9}	0	.3513	.6487	0	.6487	.3513	.3513
31	.3502	.6498	0	.6498	10^{-9}	.3502	0	10^{-9}	.6498	0	10^{-9}	0	10^{-9}	0	.3502	.6498	0	.6498	.3502	.3502
32	.2606	.7394	0	.7394	10^{-9}	.2606	0	10^{-9}	.7394	0	10^{-9}	0	10^{-9}	0	.2606	.7394	0	.7394	.2606	.2606
33	.2592	.7408	0	.7408	10^{-9}	.2592	0	10^{-9}	.7408	0	10^{-9}	0	10^{-9}	0	.2592	.7408	0	.7408	.2592	.2592
34	.2579	.7421	0	.7421	10^{-9}	.2579	0	10^{-9}	.7421	0	10^{-9}	0	10^{-9}	0	.2579	.7421	0	.7421	.2579	.2579
35	.3001	.6999	0	.6999	10^{-9}	.3001	0	10^{-9}	.6999	0	10^{-9}	0	10^{-9}	0	.3001	.6999	0	.6999	.3001	.3001
36	.2975	.7025	0	.7025	10^{-9}	.2975	0	10^{-9}	.7025	0	10^{-9}	0	10^{-9}	0	.2975	.7025	0	.7025	.2975	.2975
37	.2923	.7077	0	.7077	10^{-9}	.2923	0	10^{-9}	.7077	0	10^{-9}	0	10^{-9}	0	.2923	.7077	0	.7077	.2923	.2923
38	.2201	.7799	0	.7799	10^{-9}	.2201	0	10^{-9}	.7799	0	10^{-9}	0	10^{-9}	0	.2201	.7799	0	.7799	.2201	.2201
39	.2149	.7851	0	.7851	10^{-9}	.2149	0	10^{-9}	.7851	0	10^{-9}	0	10^{-9}	0	.2149	.7851	0	.7851	.2149	.2149
40	.2171	.7829	0	.7829	10^{-9}	.2171	0	10^{-9}	.7829	0	10^{-9}	0	10^{-9}	0	.2171	.7829	0	.7829	.2171	.2171
41	.2281	.7719	0	.7719	10^{-9}	.2281	0	10^{-9}	.7719	0	10^{-9}	0	10^{-9}	0	.2281	.7719	0	.7719	.2281	.2281
42	.227	.773	0	.773	10^{-9}	.227	0	10^{-9}	.773	0	10^{-9}	0	10^{-9}	0	.227	.773	0	.773	.227	.227
43	.2242	.7758	0	.7758	10^{-9}	.2242	0	10^{-9}	.7758	0	10^{-9}	0	10^{-9}	0	.2242	.7758	0	.7758	.2242	.2242
44	.2231	.7769	0	.7769	10^{-9}	.2231	0	10^{-9}	.7769	0	10^{-9}	0	10^{-9}	0	.2231	.7769	0	.7769	.2231	.2231
45	.2045	.7955	0	.7955	10^{-9}	.2045	0	10^{-9}	.7955	0	10^{-9}	0	10^{-9}	0	.2045	.7955	0	.7955	.2045	.2045

Table 4.7 Total Expected Pause Times and Overall Transmission Duration for Various Subchannel Breakout Parameters

Subchannel breakout parameter	Discrete Markov time unit (inter-data unit arrival time) (.01E-6 sec)	Average number of times reach state 7 for 2 hr. video (summed over all subchannels)	Total demultiplexing pause time due to exceding window buffer size for a 2 hr. video (1.0E-6 sec)	Total file transfer duration for a 2 hr. video given the average pause delay times (sec)
20	125	0.574	502.25	179.521779
21	131.3	0.6027	553.9416	179.521831
22	137.5	0.6314	607.7225	179.521884
23	143.8	0.6601	664.4567	179.521941
24	150	0.6888	723.24	179.522000
25	156.3	0.7175	785.0167	179.522062
26	162.5	0.7462	848.8025	179.522125
27	168.8	0.7749	915.6218	179.522192
28	175	0.8036	984.41	179.522261
29	181.3	0.8323	1056.272	179.522333
30	187.5	0.861	1130.063	179.522407
31	193.8	0.8897	1206.967	179.522484
32	200	0.9184	1285.76	179.522562
33	206.3	0.9471	1367.707	179.522644
34	212.5	0.9758	1451.503	179.522728
35	218.8	1.0045	1538.492	179.522815
36	225	1.0332	1627.29	179.522904
37	231.3	1.0619	1719.322	179.522996
38	237.5	1.0906	1813.123	179.523090
39	243.8	1.1193	1910.197	179.523187
40	250	1.148	2009	179.523286
41	256.3	1.1767	2111.117	179.523388
42	262.5	1.2054	2214.923	179.523492
43	268.8	1.2341	2322.083	179.523599
44	275	1.2628	2430.89	179.523707
45	281.3	1.2915	2543.093	179.523820

unable to accept the next data unit. The input port then clears its buffer and signals the demultiplexing function to resume. Note, probability of link failure was not included in this analysis. If this probability is significant, it may be wise to add extra buffering to each input port to avoid low throughput operation.

4.1.2.2 Remultiplexing

At the receiver end, the transport functions to remultiplex the subchannels back into the original file. If a constant rate remultiplexing function is implemented, difficulties arise in operating the error control for each subchannel while simultaneously trying to remultiplex the subchannels, Figure 4-7 depicts the problem.

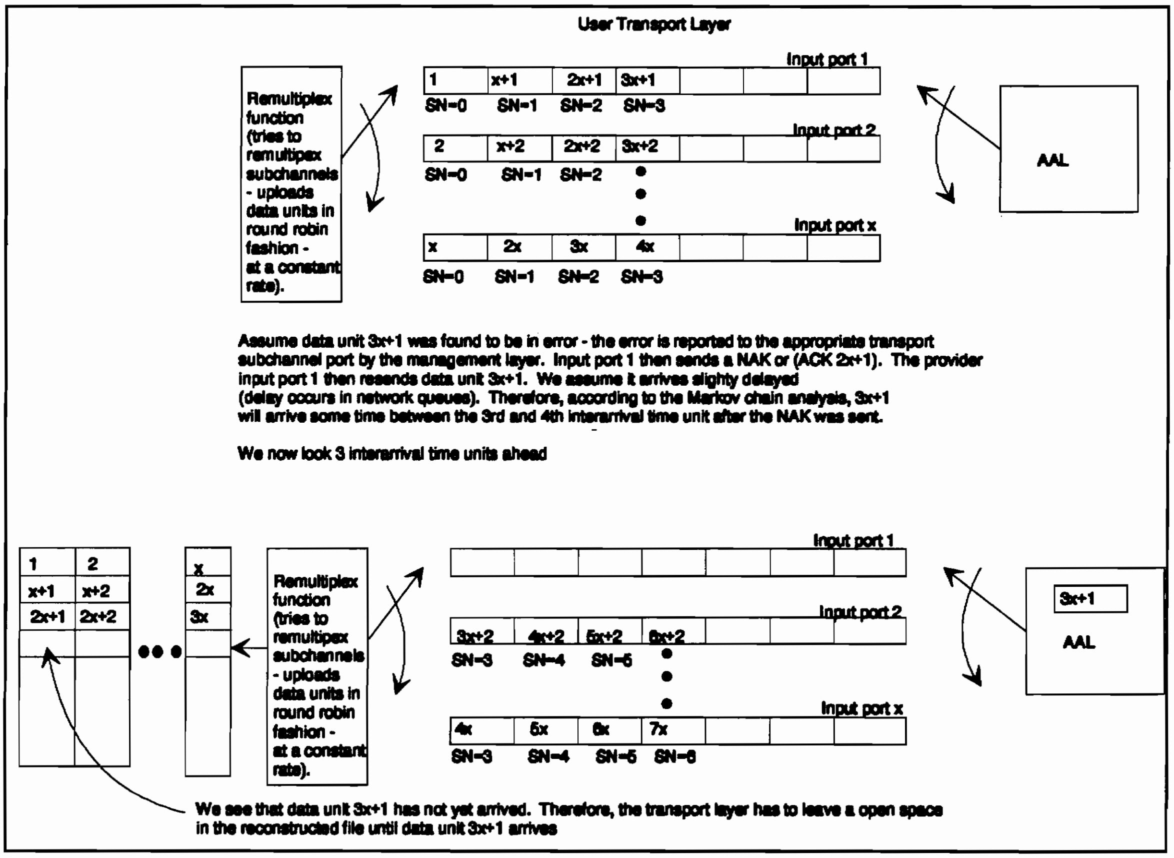

Figure 4-7 Remultiplexing Scheme

The transmitter operation will halt after 4 inter-arrival time units, because port 1, at the transmitter will have exhausted its window. This assures that each subchannel will never exceed 7 data units beyond the NAKed data unit. The receiver will discard the data units received after 3x+1 and wait for the 3x+1 data unit to be retransmitted. During the fourth inter-arrival time unit slot (as depicted in Figure 4-7), the retransmitted 3x+1 data unit will not have arrived yet, and the remultiplexing function will not be able to remultiplex the correct data unit needed to maintain proper file sequencing. If the remultiplexing operation is designed such that it also stops and waits for the proper data unit, similar to the demultiplexing, this would assure that proper data unit sequencing is maintained. Furthemore, since the demultiplexing function also pauses when an error occurs (or its window is exhausted) the remultiplexing function will essentially shadow the demultiplexing operation except that it will be 1.5 inter-arrival time units ahead. The receiver input ports can also be modeled as Markov chains, these models will be exactly the same as the transmitting input port models. Also, since the probability of reaching state 7 is 10^{-9}, subchannel throughput will essentially remain at a utilization level of ~ 1.

This function can easily be implemented at the receiver. The receiver input ports consist of 7 47 byte buffers and the error control algorithm. The remultiplexing rate is fixed, as is the demultiplexing rate. Whenever a particular subchannel input port does not have the proper data unit to present to the remultiplexing function, corresponding to an empty buffer, the remultiplexing operation will pause. The demultiplexer will also pause (1.5 inter-arrival time units later) because the pause condition will also appear at the transmitter input port. The remultipexing function will have to wait at the empty port until 7 data units are received (since the demultiplexing port will clear its buffer). Once the expected data units are received, the remultiplexing operation restarts. The remultiplexing operation will mirror the demultiplexing operation by initially collecting 3 data units in each buffer before the remultiplexing commences. This insure that the receiver input buffer ports do not deplete themselves unless the corresponding transmitter input port reaches state 7. Furthermore, since the demultiplexing operation will pauses when a port reaches state 7, this insures that the input port buffers at the receiver never overflows.

4.1.3 Call Set-Up

Next the call set-up function is examined. To operate over the ATM protocol stack, the transport layer must first determine the number of subchannels it will establish in order to optimize the overall VTA data rate. Each subchannel must then be established through the network and with the video provider (each with a unique VCI). Finally, the two ends (user and provider) must agree that the connection has been established and that data transfer may commence. Each phase of the call set-up process will be examined in order to develop the call set-up procedure. The call set-up process starts with the residential user requesting a video. This request is passed from the application to the transport protocol. The transport protocol communicates the request to the management layer. It has already been established that the ATM protocol parameters will always be the same for every VTA connection

- PT code for data cells is 000
- GFC is set to uncontrolled access 0000
- CLP is set to high priority 0
- Overall data rate is 60.16 Mbps at the AAL layer (67.84 at the TC layer prior to SONET overhead). This allows the VTA to send 20 data units in each SONET frame.

The management layer will first route the video request cell to the common video provider facility which will look up the specific video provider address that has the requested video in storage (not every provider facility will store every video). The common video provider facility will send back the address of the specific video provider facility containing the requested video to the user. The management layer will then create a call request cell with the above ATM protocol parameters and send the cell through the ATM protocol stack to the local ATM node. The cell includes the above parameters, the requested video, the destination address (of the specific video provider facility), and the source address. This cell is routed through the network to the video facility. The local network node determines if there is a virtual path connection (VPC) already defined between the source and the destination where a VPC is a sequential collection of VPs. If there exists at least one VPC that has enough bandwidth to accommodate the new connection, one of these paths is chosen to establish the connection. If no VPC exists, a new VPC needs to be established. If there are existing VPCs but none that can accommodate the new connection, the network node requests that the amount of bandwidth allocated to the VPs that carry the connection be increased. If this request for increased bandwidth is granted, the connection is established, otherwise it is rejected. Hence the original call request cell from the user (if accepted) will reserve sufficient bandwidth through the network for the video transfer. Each subchannel (with corresponding VCI) will be routed via the same VPI through the network.

The video provider will then acknowledge the request (if it has sufficient capacity to handle the transfer) and send a request received cell back to the user, acknowledging that the request has been approved. Once the user receives the request acknowledgment cell, it will send a ready to receive cell to the provider to initiate the video file transfer. This is a simple 3-way handshake connection set-up scheme used in many transport protocols today.

The unique problems facing the VTA transport protocol is that it must somehow determine the average round trip delay (to determine the subchannel breakout parameter) and coordinate the VCI addresses with its management layer and the provider management layer. The VCI will only be used by the end user and provider AAL layers to separate the subchannels (network routes all subchannels via a single VPI).

The following approaches to providing link parameters are proposed and compared,

- Store connection parameter information at provider locations;
- Store connection parameter information distributively throughout network (each node contains link parameters for interfacing links) As call set-up cell travels from user to provider, each node appends information to cell for fiber link and queue;
- Once connection is set-up, user send several cells with timers to calculate average end-to-end delay, then calculates VTA subchannel rate and then resets overall channel to x subchannels (VCIs must be reestablished)

Storing all link parameters at each local network node will require that each node store all link and node parameters (propagation delay, fiber length, and average queuing delay per node) in its local buffer. Furthermore, changes in traffic loading and fiber parameters will have to be reported constantly. This approach seems to be inefficient in terms of buffer space, node processing, and internode communications.

The second approach is to task each network node to store the link parameters (propagation velocity, fiber length, and average queuing delay- with the proposed connection in place) of all interfacing links. Thus, as the video request cell is routed through the network, each node can append the parameters of the queue and output link (we assume the local link parameters are known by the user). Hence, upon reception of the request cell at the provider end, the cell will contain all necessary parameters to calculate the subchannel breakout parameter.

In the scenario, the provider is tasked to calculate the subchannel breakout parameter upon reception of the original video request cell. If the provider can accommodate the connection, it will assign x number of subchannels to the VTA application, each of which will have a unique VCI. The provider must coordinate with its management layer to inform the management layer of the number of subchannels and get the VCI assignments for each. The AAL multiplexing assignments are then coordinated by the management layer. The transport protocol demultiplexing is established via the management layer AAL multiplexing durations.

The provider end then sends back a request acknowledged cell along with (in order - from input port 1 through x) the subchannel VCI addresses. The user then receives these cells and establishes its subchannels and reports the number of subchannels and VCI assignments (from the provider end) to its management layer. The user then sends a ready to receive cell to the provider to initiate the video file transfer.

The third approach would require a lengthy call set-up procedure and would be unable to guarantee average link parameters (cell would generate round trip delay times based on current conditions of the network).

Option 2 is recommended as the call set-up procedure, this is a slightly modified version of the currently utilized 3-way handshake protocol. Currently under CCITT recommendations, there is no standard for reporting link and queuing parameters to the call request cell. Up to now, queuing and propagation delay have been insignificant compared to transmit times. As a result of this analysis, it is recommended that this function be included in the standard ATM node operating procedures during call set-up. All very high speed connections could then use the subchannel breakout method to more efficiently utilize the available network speeds.

This function could easily be implemented at each node with the following,

- Reserve a bit in the call set-up cell to signify that parameter information is desired;
- The ATM node uploads the proposed connection bandwidth and recalculates average queuing delay;
- The ATM node writes the average queuing delay and output link parameters (propagation velocity and fiber length) to the video request cell;
- The request cell is then routed to the next node.

The video request cell must then carry the following information fields, as depicted in Figure 4-8.

Header Fields- contains standard ATM cell header - Routing addresses via VCI and VPI (to route request cell) - PT - GFC - CLP		
Information Field Breakout		
Required header parameter	Ranges parameters must support	Required number of bits
data rate	0-622.05 Mbps	32
video request	up to 10^9	40 (conservative estimate)
destination address	VPI and VCI yield 6.6843546×10^8 unique addresses	28
source address	VPI and VCI yield 6.6843546×10^8 unique addresses	28
link parameters		
link identification (sequence in the connection)	1-16 (assuming 16 fibers is maximum number any connection can have)	4
queuing delay (normalized to cell duration)	1-8	3
propagation velocity ($\times 10^7$ m/s)	1-30	5
fiber length (km)	1-20	5

Figure 4-8 Call Set-up Cell

The total information field is,

$$128 \text{ bits } + (\text{number of links})17$$

An ATM cell information field consists of 376 bits, hence the information field can support,

$$\frac{376 - 128}{17} = 14 \text{ (links)}$$

This more than sufficient for reasonable connections from provider to user.

4.1.4 Other Transport Protocol Functions

Up to this point, the following transport layer functions have been defined for the VTA,

- Addressing - make use of VCI and VPI and demultiplexing subchannels into designated input ports in order to maintain overall video file sequence;
- Connection establishment and termination - use the modified 3-way handshake for connection establishment and a standard 3-way or 2-way handshake for termination;
- Buffering -buffering is tied into error control scheme, multiplexing is halted to insure error control window buffers (input port buffers) will not overflow;
- Segmentation and reassembly - demultiplexing and remultiplexing scheme discussed in Section 4.1.2;
- Error recovery and control - use a Go-Back-N error control scheme over subchannels.

The remaining standard transport protocol functions are,

- Flow and rate control
- Handling duplicated cells
- Priority handling
- Control signaling

Flow and rate control are provided inherently through the error control scheme. By synchronizing the transport demultiplexing /remultiplexing clocks with the AAL multiplexing clock, proper flow control is assured. Furthermore, when any of the subchannel Go-Back-N window buffers (input ports) are filled, the transport multiplexing operation pauses and waits until the subchannel window clears. This provides the flow and rate control over the demultiplexing and remultiplexing schemes which insures that the subchannels will not become congested (subchannels are dedicated for duration of the connection).

Handling duplicated cells. This is also handled by the error control scheme. The receiver input port will only acknowledge data units with proper SN sequence. All other data units are disregarded. Therefore, if a data unit arrives in duplicate (same SN) the second one will be disregarded.

Priority handling. This function is not needed in the VTA transport protocol because each data unit will be given high priority indication. Furthermore, there will not be contention for bandwidth because the downloading rates are matched to the subchannel rates and remain constant throughout the connection (dedicated bandwidth for each subchannel).

Control signaling can either be in-band or out-of-band. The error control scheme for each subchannel requires that the SN ACK/NAK signaling parameter be delivered in-band for each subchannel. This signaling is most efficiently done over the reverse direction of the full duplex subchannel. This is the only transport functional signaling parameter needed in the proposed design. Recall,

- Flow control is a derivative of the error control schemes;
- Channel parameters and data rates do not change;
- File sequencing is implicit from the port numbers.

Maintenance and other internal transport layer communications (not the focus of this report) can be communicated over a low rate out band channel so as not to effect the data throughput.

4.2 Current Transport Protocols

Various current transport protocols were analyzed for VTA compatibility. Since none of these protocols were designed to operate over an ATM protocol, none of them provide an efficient means of supporting the VTA when compared to the VTA specific transport protocol design. The major issues are that none of these protocol work on a 47 byte basis (which will require the SAR function at the AAL to be active, increasing processing time) none support the subchannel breakout calculation function, and all have extra transport header fields. Table 4.8 [1] list the current protocols and there functions as compared to the VTA specific protocol. Also a comparison of average transmission duration is done based on the transport protocol and is given in column 8 of Table 4.8.

Of the current transport protocols, the TP4 protocol is most similar to the VTA specific protocol. Modification to the TP4 protocol should include,

- Operation on a 47 byte data unit basis;
- Allow for subchannel breakout parameter calculation;
- Allow subchannel breakout parameter to determine demultiplexing/remultiplexing rates;

- Eliminate Transport header fields (or allow it to be an option);
- Support out-band signalling (other than error control).

These are significant modification which may not be feasible with respect to the software algorithms. A consideration not examined in this comparison is software run times. It may be that due to the complexity of the protocols, that the software may not be able t support the transfer rates needed by the VTA.

Table 4.8 Current Transport Protocol Comparison

Transport Protocol	Connection Establishment	Connection Termination	Connection Signaling Parameters	Multiplexing Schemes	Flow Control	Error Control Scheme	Estimated Channel Utilization/ Transmission Duration (based on 60.16 Mbps channel rate) **	Comments
APPN	2-way handshake (does not support subchannel breakout parameter calculation)	2-way handshake	Set-up	None	None	Abort	Maximum of 40	- No demultiplexing capability - No error control capability
Datakit	2-way handshake (does not support subchannel breakout parameter calculation)	2-way handshake	Set-up, connection update, mode selection	None	Adaptive window	Go-Back-N	Max of 40% (excluding header fields) 8 min	- No demultiplexing capability - Operates on a 9 bit data unit basis - Additional header fields
Delta-T	Implicit - connection established at arrival of first packet (does not support subchannel breakout parameter calculation)	Timer-based	Connection update	Only supports multipexing several application channels into a single channel	Adaptive window	Go-Back-N	Max of 40% (excluding header fields) 8 min	- No demultiplexing capability - Additional header fields
NETBLT	2-way handshake (does not support subchannel breakout parameter calculation)	2-way handshake	Set-up, connection update	Only supports multipexing several application channels into a single channel	Cumulative window or rate control	Selective-Reject	Max of 40% (excluding header fields) 8 min	- No demultiplexing capability - Additional header fields

Transport Protocol	Connection Establishment	Connection Termination	Connection Signaling Parameters	Multiplexing Schemes	Flow Control	Error Control Scheme	Estimated Channel Utilization/ Transmission Duration (based on 60.16 Mbps channel rate) **	Comments
TP4	3-way handshake (does not support subchannel breakout parameter calculation)	2-way handshake	Set-up, connection update	Supports demultipexing and remultiplexing at pre-selected rates (does not allow for subchannel rate to be calculated based on network parameters)	Adaptive window	Form of Go-Back-N where NAKs are not reported (transmitter resends based on timer)	Max of 80%	- Additional header fields - All signaling is in-band which reduces data throughput
TCP	3-way handshake (does not support subchannel breakout parameter calculation)	3-way handshake	Set-up, connection update	Only supports multipexing several application channels into a single channel	Adaptive window	Form of Go-Back-N where NAKs are not reported (transmitter resends based on timer)	Max of 40% (excluding header fields) 8 min	- No demultiplexing capability - Additional header fields
VMTP	Implicit - connection established at arrival of first packet (does not support subchannel breakout parameter calculation)	Implicit	Set-up, connection update, mode selection	Only supports multipexing several application channels into a single channel	Rate control	Selective-Reject	Max of 40% (excluding header fields) 8 min	- No demultiplexing capability - Additional header fields
XTP	Implicit - connection established at arrival of first packet (does not support subchannel breakout parameter calculation)	3-way handshake	Set-up, connection update, mode selection	Only supports multipexing several application channels into a single channel	Adaptive window or rate control	Selective-Reject	Max of 40% (excluding header fields) 8 min	- No demultiplexing capability - Additional header fields

Transport Protocol	Connection Establishment	Connection Termination	Connection Signaling Parameters	Multiplexing Schemes	Flow Control	Error Control Scheme	Estimated Channel Utilization/ Transmission Duration (based on 60.16 Mbps channel rate) **	Comments
VTA specific protocol	Modified 3-way handshake (establishes subchannel breakout parameter)	3-way handshake	Set-up	Supports demultiplexing and remultiplexing based on subchannel breakout parameter	Adaptive window (multiplexing operations stops when window is exhausted)	Go-Back-N	100% utilization 3 min	Ideal protocol

** Values in this column are raw approximations based on demultiplexing capability and additional transport header fields

5.0 Conclusion

From our analysis, it is recommended that the VTA specific design to implemented as the transfer protocol to support the VTA. The VTA specific protocol is relatively simple and seems to outperform current protocols based on the VTA system requirement. The main advantages of the VTA specific protocol are as follows,

- Protocol works on a 47 byte data unit basis for demultiplexing/remultipexing (no SAR needed at AAL);
- Supports the decomposition of the overall channel rate into subchannels to maximize resource utilization;
- No transport header fields needed (communicates with ATM and management layer to coordinate SN and VCI for each subchannel - utilizes ATM header fields);
- Implicit reassembly of original file from input ports (no need for additional SN at transport layer);
- Error control at each sublayer can control demultiplexing/remultiplexing rate, hence it can provide flow control to avoid buffer overflow;
- Call set-up (modified three-way-handshake) establishes subchannel breakout parameter.

The drawbacks are,

- Requires strict connection link parameters to support CLR of 10^{-8} end-to-end;
- Will require ATM network nodes to write expected queuing delay and output link (fiber) parameters upon call set-up.

The transport protocol analysis presented in Section 4.0 was heavily dependent on processing, queuing, and propagation delays. These delays were based on current information available from various references. When implementing an actual system, these delays should be revised to reflect actual network values (e.g. total processing delay of actual ATM interface module). The analysis techniques presented in this report can then be recalculated to derive specific values needed to optimize the transport protocol design.

Appendix A Link Analysis

The design of the link (physical connections) between all nodes (users, switching offices, and video providers) is a complex process. The level of design that is considered for the network is a preliminary design of the fiber system based on the requirements set forth by the VTA. The equations and values presented in this appendix were derived from [9].

The basic requirement is that the end-to-end connection (including transmitters, repeaters, and receivers) have a CLR of 10^{-8}. This corresponds to a end-to-end BER of,

$10^{-8} = 1 - [\prod(1\text{-}BER_i)]^{376}$ (376 is the number of bits per data unit) where the product is taken over the number of fibers in the end-to-end connection. Therefore, assuming all fiber links in the connection have the same average BER, we can solve for the required BER to support the VTA requirement.

$[1 - BER]^n = .99999999997340$ or $BER_i = 1 - .9999999999734^{1/n}$

The other requirements are:

- From the error control analysis - propagation delay = 2.2×10^8
- From CCITT recommendations for SDH interfaces - bandwidth of the fibers are to be 155.55 Mbps or 622.08 Mbps

The objective in implementing a fiber system that will meet the above requirements will be to select a transmitting device, a optical fiber , a receiving device, and the number of required repeaters to support the requirements for a particular end-to-end connection. Typical specifications for these type of devices are given below (at 1.3×10^{-6}m):

Transmitter	Optical Fiber	Receiver
- Rise/fall time: t_r,t_f (ns) - Peak output power into the fiber pigtail: P_t (dBm) - Laser spectral width: w(nm)	- Reaction index: n_o - Attenuation: α (dB/km) - Dispersion: D (ps/nm-km)	- Capacitance: C (pf) - Rise/fall time: t_r,t_f (ns) - Sensitivity: R_o (A/W) - Diameter of active area (μm) - Dark current: I_d (nA)

Assume ideal repeaters in which the output power level is the same as the transmitter. Usually, repeaters will be installed at ATM switching nodes, however, if a single fiber link length is to long to support the BER rate, then a repeater may be inserted at the half-way point to reduce the BER.

Fibers (and transmit/receive equipment) required to support data rates of the two user standards: 155.52 Mbps and 622.08 Mbps, are analyzed. This same analysis method can be used for fibers requiring higher data rates (i.e. internode fibers). A bit error rate (BER) of 10^{-9} will be assumed for the analysis.

A.1 Fiber dispersion

Fiber dispersion is defined as the amount of signal spreading that will occur per unit length of fiber. The dispersion value will limit the fiber length and drive transmitter/receiver rise and fall times. The dispersion effect in a fiber can be measured in terms of the fiber transfer characteristic. An impulse function is introduced to the fiber transfer characteristic. The impulse response spectral width is a measure of the dispersion.

Dispersion causes the pulse width of a bit in the fiber to spread. The bit pulse widths are calculated from the fiber bit rates (which are related to the fiber bandwidths)

T = 1/(data rate) = 1/R	=	6.4300412 ns	(for 155.52 Mbps)
		1.6075103 ns	(for 622.08 Mbps)

The duration of an impulse response applied to the fiber provides a reasonable estimate of the pulse stretching due to fiber dispersion. A Gaussian impulse response is used to estimate the fiber transfer characteristic, with σ=the response width. Figure A-1 illustrates the impulse response concept.

The 1/2 power bandwidth of the impulse response relates to the impulse spectral width, σ.

The time and spectral representation of the impulse response to the Gaussian transfer

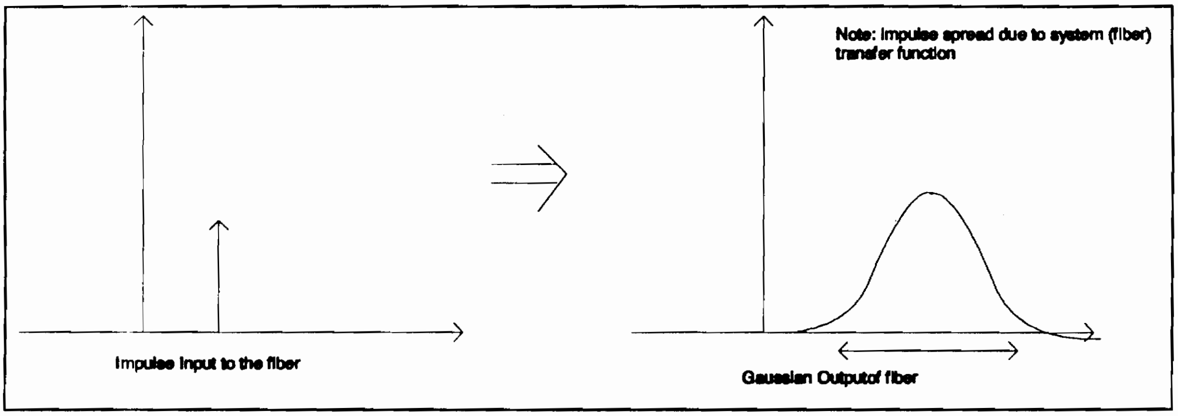

Figure A-1 Gaussian Impulse Response

characteristic is,

$$h(t) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\frac{-t^2}{2\sigma^2}$$

$$H(\omega) = \exp\frac{-\omega^2\sigma^2}{2}$$

The 3dB point is where $H(\omega)$ is reduced to,

$$\frac{1}{\sqrt{2}} \times H(0) \qquad \text{where} \qquad H(0) = \exp 0 = 1$$

Therefore,

$$H(\omega_{3dB}) = \frac{1}{\sqrt{2}} = \exp\frac{-\omega^2\sigma}{2}$$

$$.6931472 = \omega^2\sigma^2$$

The effective electrical bandwidth of the fiber as related to the impulse response spreading (σ) is 2 times the ω_{3dB} value or,

$$\frac{.8325546}{\sigma} = \omega_{3dB}$$

$$B = 2(\frac{.8325546}{\sigma}) \text{ rads/s}$$

or, in terms of frequency,

$$B = 2(\frac{.13}{\sigma}) \text{ Hz}$$

The equation above establishes a relationship between fiber data rate (or bandwidth) and end-to-end fiber dispersion (pulse spreading). Substituting the required data rates into the equation

$$\sigma = \frac{.26}{R}$$

above (assuming that the modulation scheme provides a 1 bit/Hz bandwidth efficiency), results in maximum allowable end-to-end fiber dispersion values.

Maximum allowable end-to-end dispersion:	1.67 ns	(155.52 Mbps)
	.042 ns	(622.08 Mbps)

From these maximum end-to-end dispersion values, the fiber dispersion/unit length can be derived by dividing by the fiber length. We will revisit the dispersion requirements in Section A.5.

A.2 Attenuation

The choice of wavelength for the fiber (carrier frequency) will be based on component availability and attenuation. Figure A-2 shows theoretical minimum attenuations determined by Rayleigh scattering,

Hence the optimum areas of operation are around 1.3 x 10^{-6}m and 1.5 x 10^{-6}m. High

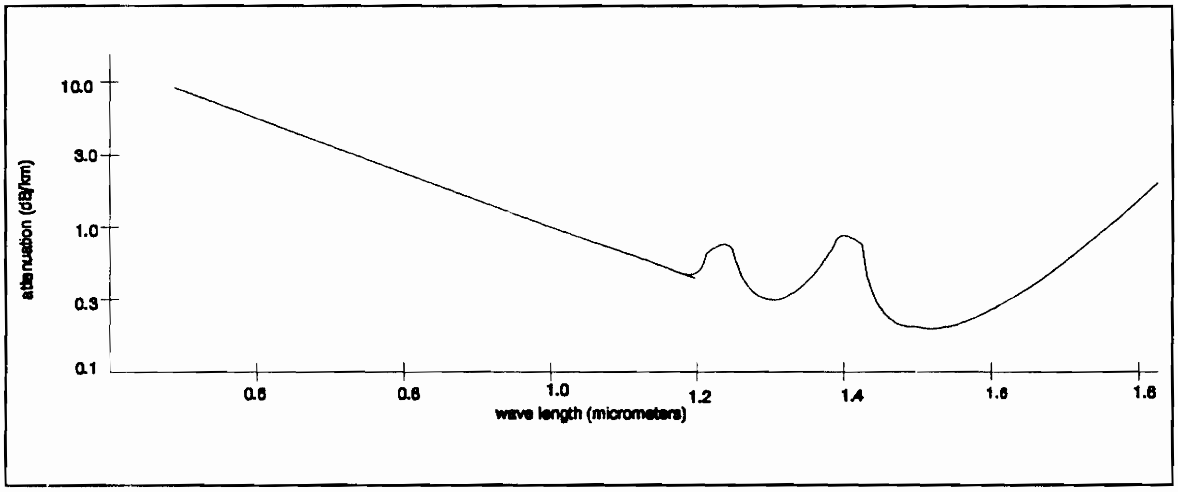

Figure A-2 Attenuation Curve

performance components are more readily available for wavelengths of 1.3 micrometers as opposed to 1.5 micrometers. Therefore, fiber analysis in this appendix will assume an operating wavelength of 1.3 micrometers. The minimum attenuation that can be achieved at this wavelength is .3 dB/km.

A.3 SNR

The required signal to noise ratio is dependent upon the fiber BER values (which are dependent upon the end-to-end CLR value required for the VTA). The signal to noise ratio will in turn determine the required transmitter power and receiver sensitivity. BER versus S/N ratios tables are given in most communications text books, Figure A-3 relates the two parameters for simple PCM [9]. From the figure, the required S/N ratio can be determined.

For example, a fiber link requiring a BER of 10^{-11} will required a S/N ratio of ~17 dB. This is based on the signal amplitude being interpreted as the amount by which the peak pulse amplitude exceeds the threshold. If the threshold is places at one-half the peak signal amplitude, the peak signal is twice the magnitude of the amplitude above the threshold. Therefore, the peak-signal to rms-noise ratio is 6 dB higher then the S/N ratio read from Figure A-3, or 23 dB. Recall that the BERs of all fibers in the end-to-end connection must obey the following relationship,

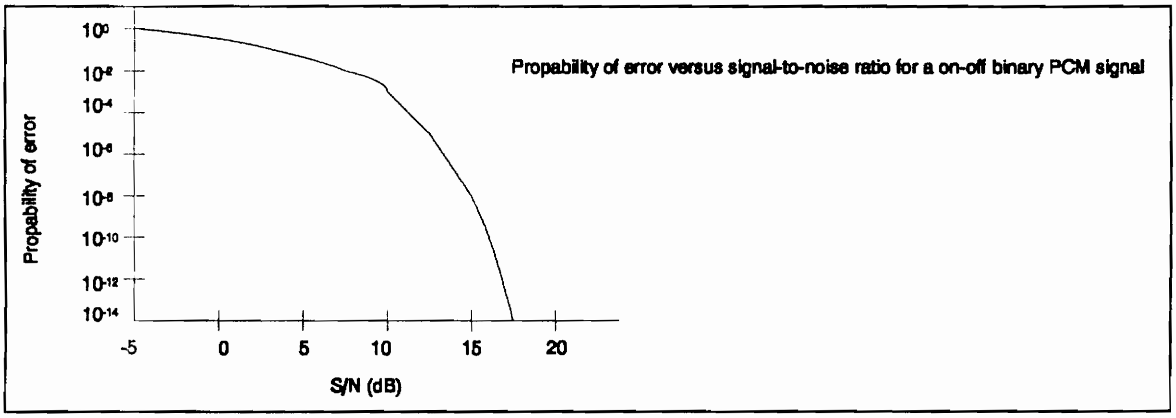

Figure A-3 BER Versus SNR from [9]

$[1\text{-BER}]^n$ = .9999999999734 product of all BERs assuming all fibers have same BER values. The average BER value per fiber is then'

BER (per fiber) = 1- $.9999999999734^{1/n}$

The following analysis develops the link budget equation on a per fiber basis (assuming each link is designed to the average BER from above.

The end-to-end S/N ratio, coupled with the required receiver operational bandwidth (this is assumed to be the same as the required fiber bandwidth), will determine the transmitting power and the receiver sensitivity. The signal component will be a summation of the received average photocurrent due to the signal power propagating in the fiber and various noise components. The main noise components include thermal noise, shot noise, and receiver component noise.

Therefore, the link budget equation can be written as,

$$\frac{S}{N} = \frac{I_{ph}^2}{(2q(I_{ph} + I_d)) + \frac{4kT}{R_f \| R} + \frac{I_a^2}{M^2} + \frac{v_a^2}{M^2}\left(\frac{1}{(R_f \| R)^2} + \frac{(2\pi \Delta f C)^2}{3}\right)\Delta f}$$

where, M is the current amplification due to the avalanche effect in APD detectors (M=1 for PIN diodes). The photodetector current, I_{ph} is the signal power in the above equation, and is related to the receiver input power by , $P_{in} = I_{ph}/R_o$, where R_o is the receiver sensitivity. Figure A-4 depicts the circuit diagram for the receiver.

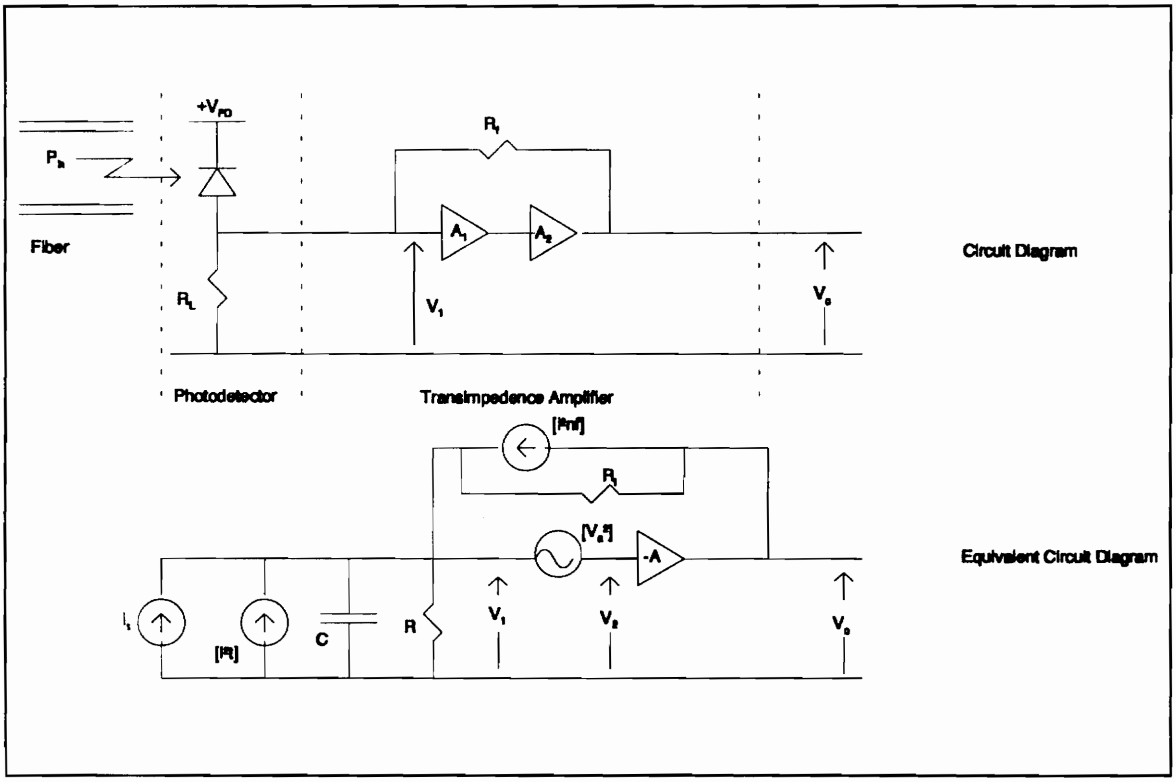

Figure A-4 Receiver Circuit Diagram

The noise components include,

$2q(I_{ph} + I_d)$ Shot noise

$$\frac{4kT}{R_f \| R}$$ Thermal noise

$$\frac{i_a^2}{M^2}$$ Mean squared output current noise at amplifier

$$\frac{v_a^2}{M^2}\left(\frac{1}{(R_f \| R)^2} + \frac{(2\pi\Delta f C)^2}{3}\right)$$ Mean squared output voltage noise at amplifier

Δf is the bandwidth of the fiber and receiver, it is assumed that a bandpass filter removes the noise components from the rest of the spectrum. For most detectors (receivers) the thermal noise dominates, therefore, an approximation for the signal to noise ratio is,

$$\frac{S}{N} \propto \frac{I_{ph}^2}{\frac{4kT\Delta f}{R_f \| R}} \quad \Rightarrow \quad I_{ph} = \sqrt{\log^{-1}\left[\left(\frac{22}{10}\right)\left(\frac{4kT\Delta f}{R_f \| R}\right)\right]}$$

where I_{ph} is the required signal current at the receiver. With the receiver signal current defined, the receiver input power required can be calculated via the following equation,

$$P_{in} = \frac{I}{R_o}$$ (R_o is the receiver sensitivity)

The power into the receiver can be derived from the transmitting power and the various link losses due to attenuation, splicing, system margin, and receiver coupling loss,

$$P_{in(receiver)}(dBm) = P_{t(transmitter)}(dBm) - \{[\ \alpha(dB/km)\ -\ splice\ loss(dB/km)] \times L(km)\}$$
$$-\ system\ margin\ (dB) - receiver\ coupling\ loss\ (dB)$$

where α=attenuation and L=fiber length

From the above relationship, the required P_t and R_o values can be derived to support each fiber link BER and the required end-to-end CLR.

A.4 Refractive Index

For a propagation velocity of $2.2x10^8$ m/s, a refractive index of,

$$n_o \geq 4.87$$

is required.

A.5 Rise and Fall Times (Dispersion Revisited)

The fiber dispersion per unit length (ps/nm-km) and transmitter laser spectral width can be derived from the maximum fiber length and total end-to-end dispersion. Recall that the maximum end-to-end dispersion was calculated to be,

1.67 ns (155.52 Mbps)
.042 ns (622.08 Mbps)

Dispersion will be a function both of fiber length and laser spectral width as it is coupled into the fiber.

dispersion/length = D (ps/nm-km) x Laser spectral width (nm) = ps/km

The total end-to-end maximum allowable dispersion includes dispersion contributed by the transmitting and receiving devices. The dispersion/unit length of fiber equation must allow for some dispersion contributions from the transmitter and receiver, hence the amount of dispersion at these devices is subtracted from the total end-to-end dispersion, therefore,

$$(\text{D x Laser spectral width})^2 = (\text{end-to-end dispersion}^2 - \text{device dispersion}^2) / \text{fiber length}$$

Laser spectral width and fiber dispersion (ps/nm-km) must be balanced to achieve the required end-to-end dispersion and fiber length. Note that if repeaters are used, the end-to-end dispersion becomes the repeater-to-repeater total dispersion since the repeaters regenerate the signal.

The allowable device dispersion drives the rise and fall times of the transmitter and receiver. The following equation relates the fiber, transmitter, and receiver dispersion components to the total end-to-end dispersion.

$$(\text{total end-to-end dispersion})^2 = \delta_{\text{transmitter}}^2 + \delta_{\text{fiber}}^2 + \delta_{\text{receiver}}^2 + \delta_{\text{receiver amplifier}}^2$$

The device dispersion must then equal,

$$(\text{total end-to-end dispersion})^2 - \delta_{\text{fiber}}^2 = \text{device dispersion} = \delta_{\text{transmitter}}^2 + \delta_{\text{receiver}}^2$$

where both the receiver and receiver amplifier dispersion term are lump into the $\delta_{\text{receiver}}^2$ term. The transmitter and receiver dispersion term are related to the rise and fall times as follows, assuming that $t_r = t_f$,

$$\delta = t_r/2$$

hence,

$$(\text{total end-to-end dispersion})^2 - \delta_{\text{fiber}}^2 = \text{device dispersion} = t_{r(\text{transmitter})}^2/4 + t_{r(\text{receiver})}^2/4$$

From the above relationships, the device rise and fall times, the fiber dispersion values, and the fiber spectral laser widths can be determined for each fiber in a particular end-to-end connection.

A.6 Fiber Link Analysis Tool

An analysis between implementation and maintenance costs versus performance requirement must be weighed in order to select the optimum devices and optical fibers. An analysis tool is developed to calculate optical transmission system parameters from connection values to meet the VTA and network requirements. The optical transmission system parameter ouputs can be used to determine if the values fall within the current design limits. Inputs to the analysis tool are number and length of fibers in the connection, operating wavelength, splice loss per fiber length, noise margin, and end-to-end dispersion. The tool calculates the required transmitter power, the required I_{PH}, the required receiver sensitivity, fiber dispersion per unit length, transmitter and receiver rise and fall times, and fiber laser spectral width for each fiber in the connection. The analysis tool is created on a Lotus 1-2-3 spreadsheet format, a sample analysis of a connection containing 3 fibers, each 15 km in length, is presented on the following page. From [7], it can be shown that fibers operating at the wavelength assumed in the sample run, and supporting data rates up to several 100s Mbps, can be designed to yield BER up to $1x10^{-11}$.

	on/off value(1 or 0)	length (km)	Bandwidth (Mbps)
fiber #1	1	15	155.52
fiber #2	1	15	155.52
fiber #3	1	15	155.52
fiber #4	0	0	0
fiber #5	0	0	0
fiber #6	0	0	0

# of links 3

Fibers operating wavelength:	1.3	0.3 (attenuation dB/km)		
Fibers end-to-end dispersion (ns):	1.67			
Fiber splice loss dB/km:	0.2		Device Dispersion (ns):	0.3
Noise margin of connection (dB):	2			

Required BER/link:	4.220E-11	
Required SNR/link:	16.9894525	BER is calculated from curve in Appendix A

Device Parameters:

Fiber link #	Iph (ampere)	Pin(max dBm)	Pin(min dBm)	Ro(min)	Ro(max)	Pt(min dBm)	Pt(max dBm)
1	1.40811E-07	-36.295154119	-38.056067	0.6	0.9	-26.5561	-24.7952
2	1.40811E-07	-36.295154119	-38.056067	0.6	0.9	-26.5561	-24.7952
3	1.40811E-07	-36.295154119	-38.056067	0.6	0.9	-26.5561	-24.7952
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0

Fiber Parameters:

Fiber link #	Fiber Dispersion (min)	Fiber Dispersion (max)	Laser width (max)	Laser width (min)	Rise/fall time for both tx & rx (ns)
1	0.0129099445	0.05163977795	4	1	0.774597
2	0.0129099445	0.05163977795	4	1	0.774597
3	0.0129099445	0.05163977795	4	1	0.774597
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0

Note: dispersion in ps/nm-km, laser width in nm

Appendix B Markov State Probability Analysis

The results presented in Table 4.5 are the basis for which the demultiplexing/remultiplexing schemes were designed. The table parameters were calculated as follows:

Number of subchannels - these parameters were established in Section 3.2, the number of subchannels required range from 20-45.

Subchannel data rate - this parameter was calculated for each subchannel breakout parameter as follows,

subchannel data rate = (total application rate)/(# of subchannels)

= 60.16 Mbps/(# of subchannels)

Inter-data unit arrival time - this parameter related to the number of subchannels. For each subchannel breakout parameter, the inter-data unit arrival time (from application to a input port) was calculated as follows,

data unit arrival rate per input port = (8 x 47)/(VTA subchannel rate)

Average round trip delay - from the Go Back N analysis, each subchannel breakout parameter subchannel data rate was solved such that the subchannel utilization was ~ 1. Hence the average round trip delay for a subchannel utilization of 1 is 3 times the inter-data unit arrival time.

Average round trip delay = 3 x (Inter-data unit arrival time)

Maximum number of switches - this parameter is calculated for each subchannel breakout parameter. From the VTA subchannel rate, parameter "a" is known.

a = 3/(VTA subchannel rate)

From "a", the number of fibers, n, and the number of switches, n-1, is recalculated assuming a minimum fiber length of 5 km. Note n and n-1 are the one-way values. To calculate the round trip numbers, these values were multiplied by 2.

$$a = \frac{2(139.38\ \mu sec + propagation + queuing\ delay)}{376}$$

$$propagation + queuing\ delay = \frac{a \times 376}{2} - 139.38\ \mu sec$$

$$propagation + queuing\ delay = \sum_{i=1}^{n} \frac{5\ km}{2.2x10^8} + \sum_{i=1}^{n-1} 5.45\ \mu sec$$

therefore,

$$\sum_{i=1}^{n} \frac{5\ km}{2.2x10^8} + \sum_{i=1}^{n-1} 5.45\ \mu sec = \frac{a \times 376}{2} - 139.38\ \mu sec$$

$$n(2.273x10^{-5}) + (n - 1)(5.45\mu sec) = \frac{a \times 376}{2} - 139.38\ \mu sec$$

hence,

$$n = \frac{(\frac{a \times 376}{2} - 139.38\ \mu sec) + 5.45\ \mu sec}{2.818x10^{-5}}$$

Multiplying by 2 for the round trip values results in:

Maximum number of 5 km fibers = 2(round(n))

Maximum number of switches = 2(round(n-1))

Maximum number of switches to inter-data unit arrival time ratio - This ratio is just

ratio = [2(round(n-1))]/(inter-data unit arrival time)

State probabilities - for each subchannel breakout parameter, the maximum number of switches (round trip) were calculated. A MATLAB routine was written which selects random queuing delays (based on the distribution in section 3.2.2) for each queue in the round trip of a particular connection. The total delay due to the switches is then summed and compared to the inter-data unit arrival time.
This is run 10^9 times to determine the probabilities of jumping states.

If the *total delay* $\leq$ *2(round(n-1))×5.45μsec*, then the data unit arrives within 3 inter-arrival time unit. This probability is referred to as p_1.

If *[2(round(n-1))×5.45μsec]* < *total delay* $\leq$ *[2(round(n-1))×5.45μsec + 1 inter-arrival time unit]*, then the data unit arrives after the 3rd inter-arrival time unit (4th data unit) but before the 4th inter-arrival time unit (5th data unit). This probability is referred to as p_2.

These same comparisons are done up to the probability of exceeding 6 time units (7 data units). The probabilities p_3, p_4, and p_5 correspond to data units arriving between the 4th to 5th time units, 5th to 6th time units, and 6th to 7th time units respectively.

This algorithm was repeated 10^9 times to derive the probabilities (sum the number of times each total delay meets probability requirement and divide by 10^9). From these probabilities, the Markov chain probabilities were calculated. The calculations are presented in Table B-1 (note - probabilities concerning 3 or more subsequent data units are insignificant and are omitted from the calculations).

Table B-1 Markov Chain Probability Derivations

Markov probabilities	Probability conditions	Calculated probability (from MATLAB simulation)
$1-L_3-U_2$	probability that a data unit is received within 3 interarrival time unit w/o error	P_1-10^{-9}
L_3	probability that a data unit is not received within 3 interarrival time units or error	$P_2+P_3+P_4+P_5+10^{-9}$
U_2	Probability that the subsequent data unit is received within 2 interarrival time units w/o error	0
$1-L_4-U_3$	probability that the subsequent data unit is not received within 3 interarrival time units or error	P_2
L_4	probability that a data unit is not received within 4 interarrival time units or error	$P_3+P_4+P_5+10^{-9}$
U_3	probability that the subsequent data unit is received within 3 interarrival time units w/o error	P_1
$1-L_5-U_4-U_{53}$	probability that the subsequent data unit is not received within 4 interarrival time units or error	P_3
L_5	probability that a data unit is not received within 5 interarrival time units or error	$P_4+P_5+10^{-9}$
U_4	probability that the subsequent data unit is received between 3 and 4 interarrival time units	P_2
$1-L_6-U_5-U_{64}-U_{63}$	probability that the subsequent data unit is not received within 5 interarrival time units or error	P_4
L_6	probability that a data unit is not received within 6 interarrival time units or error	P_5+10^{-9}
U_5	probability that the subsequent data unit is received between 4 and 5 interarrival time units	P_3
$1-U_6-U_{75}-U_{74}-U_{73}$	probability that the subsequent data unit is not received within 6 interarrival time units or error	P_5+10^{-9}
U_6	probability that the subsequent data unit is received between 5 and 6 interarrival time units	P_4
U_{35}	probability that a data unit is received within 3 interarrival time unit w/o error	P_1-10^{-9}
U_{75}	probability that the subsequent data unit is received between 4 and 5 interarrival time units	P_3
U_{73}	probability that a data unit is received within 3 interarrival time unit w/o error	P_1-10^{-9}
U_{74}	probability that the subsequent data unit is received between 3 and 4 interarrival time units	P_2
U_{64}	probability that the subsequent data unit is received between 3 and 4 interarrival time units	P_2
U_{63}	probability that a data unit is received within 3 interarrival time unit w/o error	P_1-10^{-9}

The MATLAB simulation and chain probability estimates were used to tabulate the results in Table 4.6. A printout of the MATLAB simulation is presented in the following pages.

```
ns=16;
pnd=ns*5.45;
iat=256.3;
itwoat=iat+iat;
ithrat=iat*3;
nd=0;
one=0;
two=0;
three=0;
four=0;
for x=1:1000000000
tot=0;
for y=1:ns
x=rand;
if x <= .25
   t=2.725;
end
if ((x > .25) & (x <= .75))
   t=5.45;
end
if ((x >.75) & (x <= .875))
   t=8.175;
end
if ((x > .875) & (x <= .9375))
   t=10.9;
end
if ((x >.9375) & (x <= .96875))
   t=13.625;
end
if ((x > .96875) & (x <= .984375))
   t=16.35;
end
if ((x >.984375) & (x <= .99218751))
   t=19.075;
end
if ((x >.99218751) & (x <= 1))
   t=21.8;
end
```

```
tot=tot+t;
end
if tot <= pnd
   nd=nd+1;
end
if ((tot > pnd) & (tot <=iat))
   one=one+1;
end
if ((tot > iat) & (tot <= itwoat))
   two=two+1;
end
if ((tot > itwoat) & (tot <= ithrat))
   three=three+1;
end
if (tot > ithrat)
   four=four+1;
end
end
nd=nd/1000000000
one=one/1000000000
two=two/1000000000
three=three/1000000000
four=four/1000000000
```

Appendix C References

[1] Asynchronous Transfer Mode Networks, Performance Issues, Raif O. Onvral, ARTECH House, Inc. 1994

[2] The ATM Layer Chip: An ASIC for B-ISDN Applications, Cesar A. Johnston and H Jonathan Chao, IEEE Journal on Selected Areas in Communications, Vol 9, No. 5 June 1991

[3] B-ISDN Multimedia Workstation Architecture, Shiro Sakata, IEEE Communications Magazine, August 1993

[4] Data Communications Principles, Richard U. Gitlin, Jeremiah F. Huges, Stephen B Weinstein, Plenum Press, 1992

[5] Data and Computer Communications, William Stallings, MacMillian Publishing Company, 1994

[6] Data Networks, Bertsekas/Gallager, Prentice Hall, 1992

[7] Fiber-Optic Subcarrier Multiplexing Video Transport Employing Multiple QAM, Nario Kanno and Katsuyohi Ito, IEEE Journal on Selected Areas in Communications, Vol 8, No 7, September 1990

[8] The Impact of Synchronous Digital Hierarchy on Digital Microwave Radio: A View From Australasia, Mansoor Shafi, Les Davy, Walter Smith, IEEE Communications Magazine, May 1990

[9] Optical Fiber Communication Systems, William B Jones Jr, Holt, 1988

[10] Optical Transport Networks, Tetsuya Miki, Proceedings of the IEEE, vol 81, No 11, Nov 1993

[11] Performance Evaluation of Input and Output Queuing Techniques in ATM Switching Systems, Ehrico Del Re, IEEE Transactions on communications, Vol 41, No 10, Oct 1993

[12] A SONET STS-3c User Network Interface Integrated Circuit, Thomas J Rohe, Kenneth A Walsh, IEEE Journal on Selected Areas in Communications, Vol 9, No 5, June 1991

[13] 2.488 Gb/s SONET Multiplexer/Demultiplexer with Frame Detection Capability, Dennis T Kong, IEEE Journal on Selected Areas in Communications, Vol 9, No 5, June 1991

Appendix D Acronym List

A	Amperes
AAL	ATM Adaptation Layer
ACK	Acknowledgment
APD	Avalanche Photo Detector
ARQ	Acknowledgement Request
ASK	Amplitude Shift Keying
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BER	Bit Error Rate
CBR	Constant Bit Rate
CCITT	Consultative Committe on International Telegraphy and Telephony
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CRC	Cyclic Redundancy Check
CS	Convergence Sublayer
D	Dispersion
dB	Decibel
DCT	Discrete Cosine Transform
GFC	Generic Flow Control
HEC	Header Error Check
HGA	High Gain Amplifier
hr	Hours
Hz	Hertz
LFC	Local Function Capability
LPF	Low Pass Filter
m	Meters
Mbps	Megabits per second
MC	Motion Compression
MPEG	Motion Picture Experts Group
NAK	Nonacknowledgement
OAM	Operations and Maintenance
OOK	On-Off Keying
P	Power
PCM	Pulse Code Modulation
PDU	Payload Data Unit
PLL	Phase Lock Loop
PM	Physical Medium Sublayer
PT	Payload Type
R	Data Rate
RES	Reserved
s	Seconds
S/N	Signal to Noise Ratio
SAR	Segmentation and Reassembly Sublayer
SDH	Synchronous Digital Hiearchy
SDU	Service Data Unit
sec	Second
SN	Sequence Number
SNP	Sequence Number Protection
SNR	Signal to Noise Ratio
SONET	Synchronous Optical Network
SPE	Synchronous Payload Envelope
STS	Synchronous Transport Signal
T	Time Period
TC	Transmission Control Sublayer
TE	Terminal Equipment
TV	Television
VBR	Variable Bit Rate
VC	Virtual Channel
VCI	Virtual Channel Identifier
VCR	Video Recorder
VLC	Variable Length Coder
VOD	Video on Demand
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
VTA	Video Transfer Application
W	Watts