Traffic Management in ATM Networks Over Satellite Links

Rohit Goyal, Raj Jain, Mukul Goyal, Sonia Fahmy, Bobby Vandalore
Department of Computer Information Science
2015 Neil Ave, DL395
Columbus, OH 43210
Phone: (614)-688-4482. Fax: (614)-292-2911.
Email: goyal@cis.ohio-state.edu, jain@cis.ohio-state.edu

Tom vonDeak
NASA Lewis Research Center
21000 Brookpark Road, MS 54-2
Cleveland, OH 44135
Phone: 216-433-3277 Fax: 216-433-8705
Email: tvondeak@lerc.nasa.gov

Abstract

This report presents a survey of the traffic management issues in the design and implementation of satellite-ATM networks. First a reference satellite-ATM network architecture is presented along with an overview of the service categories available in ATM networks. The error characteristics of satellite channels, techniques to improve the error characteristics, and the impact on ATM network performance are then discussed. A delay model for satellite networks and the major components of delay and delay variation are described. A survey of design options for TCP over UBR, GFR and ABR services in ATM is presented next. The main focuses is on traffic management issues. Several recommendations on the design options for efficiently carrying data services over satellite-ATM networks are presented.

Table of Contents

1

Introduction

ATM technology is expected to provide quality of service based networks that support voice, video and data applications. ATM was originally designed for fiber based terrestrial networks that exhibit low latencies and low error rates. With the widespread availability of multimedia technology, and an increasing demand for electronic connectivity across the world, satellite networks play an indispensable role in the deployment of global networks. Ka-band satellites using the gigahertz frequency spectrum reach user terminals across most of the populated world. As a result, ATM based satellite networks effectively provide real time as well as non-real time communications services to remote areas.

Satellite communications technology offers a number of advantages over traditional terrestrial point-to-point networks **[AKYL97]**. These include,

- wide geographic coverage including interconnection of "ATM islands",
- multipoint to multipoint communications facilitated by the inherent broadcasting ability of satellites,
- bandwidth on demand, or Demand Assignment Multiple Access (DAMA) capabilities, and
- an alternative to fiber optic networks for disaster recovery options.

However, satellite systems have several inherent constraints. The resources of the satellite communication network, especially the satellite and the earth station are expensive and typically have low redundancy; these must be robust and be used efficiently. Also, satellite systems can use Time Division Multiplexed (TDM) physical layer, where individual earth stations can transmit frames during fixed time slots. In this case, the cell based ATM layer must be mapped onto the frame based satellite layer. This involves the use of efficient bandwidth allocation strategies for Demand Assignment Multiple Access (DAMA) based media access techniques.

Current and proposed satellite communications networks use low earth orbit (LEO) constellations as well as geosynchronous (GEO) satellite systems. GEO satellites have a high

propagation delay but a few satellites are enough to provide connectivity across the globe. LEO satellites have lower propagation delays due to their lower altitudes, but many satellites are needed to provide global service. While LEO systems have lower propagation delay, they exhibit higher delay variation due to connection handovers and other factors related to orbital dynamics **[IQTC97]**. The effects of the propagation delays for LEO systems are further intensified by the buffering delays that could be of the order of the propagation delays especially for best effort TCP/IP traffic. The large delays in GEOs, and delay variations in LEOs, affect both real time and non-real time applications. Many real time applications are sensitive to the large delay experienced in GEO systems, as well as to the delay variation experienced in LEO systems. In an acknowledgment and timeout based congestion control mechanism (like TCP), performance is inherently related to the delay-bandwidth product of the connection. Moreover, TCP Round Trip Time (RTT) measurements are sensitive to delay variations that may cause false timeouts and retransmissions. As a result, the congestion control issues for broadband satellite networks are somewhat different from those of low latency terrestrial networks. Both interoperability, as well as performance issues between satellite and terrestrial networks must be addressed before data, voice and video services can be provided over a Satellite-ATM network.

This report provides a survey of the issues involved in designing satellite-ATM networks for transporting data traffic, especially TCP/IP traffic. The report first provides an introduction to the satellite-ATM architectural issues, and presents a reference architecture for satellite-ATM networks. The report then discusses the error characteristics of satellite channels and presents techniques to improve the error characteristics. The report then discusses the implementation and performance of TCP/IP over the UBR and ABR service categories. The focus of this report is to present the issues involved, to make recommendations in the design of satellite-ATM networks.

2 Architectural Issues

In this section we present the basic architectural issues for ATM over Satellite. A reference architecture is presented, and a summary of the various ATM service categories is given.

2.1 A Reference Architecture for Satellite-ATM Networks

Figure 1 illustrates a satellite-ATM network represented by a ground segment, a space segment, and a network control center. The ground segment consists of ATM networks that may be further connected to other legacy networks. The network control center (NCC) performs various management and resource allocation functions for the satellite media. Inter-satellite links (ISL) in the space segment provide seamless global connectivity to the satellite constellation. The network allows the transmission of ATM cells over satellite, multiplexes and demultiplexes ATM cell streams from uplinks and downlinks, and maintains the QoS objectives of the various connection types. The satellite-ATM network also includes a satellite-ATM interface device connecting the ATM network to the satellite system. The interface device transports ATM cells over the frame based satellite network, and demultiplexes ATM cells from the satellite frames. The device typically can use a DAMA technique to obtain media access to the satellite physical layer. The interface unit is also responsible for forward error correction techniques to reduce the error rates of the satellite link. The unit must maintain ATM quality of service parameters at the entrance to the satellite network. As a result, it translates the ATM QoS requirements into corresponding requirements for the satellite network. This interface is thus responsible for resource allocation, error control, and traffic control. Details about this model can be obtained from [KOTA97]. [the reference model from TIA/EIA/TSB-91 should be used]

This architectural model presents several design options for the satellite and ground network segments. These options include

- No on-board processing or switching.
- On-board processing with ground ATM switching.
- On-board processing and ATM switching.

About 53% of the planned Ka-band satellite networks propose to use on-board ATM like fast packet switching [PONZ97]. An overview of the network architectures of some of the proposed systems can be found in [WUPI94]. In a simple satellite model without on-board processing or

switching, minimal on-board buffering is required. However, if on-board processing is performed, then on-board buffering is needed to achieve the multiplexing gains provided by ATM. On-board processing can be used for resource allocation and media access control (MAC). MAC options include TDMA, FDMA, and CDMA and can use contention based, reservation based, or fixed media access control. Demand Assignment Multiple Access (DAMA) [KOTA97b] can be used with any of the MAC options. If on-board processing is not performed, DAMA must be done by the NCC. On-board DAMA decreases the response time of the media access policy by half because link access requests need not travel to the NCC on the ground any more. In addition to media access control, ABR explicit rate allocation or EFCI control, and UBR/GFR buffer management can also be performed on-board the satellite. On-board switching may be used for efficient use of the network by implementing adaptive routing/switching algorithms. Trade-offs must be made with respect to the complexity, power and weight requirements for providing on-board buffering, switching and processing features to the satellite network. In addition, on-board buffering and switching will introduce some additional delays within the space segment of the network. For fast packet or cell switched satellite networks, the switching delay is negligible compared to the propagation delay, but the buffering delay can be significant. Buffering also results in delay variations due to the bursty nature of ATM traffic.

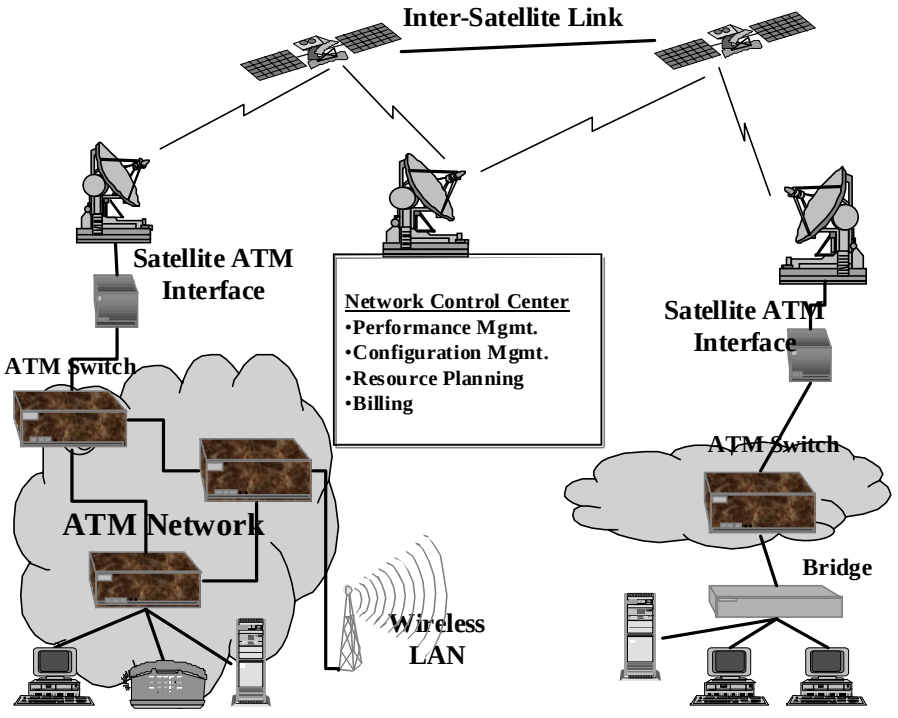


Figure 1: Satellite-ATM network model

2.2 Service Categories in ATM Networks

Satellite-ATM network architectures present tradeoffs in cost/complexity and on-board switching/processing capabilities.

Service (QoS) requirements for each service category. The ATM Forum Traffic Management Specification 4.0 [TM4096] defines five service categories for ATM networks. Each service category is defined using a traffic contract and a set of QoS parameters. The *traffic contract* is a set of parameters that specify the characteristics of the source traffic. This defines the requirements for compliant cells of the connection. The *QoS parameters* are negotiated by the source with the network, and are used to define the expected quality of service provided by the network. For each service category, the network guarantees the negotiated QoS parameters if the end system complies with the negotiated traffic contract. For non-compliant traffic, the network need not maintain the QoS objective.

The *Constant Bit Rate (CBR)* service category is defined for traffic that requires a constant amount of bandwidth, specified by a Peak Cell Rate (PCR), to be continuously available. The network guarantees that all cells emitted by the source that conform to this PCR will be transferred by the network with minimal cell loss, and within fixed bounds of cell delay and delay variation. The *real time Variable Bit Rate (VBR-rt)* class is characterized by PCR, Sustained Cell Rate (SCR) and a Maximum Burst Size (MBS) in cells that controls the bursty nature of VBR traffic. The network attempts to deliver cells within fixed bounds of cell delay and delay variation. *Non-real-time VBR* sources are also specified by PCR, SCR and MBS, but are less sensitive to delay and delay variation than the real time sources. The network does not specify any delay and delay variation parameters for the VBR-nrt service.

The *Available Bit Rate (ABR)* service category is specified by a PCR and Minimum Cell Rate (MCR) which is guaranteed by the network. The bandwidth allocated by the network to an ABR connection may vary during the life of a connection, but may not be less than MCR. ABR connections use a rate-based closed-loop feedback-control mechanism for congestion control. The network tries to maintain a low Cell Loss Ratio by changing the allowed cell rates (ACR) at

which a source can send. The *Unspecified Bit Rate (UBR)* class is intended for best effort applications, and this category does not support any service guarantees. UBR has no built in congestion control mechanisms. The UBR service manages congestion by efficient buffer management policies in the switch. A new service called Guaranteed Frame Rate (GFR) is being introduced at the ATM Forum and the ITU-T. GFR is based on UBR, but guarantees a minimum rate to connections. The service also recognizes AAL5 frames, and performs frame level dropping as opposed to cell level dropping.

In addition, the ITU-T has specified four QoS classes to be used to deliver network based QoS [I35696]. It is imperative that a broadband satellite network be able to support the various QoS services specified by the standards. Most importantly, the network should be able to support TCP/IP based data applications that constitute the bulk of Internet traffic.

Most of the parameters specified in the standards are relevant only to terrestrial networks. These values have to be re-evaluated for Satellite-ATM networks. For example, the ITU-T specifies a maximum cell transfer delay of 400 ms for the ITU Class 1 stringent service [I35696]. This class is expected to carry CBR traffic for real-time voice communications over ATM. However, the 400ms maximum delay needs to be reviewed to ensure that it properly accounts for the propagation delays in geosynchronous satellite networks. The peak-to-peak cell delay variation of QoS Class 1 is also specified to be a maximum of 3 ms by the ITU-T [I35696]. This value may be too stringent for many satellite systems. As a result, the QoS parameters are under careful consideration by ITU-4B [IT4B97]. In this context, the ITU-4B preliminary draft recommendations on transmission of Asynchronous Transfer Mode (ATM) Traffic via Satellite is in the process of development.

[t may be us

ATM provides a variety of service categories for real-time and non-real time communications.

3 Satellite Channel Error Characteristics

[It was suggested by E. Cuevas/ATT that Section 3 should be removed or greatly abbreviated and a reference given to a pending ITU-R document that Enrique is working on.]

For a satellite channel, there are two general performance requirements:

- High Throughput. To obtain good throughput, there is a need to minimize data retransmission. This is especially important if Go-back-N window based flow-control is used. When the propagation delay is large, data that is transmitted after the missing segment and before the retransmission request reaches the source, maybe lost.

- Low Cost: To reduce the cost of ground station, the power required to transmit data should be minimized while maintaining required signal-to-noise ratio at the receiver.

Inherently, satellite channels produce random single-bit errors in the data being transmitted. The Bit Error Rate (BER) depends on the Signal-to-Noise ratio at the receiver. Thus for an acceptable level of error rate, a certain minimum signal-to-noise ratio must be ensured at the receiver and hence maintained at the transmitter.

Forward Error Correction (FEC) techniques provide a solution that satisfies both these requirements. These techniques introduce some redundancy in the transmitted data. When the receiver gets the corrupted data, it uses this redundancy to decide if received data is corrupted and find out what must have been the original data. FEC codes can broadly be classified as block codes and tree codes. Block codes are 'memory-less' codes that map ‘k’ input binary signals to ‘n’ output binary signals, where ‘n’ > ‘k’ for redundancy. Tree codes, on the other hand, use 'memory' by remembering ‘v’ input signals immediately preceding the target block of ‘k’ input signals. These ‘v’ + ‘k’ input binary signals are used in the generation of ‘n’ output binary signals corresponding to ‘k’ input signals.

Convolutional coding, a subset of tree codes, and Viterbi decoding are the most popular FEC techniques used on satellite channels **[**CLAR81**]**. Thus, when the transmitted signal is FEC coded, the receiver during decoding is able to decide in most cases if the signal has been corrupted during transmission & in some cases the receiver is able to correct the corrupted signal. Thus, the receiver makes requests for data retransmission only when it detects loss of data or when data is so much corrupted that receiver can not correct it. Since receiver can tolerate a

certain level of errors in the received data, the required signal-to-noise ratio at the receiver reduces. Thus less power is required for transmission.

The reduction in required signal-to-noise ratio at the transmitter to maintain an acceptable BER can also be viewed as the reduction in satellite channel's BER for a given signal-to-noise ratio at the transmitter. Thus, use of FEC coding reduces the BER of the satellite channel for a given signal-to-noise ratio at the receiver.

However, whenever the receiver commits a mistake in detecting corrupted data or in deciding what must have been the original data before corruption, a whole bunch of successive bits are affected i.e. a 'burst' of errors occurs. Thus, the original random error nature of satellite channels gets transformed to one with bursty errors. This change from random error environment to bursty error environment for satellite channels profoundly affects the operation of ATM and AAL protocols and their transport over SDH/PDH frames as described in next 3 subsections.

Forward Error Correction coding reducess the Bit Error Rate of Satellite links but makes the errors bursty.

The most important impact of bursty errors on the functioning of ATM layer is the dramatic increase in the Cell Loss Ratio (CLR). The 8 bit ATM Header Error Control (HEC) field in the ATM cell header can correct only single bit errors. However, in a bursty error environment, if a burst of errors hits a cell header, it is likely that it will corrupt more than a single bit. Thus HEC field becomes ineffective in front of bursty errors & CLR rises dramatically.

It has been shown by a simplified analysis and confirmed by actual experiments that for random errors, CLR is proportional to square of bit error rate (BER) and for bursty errors, CLR is linearly related to BER. Now BER is very less than 1 in magnitude. Hence, for the same BER, in case of bursty errors, CLR value (proportional to BER) is orders of magnitude higher than CLR value for random errors (proportional to square of BER. Also, since for bursty errors, CLR is linearly related to BER, the reduction in CLR with reduction in BER is not as steep as in the case of channels with random errors (where CLR is proportional to square of BER). Finally, for bursty errors, the CLR increases with decreasing average burst length. This is because for the same

number of total bit errors, shorter error bursts mean that a larger number of cells are affected **[**AGNE95**][**RAMS95**]**.

Another negligible but interesting problem is that of misinserted cells. Since 8 HEC bits in the ATM cell header are determined by 32 other bits in the header, there are only 2^{32} valid ATM header patterns out of 2^{40} possibilities (for 40 ATM header bits). Thus for a cell header, hit by a burst of errors, there is a $2^{32}/2^{40}$ chance that corrupted header is a valid one. Moreover, if the corrupted header differs from a valid header by only a single bit, HEC will 'correct' that bit & accept the header as a valid one. Thus for every valid header bit pattern (out of 2^{32} possibilities), there are 40 other patterns (obtained by inverting one bit out of 40) that can be 'corrected'. The possibility that our 'error burst' hit header is one of these patterns is $40\times2^{32}/2^{40}$. Thus overall, there is a $41\times2^{32}/2^{40}$ (= 41/256 ≈ 1/6) chance that a random bit pattern, emerging after an ATM cell header is hit by a burst of errors, will be taken as a valid header. In that case a cell, that should have been discarded, is accepted as a valid cell. Such a cell is called a 'misinserted' cell. Also, the probability P_{mi} that a cell will be misinserted in a channel with bursty errors is around 1/6th of the cell loss ratio on the channel, i.e.,

$$P_{mi} \approx (1/6\ ) \times CLR$$

Since CLR can be written as a constant times BER, the misinserted cell probability is also a constant times BER, i.e.,

$$P_{mi} = k \times BER$$

The cell insertion rate, C_{ir}, the rate at which cells are inserted in a connection, is obtained by multiplying this probability by the number of ATM cells transmitted per second (r), divided by total possible number of ATM connections (2^{24}), i.e.,

$$C_{ir} = (k \times BER \times r)/2^{24}$$

Because of very large number of total possible ATM connections, the cell insertion rate is negligible (about one inserted cell per month) even for high BER ($\approx 10^{-4}$) & data rates (≈34 Mbps) **[**RAMS95**]**.

A transition for random errors to bursty errors causes the ATM Cell Loss Ratio metric to rise significantly.

tible to error bursts in the same way as ATM HEC code. A burst of errors that passes undetected through these codes may cause failure of protocol's mechanism or corruption in data. AAL type 1's segmentation and reassembly (SAR) header consists of 4 bits of Sequence Number (SN) protected by a 3 bit CRC code & a single bit parity check. There is a 15/255 chance that an error burst on the header will not be detected by the CRC code & parity check. Such an undetected error at the SAR layer may lead to synchronization failure at the receiver's convergence sublayer. AAL 3/4 uses a 10-bit CRC at the SAR level. Here, bursty errors & scrambling on the satellite channel increases the probability of undetected error. However, full byte interleaving of ATM cell payload can reduce undetected error rate by several orders of magnitude by distributing the burst error into two AAL 3/4 payloads. The price to be paid for distributing burst error into two AAL payloads is doubling of the detected error rate and AAL 3/4 payload discard rate. AAL type 5 uses a 32-bit CRC code that detects all burst errors of length 32 or less. For longer bursts, the error detection capability of this code is much stronger than that of AAL 3/4 CRC. Moreover, it uses a length check field, which finds out loss or gain of cells in an AAL 5 payload, even when CRC code fails to detect it. Hence it is unlikely that a burst error in AAL 5 payload would go undetected **[**CHIT94**]**.

3.3 Impact of Bursty Errors on Physical Layer Protocols

ATM AAL 1 and 3/4 are susceptible to bursty errors.

AAL 5 is robust against bursty errors.

ps) and

DS-3 (44.736) PDH Performance in Bursty Error Channels

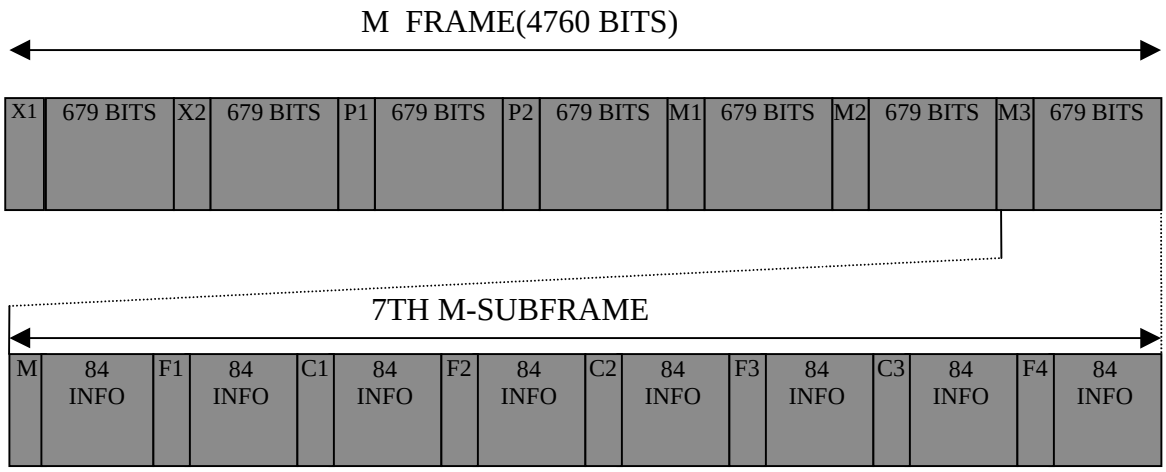


Figure 2 DS3 Signal Format [CHIT94]

The DS-3 asynchronous signal, shown in Figure 2, is defined in the ANSI T1.107 and in Bellcore TR-TSY-000499 **[**LUNS95**]**. The DS-3 signal is partitioned into 4760 bit long multi-frames (M-frames). Every M-frame is divided into seven 680 bit long M-subframes. Each M-subframe is further divided into 8 blocks of 85 bits each. 84 of these 85 bits are available to carry data.

DS-3 signal's framing mechanism is robust against burst of errors as framing bits (F1-F4 in an M-subframe) are well separated by 168 information bits & a C bit. Moreover, two multi-frame bits are separated by 679 bits. An Out-Of-Frame (OOF) state is declared in DS-3 signal when out of 8 (or 16) consecutive framing bits, 3 bits are found in error or when in 3 out of 4 consecutive M-frames, one or more multi-frame bit errors are detected. It is unlikely that a single burst of errors will effect more than one or two framing or control bits **[**LUNS95**]**.

However, the parity checking mechanism of DS-3 signal may not accurately indicate the number of errors for a bursty error channel **[**CHIT94**]**.

Performance of PLCP Over DS-3 PDH Over Bursty Error Channels

A Physical Layer Convergence Protocol (PLCP) has been defined in IEEEP802.6, Bellcore TR-TSV-000773 and the ATM Forum UNI Specification Version 3.0 for mapping ATM cells in DS-3 signal. The PLCP frame, shown in Figure 3, is 125 μs long and consists of 12 ATM cells. Each ATM cell is preceded by 4 bytes of PLCP overhead. Bytes A1 and A2, preceding every ATM

cell, are fixed framing bytes having values F4H and 28H respectively. Among other bytes, path overhead (POH) bytes Z1-Z6 are defined by the user. In order to maintain a nominal frame interval of 125 μs, the last ATM cell is followed by a 13 or 14 nibble long trailer. The C1 byte in the overhead of 12th cell indicates whether 13 or 14 nibbles are included in the trailer.

PLCP Framing		POI	POH	PLCP payload	
A1	A2	P11	Z6	First ATM Cell	
A1	A2	P10	Z5	ATM Cell	
A1	A2	P09	Z4	ATM Cell	
A1	A2	P08	Z3	ATM Cell	
A1	A2	P07	Z2	ATM Cell	
A1	A2	P06	Z1	ATM Cell	
A1	A2	P05	X	ATM Cell	
A1	A2	P04	B1	ATM Cell	
A1	A2	P03	G1	ATM Cell	
A1	A2	P02	X	ATM Cell	
A1	A2	P01	X	ATM Cell	
A1	A2	P00	C1	12th ATM Cell	Trailer
1 byte	1 byte	1 byte	1 byte	53 bytes	13 or 14 nibbles

A1, A2 = PLCP framing bytes
POI = Path Overhead Indicator
POH = Path Overhead
Z1 – Z6 = Reserved Byte
X = Unassigned Byte
B1 = BIP-8 Byte
G1 = PLCP Path Status Byte
C1 = Cycle/Stuff Counter Byte

Figure 3 PLCP Frame Format for ATM [CHIT94]

An Out-Of-Frame (OOF) state is entered whenever an error burst causes consecutive A1 and A2 bytes to be in error. The in-frame state is not reentered until two valid and consecutive A1 and

A2 byte pairs with valid POI bytes are found. For every transition to OOF state, depending on the PLCP implementation and relative location of error in the frame, 2 to 14 ATM cells will be lost.

A burst of errors with a weight of 4 or more can corrupt C1 byte beyond repair. This will cause error in deciding whether trailer is 13 or 14 nibbles long, ultimately leading to framing errors.

Finally, PLCP calculates an 8-bit interleaved parity code and stores it in B1 byte. The first bit of the code provides even parity over an array made by the 1st bit of all the bytes in the 12×54 structure (12 rows consisting of an ATM cell plus one POH byte) and so on. This mechanism works well only for random single bit errors. In the presence of a burst of errors that corrupts an even number of bits in the parity count, the B1 byte, though it will likely detect the burst of errors, won't give the accurate information about the number of errors.

Thus, it is clear that PLCP format does not function well in a bursty error environment **[**CHIT94**]**.

Performance of E-3 (34.368 Mbps) PDH over bursty error channels

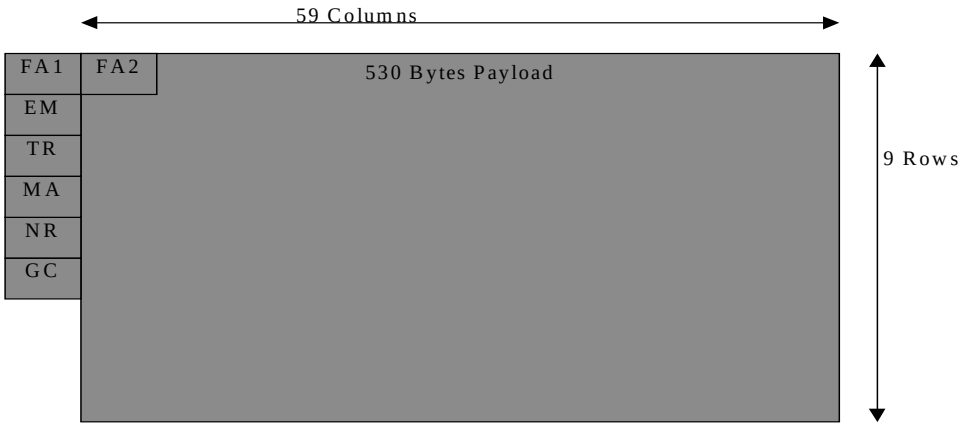


Figure 4: 34 Mbps PDH Frams as specified in ITU-T G.832 [AGNE95]

[AGNE95**]** reports field trials done to check the performance of ATM transmission over E-3 (34.368Mbps) PDH frames. In the E-3 frames, Error-Monitoring(EM) byte stores a Bit Interleaved Parity 8 (BIP-8) code over all 537 bytes in the frame i.e. the first bit of the EM byte ensures an even parity over the array made by the first bits of all 537 bytes in the frame and so on. The other overhead bytes store timing & management information. One or more parity errors in the EM byte makes the frame an Errored Block (EB). Also, BER_{BIP-8}, calculated as the number

of wrong bits in the EM byte divided by the total number of bits in the frame, gives an approximate indication of BER.

The error performance parameters defined in ITU-T G.826 – Errored Second Ratio (ESR), Severely Errored Second Ration (SESR) and the Background Block Error Ratio (BBER), are measured by processing of Errored Blocks over a measurement period. In the field trials, for measuring these parameters, the test equipment detected Errored Blocks by a byte-by-byte comparison of generated and received E-3 frames.

A burst of errors may affect more than a single bit in the array made by the corresponding bits of 537 bytes in the frame. Thus, some of the bits of the BIP-8 code stored in EM byte may give a wrong indication. Therefore, in the tested bursty error environment, BER_{BIP-8} was not same as actual BER. The conformance between BER_{BIP-8} and actual BER increased with decreasing BER. This is because with decreasing BER, average length of error burst and the number of errors per burst decreases, thereby decreasing the chance of multiple errors affecting the same parity bit. However, there was little discrepancy between the values of G.826 parameters calculated by the test equipment and the values calculated based on Errored Blocks detected by errors in the EM byte. This is because it is highly unlikely that an error burst would cause errors in all the 8 parity bits of the BIP-8 code stored in the EM byte. Thus, if a burst of errors strikes an E-3 PDH frame, atleast one bit of the BIP-8 code will be in error and the frame will be declared an Errored Block. Therefore values of G.826 parameters won't be affected.

E-3 PDH frame seems to be robust against frame & cell synchronization errors in a bursty error environment as none of these errors were noticed during the field trials both for nominal & degraded conditions **[**AGNE95**]**.

DS3 and E3 signal formats are robust against bursty errors.

PLCP over DS-3 does not function well in the presence of bursty errors.

satellite channels adversely affect the performance of physical, ATM and AAL protocols. Currently, two popular methods used to get around the problem of bursty errors are:

- Use of an 'outer' Reed-Solomon (RS) coding/decoding in concatenation with 'inner' convolutional coding/viterbi decoding. 'Outer' RS coding/decoding will perform the function of correcting error bursts resulting from 'inner' coding/decoding. RS codes consume little extra bandwidth (eg. 9% at 2Mbps) **[**CUE95a**]**. In December 1992, INTELSAT approved the use of RS codes as an optional feature **[**IESS308**]**. Section 3.5 Performance Studies of Reed-Solomon codes discusses some of the tests and field trials conducted to test the performance of Reed Solomon codes.

- CRC codes used in ATM and AAL layer headers are able to correct single bit errors. Thus, if the bits of N headers are 'interleaved' before encoding and 'deinterleaved' after decoding, the burst of errors will get 'spread' over N headers such that two consecutive headers emerging after deinterleaving will most probably never have more than a single bit in error. Now, CRC code will be able to correct single bit errors and going by dual mode of operation, no cell/AAL PDU will be discarded. Interleaving involves re-shuffling of bits on the channel and there is no overhead involved. However, process of interleaving and deinterleaving requires additional memory and introduces delay at both sender and receiver. Section 3.6

  COMSAT's ATM Link Enhancement (ALE) technique discusses the basic ideas behind interleaving scheme ALE (ATM Link Enhancement) developed by COMSAT and how it addresses the burst error problems of ATM, AAL and physical layer protocols.

3. ***Bursty errors can be mitigated either by additional encoding (like Reed-Solomon) or by using "inerleaving" techniques.***

T g/Viterbi decoding, can improve the performance of ATM over satellite links significantly. In 1993, elaborate field trials were done by AT&T, in cooperation with GUATEL. These trials suggested that, at C-band, a 2.048 Mbps Intermediate Data Rate (IDR) link with RS coding/decoding is expected to operate at a BER lower than 10^{-9} for 99.96% of the year even at locations with high rain precipitation **[**CUE95a**]**. This conclusion took in to consideration the wide range of carrier-to-noise values assigned to IDR links by INTELSAT. Moreover, the study showed that the performance improvement with the use of RS codes is maximum when IDR link BER is between

10^{-3} & 10^{-8}. For IDR links with BER less than 10^{-8}, the improvement in performance with the use of RS codes is difficult to quantify.

Table 1 Computed G.826 Parameters for 45 Mbps Links based on Predicted Link BER Performance [**CUE95b**]

Parameter	**G.826 Objective**	**IDR Computed**	**IDR+RS Computed**
ESR	2.62×10^{-2}	1.62×10^{-2}	1.91×10^{-4}
SESR	0.07×10^{-2}	4.32×10^{-4}	3.95×10^{-6}
BBER	7.00×10^{-5}	8.90×10^{-6}	3.82×10^{-7}

As shown in Table 1, for 45 Mbps IDR satellite links, the predicted performance using RS codes substantially betters predicted performance without RS coding & easily meets the performance objectives set for 45 Mbps satellite links by G.826 **[**CUE95b**]**.

Additional Reed-Solomon encoding/decoding substantially improves error performance of satellite channels.

n a TIA document to directly refer to a Company or commercial product. This section should either be stricken or reworded.]

To take care of the problems created by bursty error environment, in the functioning of ATM and AAL type 1 and 3/4 protocols and their transport over DS-3 PDH using Physical layer Convergence Protocol (PLCP), COMSAT has developed an interleaving based scheme called ATM Link Enhancement (ALE) **[**LUNS95**][**CHIT94**]**. Since ALE, a selective interleaving technique, does not introduce any overhead in terms of additional synchronization octets, it can be transparently introduced into the satellite transmission link.

ALE has been tested both in laboratory and on actual satellite links and has been shown to restore 'random error' nature of satellite links. The tests were conducted for BER values greater than 10^{-5}. Further testing needs to be done to confirm the expected performance gains for BER values less than 10^{-5} **[**LUNS95**]**.

ALE allows header interleaving to be optional. Header interleaving is done over a frame of F cells (called 'Interleaver Frame Size') and is independent of payload interleaving. To take care of ATM's correction/detection mode, for every cell involved in header interleaving, adjacent N-1 cells are skipped over. N varies between 1 and 12. The interleaver frame size, F, is related to N by $F = N \times 40$.

For making header of a participating cell, one bit is taken from headers of each of 40 participating cells.

As described before, AAL type 5 payload has a very strong CRC code. Hence the probability of any burst error going undetected is very low. However for payloads of AAL types 1 and 3/4, CRC codes are not that strong and it is possible that a burst of error will go undetected, causing problems in the functioning of the protocol.

For AAL type 1 payload, if the first byte (SAR header), containing sequence number (SN) field and the code to protect it, is bit-interleaved like cell header, the deinterleaved byte is unlikely to have more than a single error which can be corrected by the SN protection code. Thus when ALE has F ('interleaver frame size') cells in store, it performs full bit-interleaving of the first byte of the AAL type 1 payload over blocks of 8 cells. This interleaving function is independent of the interleaving performed for cell headers.

For AAL type 3/4, byte interleaving is performed on all 48 bytes of the payload. Once ALE has F cells in store, it performs full-byte interleaving of AAL type 3/4 payload over a block of K cells. For every interleaved cell, L bytes are read from each of the K cells in the block. Thus $L \times K$ should be equal to 48. It is ensured that F is a multiple of 48 so that all interleaving remains with in a frame of F cells. Cell payload interleaving in ALE is optional.

One of the problems with the PLCP in the bursty error environment is possible corruption beyond correction of C1 byte (Figure 3). Corruption of C1 byte may result in the incorrect determination of the number of nibbles in the trailer of the PLCP frame. This, in turn , results in nibble misalignment at the beginning of the next frame interval and ultimate loss of frame synchronization of the PLCP. The problem has been eliminated in the ALE through the use of user-definable growth octets (Z1-Z6). On the uplink side of the ALE, the C1 octet is delayed by 1 PLCP frame. This C1 octet is then inserted in bytes Z1 through Z4 as well as the C1 byte of the following PLCP frame. On receiver's side, a preprocessor extracts the C1 byte for a PLCP frame from the Z1-Z4 & C1 bytes of the next frame and restores it **[**LUNS95**]**.

Since all the interleaving is done within a frame of F cells, the deinterleaver needs to know when does the interleaver frame begin so that it can correctly deinterleave the data. The ALE uses Z5 and Z6 bytes of the PLCP frame to denote the boundary of the interleaver frame. The interleaver inserts an all 1's pattern in Z5 and Z6 bytes of the PLCP frame immediately preceding the start of the next interleaver frame. Z5 and Z6 bytes normally contain all zeros **[**LUNS95**]**.

4 Satellite Delay Characteristics

COMSAT's ATM Link Enhancement technique reconverts bursty errors to random errors on the satellite channel.

ents of can be large. For LEO systems, delay variations can be high.

4.1 Delay Requirements of Applications

We briefly discuss the basic qualitative requirements of three classes of applications, interactive voice/video, non-interactive voice/video and TCP/IP file transfer. Interactive voice requires very low delay (ITU-T specifies a delay of less than 400 ms to mitigate echo effects) and delay variation (up to 3 ms specified by ITU-T). GEO systems have a high propagation delay of at least 250 ms from ground terminal to ground terminal. If two GEO hops are involved, then the inter-satellite link delay could be about 240 ms. [perhaps the previous sentence means to say two LEO hops because the math doesn't wok out for two GEO hops in any case its not clear what the architecture is between the measured-delay endpoints.] Other delay components are additionally

incurred, and the total end-to-end delay can be higher than 400 ms. Although the propagation and inter-satellite link delays of LEOs are lower, LEO systems exhibit high delay variation due to connection handovers, satellite and orbital dynamics, and adaptive routing. This is further discussed in section 5.3. Non-interactive voice/video applications are real-time applications whose delay requirements are not as stringent as their interactive counterparts. However, these applications also have stringent jitter requirements. As a result, the jitter characteristics of GEO and LEO systems must be carefully studied before they can service real time voice-video applications.

The performance of TCP/IP file transfer applications is throughput dependent and has very loose delay requirements. As a result, both GEOs and LEOs with sufficient throughput can meet the delay requirements of file transfer applications. It is often misconstrued that TCP is throughput limited over GEOs due to the default TCP window size of 64K bytes. The TCP large windows option allows the TCP window to increase beyond 64K bytes and results in the usage of the available capacity even in high bandwidth GEO systems. The efficiency of TCP over GEO systems can be low because the TCP window based flow control mechanism takes several round trips to fully utilize the available capacity. The large round trip time in GEOs results in capacity being wasted during the ramp-up phase. To counter this, the TCP spoof protocol is being designed that splits the TCP control loop into several segments. However this protocol is currently incompatible with end-to-end IP security protocols. Several other mechanisms are being developed to mitigate latency effects over GEOs [TCPS98].

The TCP congestion control algorithm inherently relies on round trip time (RTT) estimates to recover from congestion losses. The TCP RTT estimation algorithm is sensitive to sudden changes in delays as may be experienced in LEO constellations. This may result in false timeouts and retransmits at the TCP layer. More sophisticated RTT measurement techniques are being developed for TCP to counter the effects of delay jitter in LEO systems [TCPS98].

4.2 Satellite Network Delay Model

In this section, we develop a simple delay model of a satellite network. This model can be used to estimate the end-to-end delay of both GEO and LEO satellite networks.

The end-to-end delay (D) experienced by a data packet traversing the satellite network is the sum of the transmission delay (t_t), the uplink (t_{up}) and downlink (t_{down}) ground segment to satellite propagation delays, the inter-satellite link delay (t_i), the on-board switching and processing delay (t_s) and the buffering delay (t_q). The inter-satellite, on-board switching, processing and buffering delays are cumulative over the path traversed by a connection. In this model, we only consider the satellite component of the delay. The total delay experienced by a packet is the sum of the delays of the satellite and the terrestrial networks. This model does not incorporate the delay variation experienced by the cells of a connection. The delay variation is caused by orbital dynamics, buffering, adaptive routing (in LEOs) and on-board processing. Quantitative analysis of delay jitter in satellite systems is beyond the scope of this study. The end-to-end delay (D) is given by:

$$D = t_t + t_{up} + t_i + t_{down} + t_s + t_q$$

Transmission delay: The transmission delay (t_t) is the time taken to transmit a single data packet at the network data rate.

$$t_t = \frac{packet\_size}{data\_rate}$$

For broadband networks with high data rates, the transmission delays are negligible in comparison to the satellite propagation delays. For example, a 9180 byte TCP packet is transmitted in about 472 microseconds. This delay is much less than the propagation delays in satellites.

Propagation delay: The propagation delay for the cells of a connection is the sum of the following three quantities:

- The source ground terminal to source satellite propagation delay (t_{up})
- The Inter-satellite link propagation delays (t_i)
- The destination satellite to destination ground terminal propagation delay (t_{down})

The *uplink and downlink satellite-ground terminal propagation delays* (t_{up} and t_{down} respectively) represent the time taken for the signal to travel from the source ground terminal to the first satellite in the network (t_{up}), and the time for the signal to reach the destination ground terminal from the last satellite in the network (t_{down}).

$$t_{up} = \frac{source\_satellite\_dist}{speed\_of\_signal}$$

$$t_{down} = \frac{dest\_satellite\_dist}{speed\_of\_signal}$$

The *inter-satellite link delay* (t_i) is the sum of the propagation delays of the inter-satellite links (ISLs) traversed by the connection. Inter-satellite links (crosslinks) may be *in-plane* or *cross-plane* links. In-plane links connect satellites within the same orbit plane, while cross-plane links connect satellites in different orbit planes. In GEO systems, ISL delays can be assumed to be constant over a connection's lifetime because GEO satellites are almost stationary over a given point on the earth, and with respect to one another. In LEO constellations, the ISL delays depend on the orbital radius, the number of satellites-per-orbit, and the inter-orbital distance (or the number of orbits). Also, the ISL delays change over the life of a connection due to satellite movement and adaptive routing techniques in LEOs. As a result, LEO systems can exhibit a high variation in ISL delay.

$$t_i = \frac{\sum ISL\_lengths}{speed\_of\_signal}$$

Buffering delay: Buffering delay (t_q) is the sum of the delays that occur at each hop in the network due to cell queuing. Cells may be queued due to the bursty nature of traffic, congestion at the queuing points (earth stations and satellites), or due to media access control delays. Buffering delays depend on the congestion level, queuing and scheduling policies, connection priority and ATM service category. CBR and real time VBR connections suffer minimum buffering delays because they receive higher priority than the non-real time connections. Cells from ABR and UBR connections could suffer significant delay at each satellite hop during periods of congestion.

Switching and processing delays: The data packets may incur additional delays (t_s) at each satellite hop depending on the amount of on-board switching and processing. For high data rate networks with packet/cell switching, switching and processing delays are negligible compared to the propagation delays.

The delay experienced by satellite connections is the sum of the transmission delays, propagation delays, buffering delays, switching and processing delays.

ks, the delay variation in LEOs can be significant. The delay variation in LEO systems can arise from several factors:

Handovers: The revolution of the satellites within their orbits causes them to change position with respect to the ground terminals. As a result, the ground terminal must handover the connections from the satellite descending below the horizon to the satellite ascending from the opposing horizon. Based on the velocity, altitude and the coverage of the satellites, it is estimated that call handovers can occur on an average of every 8 to 11 minutes [IQTC97]. The handover procedure requires a state transfer from one satellite to the next, and will result in a change in the delay characteristic of the connection at least for a short time interval. If the satellites across the seam of the constellation are communicating via crosslinks, the handover rate is much more frequent because the satellites are travelling in opposite directions.

Satellite Motion: Not only do the satellites move with respect to the ground terminal, they also move relative to each other. When satellites in adjacent orbits cross each other at the poles, they are now traveling in opposite sides of each other. As a result, calls may have to be rerouted accordingly resulting in further changes in delays.

Buffering and Processing: A typical connection over a LEO system might pass through several satellites, suffering buffering and processing delays at each hop. For CBR traffic, the buffering delays are small, but for bursty traffic over real time VBR (used by video applications), the cumulative effects of the delays and delay variations could be large depending on the burstiness and the amount of overbooking in the network.

Adaptive Routing: Due to the satellite orbital dynamics and the changing delays, most LEO systems are expected to use some form of adaptive routing to provide end-to-end connectivity. Adaptive routing inherently introduces complexity and delay variation. In addition, adaptive routing may result in packet reordering. These out of order packets will have to be buffered at the edge of the network resulting in further delay and jitter.

GEO systems exhibit relatively stable delay characteristics because they are almost stationary with respect to the ground terminals. Connection handovers are rare in GEO systems and are mainly due to fault recovery reasons. As a result, there is a clear trade-off between delay and jitter characteristics of GEO and LEO systems, especially for interactive real-time applications.

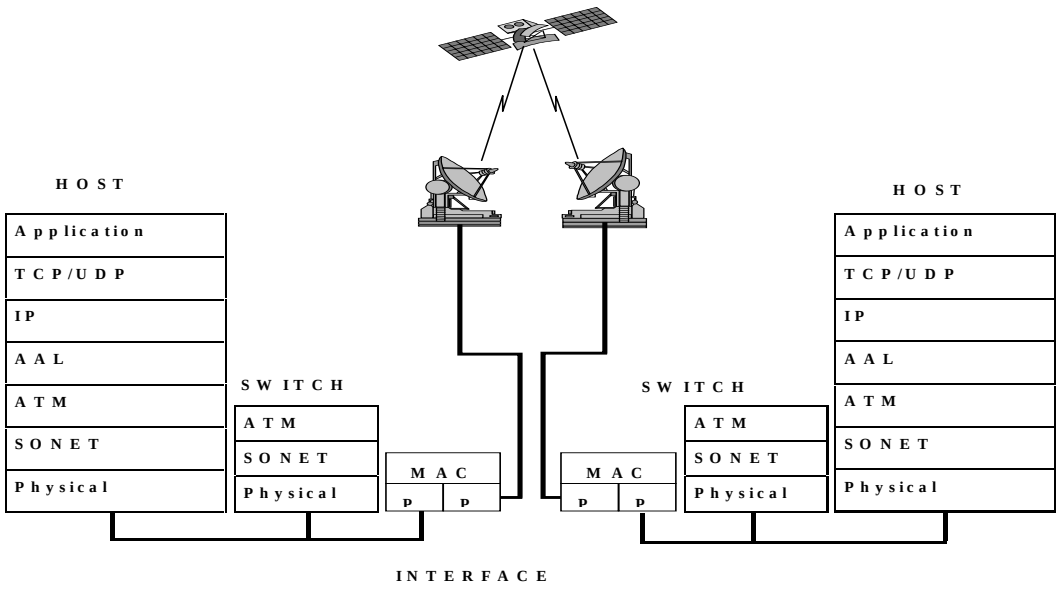
5 Media Access Protocols for ATM over Satellite

GEO systems have higher delay than LEO systems.

LEO systems can have high delay variation due to frequent handovers, satellite orbital motion, multi-hop buffering and processing, and adaptive routing.

Both interoperability issues, as well as performance issues need to be addressed before a transport layer protocol like TCP can satisfactorily work over long latency satellite-ATM networks. A crucial issue in satellite networking is that of the high end-to-end propagation delay of satellite connections. With an acknowledgment and timeout based congestion control mechanism (like TCP's), performance is inherently related to the delay-bandwidth product of the connection. As a result, the congestion control issues for broadband satellite networks are somewhat different from those of low latency terrestrial networks.

Figure \ref{satprot} illustrates the protocol stack for Internet protocols over satellite-ATM. The satellite-ATM interface device separates the existing SONET and Physical Layer Convergence Protocol (PLCP) [AKYL97][KOTA97].



The performance optimization problem can be analyzed from two perspectives -- network architectures and end-system architectures. The network can implement a variety of mechanisms to optimize resource utilization, fairness and higher layer throughput. For ATM, these include enhancements like feedback control, intelligent drop policies to improve utilization, per-VC buffer management to improve fairness, and even minimum throughput guarantees to the higher layers [GOYAL98b]. At the end system, the transport layer can implement various congestion avoidance and control policies to improve its performance and to protect against congestion collapse. Several transport layer congestion control mechanisms have been proposed and implemented. The mechanisms implemented in TCP are slow start and congestion avoidance [JACOBS88], fast retransmit and recovery, and selective acknowledgments [MATHIS96].

6.1 TCP congestion control

TCP uses a window based protocol for flow control. TCP connections provide end-to-end flow control to limit the number of packets in the network. The flow control is enforced by two windows. The receiver's window (RCVWND) is enforced by the receiver as measure of its buffering capacity. The congestion window (CWND) is kept at the sender as a measure of the capacity of the network. The sender sends data one window at a time, and cannot send more than the minimum of RCVWND and CWND into the network.

The basic TCP congestion control scheme (we will refer to this as vanilla TCP) consists of the "Slow Start" and "Congestion Avoidance" phases. The variable SSTHRESH is maintained at the source to distinguish between the two phases. The source starts transmission in the slow start phase by sending one segment (typically512 Bytes) of data, i.e., CWND = 1 TCP segment. When the source receives an acknowledgment for a new segment, the source increments CWND by 1. Since the time between the sending of a segment and the receipt of its ack is an indication of the Round Trip Time (RTT) of the connection, CWND is doubled every round trip time during the slow start phase. The slow start phase continues until CWND reaches SSTHRESH (typically initialized to 64K bytes) and then the congestion avoidance phase begins. During the congestion avoidance phase, the source increases its CWND by 1/CWND every time a segment is acknowledged. The slow start and the congestion avoidance phases correspond to an exponential increase and a linear increase of the congestion window every round trip time respectively.

If a TCP connection loses a packet, the destination responds by sending duplicate acks for each out-of-order packet received. The source maintains a retransmission timeout for the last unacknowledged packet. The timeout value is reset each time a new segment is acknowledged. The source detects congestion by the triggering of the retransmission timeout. At this point, the source sets SSTHRESH to half of CWND. More precisely, SSTHRESH is set to max(2,min(CWND/2, RCVWND)). CWND is set to one segment size.

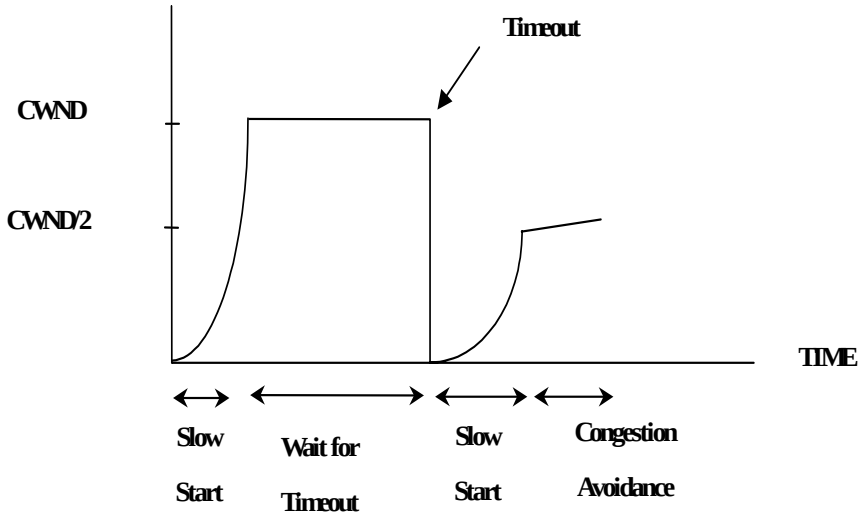


Figure 5 TCP Congestion Control

As a result, CWND < SSTHRESH and the source enters the slow start phase. The source then retransmits the lost segment and increases its CWND by one every time a new segment is acknowledged. It takes $\log_2(CWND_{orig}/(2\times MSS))$ RTTs from the point when the congestion was detected, for CWND to reach the target value of half its original size ($CWND_{orig}$). Here, MSS is the TCP maximum segment size value in bytes. This behavior is unaffected by the number of segments lost from a particular window.

If a single segment is lost, and if the receiver buffers out of order segments, then the sender receives a cumulative acknowledgment and recovers from the congestion. Otherwise, the sender attempts to retransmit all the segments since the lost segment. In either case, the sender congestion window increases by one segment for each acknowledgment received, and not for the number of segments acknowledged. This recovery can be very slow for long latency satellite connections. The recovery behavior corresponds to a go-back-N retransmission policy at the sender. Note that although the congestion window may increase beyond the advertised receiver window (RCVWND), the source window is limited by the minimum of the two. The typical changes in the source window plotted against time are shown in Figure 6.

Most TCP implementations use a 500 ms timer granularity for the retransmission timeout. The TCP source estimates the Round Trip Time (RTT) of the connection by measuring the time (number of ticks of the timer) between the sending of a segment and the receipt of the ack for the segment. The retransmission timer is calculated as a function of the estimates of the average and mean-deviation of the RTT [JACOBS88]. Because of coarse grained TCP timers, when there is loss due to congestion, significant time may be lost waiting for the retransmission timeout to trigger. Once the source has sent out all the segments allowed by its window, it does not send any new segments when duplicate acks are being received. When the retransmission timeout triggers, the connection enters the slow start phase. As a result, the link may remain idle for a long time and experience low utilization.

Coarse granularity TCP timers and retransmission of segments by the go-back-N policy are the main reasons that TCP sources can experience low throughput and high file transfer delays during congestion.

During congestion, the TCP window based flow and congestion control mechanisms are unable to efficient performance, especially for large latency connections.

6.2 Design Issues for TCP/IP over ATM

There are several options for transporting non-real time TCP connections over a satellite-ATM network.

The Unspecified Bit Rate (UBR) service provided by ATM networks has no explicit congestion control mechanisms [TM496]. However, it is expected that many TCP implementations will use the UBR service category. TCP employs a window based end-to-end congestion control mechanism to recover from segment loss and avoids congestion collapse. Several studies have analyzed the performance of TCP over the UBR service. TCP sources running over UBR with limited network buffers experience low throughput and high unfairness [FANG95, GOYAL97, LI95, LI96].

Figure 6 illustrates a framework for the various design options available to networks and end-systems for congestion control. Several design options available to UBR networks and end-systems for improving performance. Intelligent drop policies at switches can be used to improve throughput of transport connections. Early Packet Discard (EPD) [ROMANOV95] has been shown to improve TCP throughput but not fairness [GOYAL97]. Enhancements that perform intelligent cell drop policies at the switches need to be developed for UBR to improve transport layer throughput and fairness. A policy for selective cell drop based on per-VC buffer management can be used to improve fairness. Providing guaranteed minimum rate to the UBR traffic has also been discussed as a possible candidate to improve TCP performance over UBR.

Providing a rate guarantee to the UBR service category can ensure a continuous flow of TCP packets in the network. UBR with guaranteed rate requires no additional signaling requirements or standards changes, and can be implemented on current switches that support the UBR service. Guaranteed rate service is intended for applications which do not need any QoS guarantees, but whose performance depends on the availability of a continuous amount of bandwidth. The goal of providing guaranteed rate is to protect the UBR service category from total bandwidth starvation, and provide a continuous minimum bandwidth guarantee. In the presence of high load of higher priority Constant Bit Rate (CBR), Variable Bit Rate (VBR) and Available Bit Rate

(ABR) traffic, TCP congestion control mechanisms are expected to benefit from a guaranteed minimum rate.

Guaranteed Frame Rate (GFR) has been recently proposed in the ATM Forum as an enhancement to the UBR service category. Guaranteed Frame Rate will provide a minimum rate guarantee to VCs at the frame level. The GFR service also allows for the fair usage of any extra network bandwidth. GFR requires minimum signaling and connection management functions, and depends on the network's ability to provide a minimum rate to each VC. GFR is likely to be used by applications that can neither specify the traffic parameters needed for a VBR VC, nor have capability for ABR (for rate based feedback control). Current internetworking applications fall into this category, and are not designed to run over QoS based networks. These applications could benefit from a minimum rate guarantee by the network, along with an opportunity to fairly use any additional bandwidth left over from higher priority connections. In the case of LANs connected by Satellite-ATM backbones, network elements outside the ATM network could also benefit from GFR guarantees. For example, IP routers separated by a Satellite-ATM network could use GFR VCs to exchange control messages.

The Available Bit Rate (ABR) service category is another option to implement TCP/IP over ATM. The *Available Bit Rate (ABR)* service category is specified by a PCR and Minimum Cell Rate (MCR) which is guaranteed by the network. The bandwidth allocated by the network to an ABR connection may vary during the life of a connection, but may not be less than MCR. ABR connections use a rate-based closed-loop end-to-end feedback-control mechanism for congestion control. The network tries to maintain a low Cell Loss Ratio by changing the allowed cell rates (ACR) at which a source can send. Switches can also use the virtual source/virtual destination (VS/VD) feature to segment the ABR control loop into smaller loops. In a VS/VD network, a switch can additionally behave both as a (virtual) destination end system and as a (virtual) source end system. This feature can allow feedback from nearby switches to reach sources faster, and allow hop-by-hop control. Several studies have examined the performance of TCP/IP over various ABR feedback control schemes. These studies have indicated that good schemes can effectively reduce the buffer requirement for TCP over satellite especially for long delay paths.

In addition to network based drop policies, end-to-end flow control and congestion control policies can be effective in improving TCP performance over UBR. The fast retransmit and recovery mechanism [FRR], can be used in addition to slow start and congestion avoidance to quickly recover from isolated segment losses. The selective acknowledgments (SACK) option [MATHIS96] has been proposed to recover quickly from multiple segment losses [FLOYD95]. A change to TCP's fast retransmit and recovery has also been suggested in [FALL96] and [HOE96].

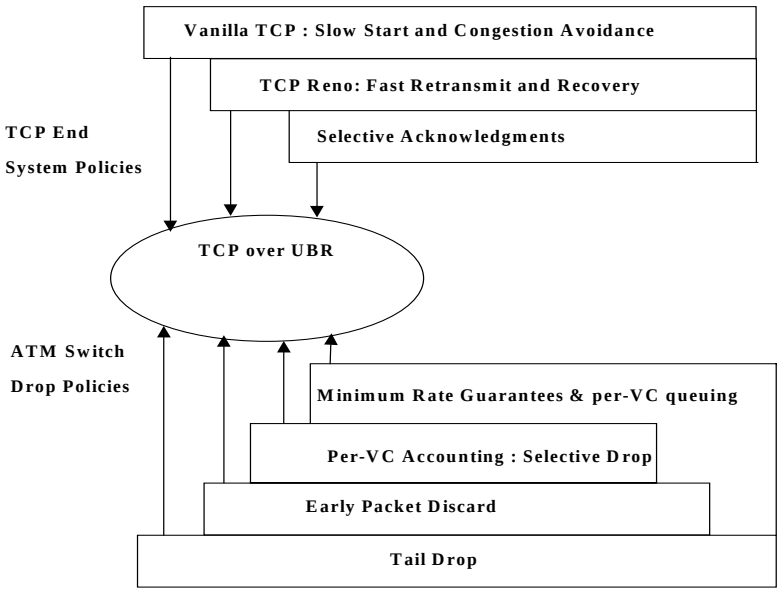


Figure 6 Design Issues for TCP over ATM

Several design options must be explored for improving the performance of TCP over ATM.

Both end-system, as well as network policies must be studied for optimal TCP performance.

P based its own flow and congestion control mechanisms, many TCP/IP connections are expected to use the UBR service. As a result, it is important to assess the performance of TCP/IP over UBR in a satellite network.

In its simplest form, an ATM switch implements a tail drop policy for the UBR service category. When a cell arrives at the FIFO queue, if the queue is full, the cell is dropped, otherwise the cell is accepted. If a cell is dropped, the TCP source loses time, waiting for the retransmission

timeout. Even though TCP congestion mechanisms effectively recover from loss, the resulting throughput can be very low. It is also known that simple FIFO buffering with tail drop results in excessive wasted bandwidth. Simple tail drop of ATM cells results in the receipt of incomplete segments. When part of a segment is dropped at the switch, the incomplete segment is dropped at the destination during reassembly. This wasted bandwidth further reduces the effective TCP throughput. Performance of TCP over UBR can be improved using buffer management policies and end-system policies. In this section we describe the important performance results of TCP over UBR and its enhancements. This section does not present the study of end-system policies including TCP parameters. In general TCP performance is also effected by TCP congestion control mechanisms and TCP parameters such as segment size, timer granularity, receiver window size, slow start threshold, and initial window size.

7.1 Performance Metrics

The performance of TCP over UBR is measured by the *efficiency* and *fairness* defined as follows:

$$Efficiency = \frac{\sum_{i=1}^{N} x_i}{x_{\max}}$$

Where x_i is the throughput of the i^{th} TCP connection, x_{max} is the maximum TCP throughput achievable on the given network, and N is the number of TCP connections. The TCP throughputs are measured at the destination TCP layers. Throughput is defined as the total number of bytes delivered to the destination application, divided by the total simulation time. The results are reported in Mbps. The maximum possible TCP throughput (x_{max}) is the throughput attainable by the TCP layer running over UBR on a 155.52 Mbps link. For 9180 bytes of data (TCP maximum segment size), the ATM layer receives 9180 bytes of data + 20 bytes of TCP header + 20 bytes of IP header + 8 bytes of LLC header + 8 bytes of AAL5 trailer. These are padded to produce 193 ATM cells. Thus, each TCP segment results in 10229 bytes at the ATM layer. From this, the maximum possible throughput = 9180/10229 = 89.7% = 135 Mbps approximately on a 155.52 Mbps link.

$$Fairness = \frac{\sum_{i=1}^{N}\left(\frac{x_i}{e_i}\right)^2}{N \times \left(\sum_{i=1}^{N}\frac{x_i}{e_i}\right)^2}$$

Where e_i is the expected throughput of the i^{th} TCP connection. Both metrics lie between 0 and 1, and the desired values of efficiency and fairness are close to 1 [JAIN91]. In the symmetrical configuration presented above,

$$e_i = \frac{x_{\max}}{N}$$

and the fairness metric represents a equal share of the available data rate. For more complex configurations, the fairness metric specifies max-min fairness [JAIN91].

TCP performance over UBR can be measured by the efficiency and fairness metrics.

TCP performs best when there is zero loss. In this situation, TCP is able to fill the pipe and fully utilize the link bandwidth. During the exponential rise phase (slow start), TCP sources send out two segments for every segment that is acked. For N TCP sources, in the worst case, a switch can receive a whole window's worth of segments from N-1 sources while it is still clearing out segments from the window of the Nth source. As a result, the switch can have buffer occupancies of up to the sum of all the TCP maximum sender window sizes. For a switch to guarantee zero loss for TCP over UBR, the amount of buffering required is equal to the sum of the TCP maximum window sizes for all the TCP connections. Note that the maximum window size is determined by the minimum of the sender's congestion window and the receiver's window.

TCP over vanilla UBR results in low fairness in low latency and long latency configurations. This is mainly due to the TCP congestion control mechanisms together with the tail drop policies as discussed earlier. Another reason for poor performance is the synchronization of TCP sources.

TCP connections are synchronized when their sources timeout and retransmit at the same time. This occurs because packets from all sources are dropped forcing them to enter the slow start phase. However, in this case, when the switch buffer is about to overflow, one or two connections get lucky and their entire windows are accepted while the segments from all other connections are dropped. All these connections wait for a timeout and stop sending data into the network. The connections that were not dropped send their next window and keep filling up the buffer. All other connections timeout and retransmit at the same time. This results in their segments being dropped again and the synchronization effect is seen. The sources that escape the synchronization get most of the bandwidth. The synchronization effect is particularly important when the number of competing connections is small.

For smaller buffer sizes, efficiency typically increases with increasing buffer sizes. Larger buffer sizes result in more cells being accepted before loss occurs, and therefore higher efficiency. This is a direct result of the dependence of the buffer requirements to the sum of the TCP window sizes.

TCP over UBR can result in poor performance.

Performance can be significantly improved using buffer management policies.

ffic. All these proposals all drop packets when the buffer occupancy exceeds a certain threshold. Most buffer management schemes improve the efficiency of TCP over UBR. However, only some of the schemes affect the fairness properties of TCP over UBR. The proposals for buffer management can be classified into four groups based on whether they maintain multiple buffer occupancies (Multiple Accounting -- MA) or a single global buffer occupancy (Single Accounting -- SA), and whether they use multiple discard thresholds (Multiple Thresholds -- MT) or a single global discard Threshold (Single Threshold -- ST). The SA schemes maintain a single count of the number of cells currently in the buffer. The MA schemes classify the traffic into several classes and maintain a separate count for the number of cells in the buffer for each class. Typically, each class corresponds to a single connection, and these schemes maintain per-connection occupancies. In cases where the number of connections far exceeds the buffer size, the added over-head of per-connection accounting may be very expensive. In this case, a set of

active connections is defined as those connections with at least one packet in the buffer, and only the buffer occupancies of active connections are maintained.

Table 2 Classification of Buffer Management Schemes

Buffer Management Class	Examples	Threshold Type (Static/Dynamic)	Drop Type (Deterministic/ Probabilistic)	Tag Sensitive (Yes/No)
SA--ST	EPD, PPD	Static	Deterministic	No
	RED	Static	Probabilistic	No
MA--ST	FRED	Dynamic	Probabilistic	No
	SD, FBA	Dynamic	Deterministic	No
	VQ+Dynamic EPD	Dynamic	Deterministic	No
MA--MT	PME+ERED	Static	Probabilistic	Yes
	DFBA	Dynamic	Probabilistic	Yes
	VQ+MCR scheduling	Dynamic	Deterministic	No
SA--MT	Priority Drop	Static	Deterministic	Yes

Schemes with a global threshold (ST) compare the buffer occupancy(s) with a single threshold and drop packets when the buffer occupancy exceeds the threshold. Multiple thresholds (MT) can be maintained corresponding to classes, connections or to provide differentiated services. Several modifications to this drop behavior can be implemented. Some schemes like RED and FRED compare the average(s) of the buffer occupancy(s) to the threshold(s). Some like EPD

maintain static threshold(s) while others like FBA maintain dynamic threshold(s). In some, packet discard may be probabilistic (RED) while others drop packets deterministically (EPD/PPD). Finally, some schemes may differentiate packets based on packet tags. Examples of packet tags are the CLP bit in ATM cells or the TOS octet in the IP header of the IETF's differentiated services architecture. Table 2 lists the four classes of buffer management schemes and examples of schemes for these classes. The example schemes are briefly discussed below.

The first SA-ST schemes included Early Packet Discard (EPD), Partial Packet Discard (PPD) [ROMANOV95] and Random Early Detection (RED) [FLOYD93]. EPD and PPD improve network efficiency because they minimize the transmission of partial packets by the network. Since they do not discriminate between connections in dropping packets, these schemes are unfair in allocating bandwidth to competing connections [GOYAL98b],[LI96]. For example, when the buffer occupancy reaches the EPD threshold, the next incoming packet is dropped even if the packet belongs to a connection that is has received an unfair share of the bandwidth. Random Early Detection (RED) maintains a global threshold for the average queue. When the average queue exceeds this threshold, RED drops packets probabilistically using a uniform random variable as the drop probability. The basis for this is that uniform dropping will drop packets in proportion to the input rates of the connections. Connections with higher input rates will lose proportionally more packets than connections with lower input rates, thus maintaining equal rate allocation.

However, it has been shown in [LIN97] that proportional dropping cannot guarantee equal bandwidth sharing. The paper also contains a proposal for Flow Random Early Drop (FRED). FRED maintains per-connection buffer occupancies and drops packets probabilistically if the per-connection occupancy exceeds the average queue length. In addition, FRED ensures that each connection has at least a minimum number of packets in the queue. In this way, FRED ensures that each flow has roughly the same number of packets {\em in the buffer}, and FCFS scheduling guarantees equal sharing of bandwidth. FRED can be classified as one that maintains per-connection queue lengths, but has a global threshold (MA-ST).

The Selective Drop (SD) [GOYAL98b] and Fair Buffer Allocation (FBA) [HEIN] schemes are MA-ST schemes proposed for the ATM UBR service category. These schemes use per-

connection accounting to maintain the current buffer utilization of each UBR Virtual Channel (VC). A fair allocation is calculated for each VC, and if the VC's buffer occupancy exceeds its fair allocation, its subsequent incoming packet is dropped. Both schemes maintain a threshold R, as a fraction of the buffer capacity K. When the total buffer occupancy exceeds R×K, new packets are dropped depending on the VC_i's buffer occupancy (Y_i). In the Selective Drop scheme, a VC's entire packet is dropped if

Selective Drop:

$$(X > R)\text{AND}\left(\frac{Y_i \times N_a}{X} > Z\right)$$

Fair Buffer Allocation:

$$(X > R)\text{AND}\left(\frac{Y_i \times N_a}{X} > Z \times \frac{K-R}{X-R}\right)$$

where N_a is the number of active VCs (VCs with at least one cell the buffer), and Z is another threshold parameter ($0 < Z \leq 1$) used to scale the effective drop threshold.

Both Selective Drop and FBA improve both fairness and efficiency of TCP over UBR. This is because cells from overloading connections are dropped in preference to underloading ones. As a result, they are effective in breaking TCP synchronization. When the buffer exceeds the threshold, only cells from overloading connections are dropped. This frees up some bandwidth and allows the underloading connections to increase their window and obtain more throughput.

The Virtual Queuing (VQ) [WU97] scheme is unique because it achieves fair buffer allocation by emulates on a single FIFO queue, a per-VC queued round-robin server. At each cell transmit time, a per-VC accounting variable (γ_i) is decremented in a round-robin manner, and is incremented whenever a cell of that VC is admitted in the buffer. When γ_i exceeds a fixed threshold, incoming packets of the ith VC are dropped. An enhancement called Dynamic EPD changes the above drop threshold to include only those sessions that are sending less than their fair shares.

Since the above MA-ST schemes compare the per-connection queue lengths (or virtual variables with equal weights) with a global threshold, they can only guarantee equal buffer occupancy (and

thus throughput) to the competing connections. These schemes do not allow for specifying a guaranteed rate for connections or groups of connections. Moreover, in their present forms, they cannot support packet priority based on tagging.

Another enhancement to VQ, called MCR scheduling [SIU97], proposes the emulation of a weighted scheduler to provide Minimum Cell Rate (MCR) guarantees to ATM connections. In this scheme, a per-VC, weighted variable (W_i) is maintained, and compared with a global threshold. A time interval T is selected, at the end of which, W_i is incremented by $MCR_i \times T$ for each VC i. The remaining algorithm is similar to VQ. As a result of this weighted update, MCRs can be guaranteed. However, the implementation of this scheme involves the update of W_i for each VC after every time T. To provide tight MCR bounds, a smaller value of T must be chosen, and this increases the complexity of the scheme. For best effort traffic (like UBR), thousands of VC could be sharing the buffer, and this dependence on the number of VCs is not an efficient solution to the buffer management problem. Since the variable W_i is updated differently for each VC i, this is equivalent to having different thresholds for each VC at the start of the interval. These thresholds are then updated in the opposite direction of W_i. As a result, VQ+MCR scheduling can be classified as an MA-MT scheme.

[FENG] proposes a combination of a Packet Marking Engine (PME) and an Enhanced RED scheme based on per-connection accounting and multiple thresholds (MA-MT). PME+ERED is designed for the IETF's differentiated services architecture, and can provide loose rate guarantees to connections. The PME measures per-connection bandwidths and probabilistically marks packets if the measured bandwidths are lower than the target bandwidths (multiple thresholds). High priority packets are marked, and low priority packets are unmarked. The ERED mechanism is similar to RED except that the probability of discarding marked mackets is lower that that of discarding unmarked packets. The PME in a node calculates the observed bandwidth over an update interval, by counting the number of accepted packets of each connection by the node. Calculating bandwidth can be complex that may require averaging over several time intervals. Although it has not been formally proven, Enhanced RED can suffer from the same problem as RED because it does not consider the number of packets actually in the queue.

A simple SA-MT scheme can be designed that implements multiple thresholds based on the packet priorities. When the global queue length (single accounting) exceeds the first threshold, packets tagged as lowest priority are dropped. When the queue length exceeds the next threshold, packets from the lowest and the next priority are dropped. This process continues until EPD/PPD is performed on all packets. The performance of such schemes needs to be analyzed. However, these schemes cannot provide per-connection throughput guarantees and suffer from the same problem as EPD, because they do not differentiate between overloading and underloading connections.

Table 3 illustrates the fairness properties of the four buffer management groups presented above.

Table 3 Fairness Properties of Buffer Management Schemes

Class	Equal bandwidth allocation	Weighted bandwidth allocation
SA--ST	No	No
MA--ST	Yes	No
MA--MT	Yes	Yes
SA--MT	--	--

Early Packet Discard improves efficiency but not fairness.

Selective Drop and Fair Buffer Allocation improve both efficiency and fairness.

RED employs probabilistic drop to improve fairness and efficiency.

7. T ction to q

For long latency connections, fast retransmit and recovery hurts the efficiency. This is because congestion typically results in multiple packets being dropped. Fast retransmit and recovery cannot recover from multiple packet losses and slow start is triggered. The additional segments sent by fast retransmit and recovery (while duplicate ACKs are being received) may be

retransmitted during slow start. In WAN links with large bandwidth delay products, the number of retransmitted segments can be significant. Thus, fast retransmit can add to the congestion and reduce throughput.

A modification to Reno is proposed in [FALL96],[HOE96] to overcome this shortcoming. In this scheme, the sender can recover from multiple packet losses without having to time out. In case of small propagation delays, and coarse timer granularities, this mechanism can effectively improve TCP throughput over vanilla TCP.

TCP with Selective Acknowledgments (SACK TCP) has been proposed to efficiently recover from multiple segment losses [MATHIS96]. In SACK TCP, acknowledgments contain additional information about the segments have been received by the destination. When the destination receives out-of-order segments, it sends duplicate ACKs (SACKs) acknowledging the out-of-order segments it has received. From these SACKs, the sending TCP can reconstruct information about the segments not received at the destination. As a result, the sender can recover from multiple dropped segments in about one round trip.

For most cases, for a given drop policy, SACK TCP provides higher efficiency than the corresponding drop policy in vanilla TCP. This confirms the intuition provided by the analysis of SACK that SACK recovers at least as fast as slow start when multiple packets are lost. In fact, for most cases, SACK recovers faster than both fast retransmit/recovery and slow start algorithms. For LANs, the effect of drop policies is very important and can dominate the effect of SACK. For UBR with tail drop, SACK provides a significant improvement over Vanilla and Reno TCPs. However, as the drop policies get more sophisticated, the effect of TCP congestion mechanism is less pronounced. This is because, the typical LAN switch buffer sizes are small compared to the default TCP maximum window of 64K bytes, and so buffer management becomes a very important factor.

The throughput improvement provided by SACK is significant for long latency connections. When the propagation delay is large, a timeout results in the loss of a significant amount of time during slow start from a window of one segment. With Reno TCP (with fast retransmit and recovery), performance is further degraded (for multiple packet losses) because timeout occurs at

a much lower window than vanilla TCP. With SACK TCP, a timeout is avoided most of the time, and recovery is complete within a small number of roundtrips. Even if timeout occurs, the recovery is as fast as slow start but some time may be lost in the earlier retransmissions.

The performance of SACK TCP can be improved by intelligent drop policies like EPD and selective drop. This is consistent with other results of SACK with Vanilla and Reno TCP. Thus, we recommend that intelligent drop policies be used in UBR service.

The fairness values for selective drop are comparable to the values with the other TCP versions. Thus, SACK TCP does not hurt the fairness in TCP connections with an intelligent drop policy like selective drop.

7.5 Buffer Requirements for TCP over UBR+

TCP Fast retransmit and recovery hurts performance in long latency networks.

TCP SACK significantly improves efficiency for TCP over UBR over satellite networks.

udied in

ss buffer

stations

ork with

the satellite network. In general, the satellite network model may include on-board processing and queuing. In the results stated in this section, no on-board processing or queuing is performed. The bottleneck node is the earth station at the entry to the satellite network. As a result, in the experiments, no queuing delays occur in the satellite network. All processing and queuing are performed at the earth stations. The goal of this study is to assess the buffer requirements of the bottleneck node (in this case, the earth station) for good TCP/IP performance.

All simulations use the N source configuration shown in the figure. All sources are identical and persistent TCP sources. The TCP layer always sends a segment as long as it is permitted by the TCP window. Moreover, traffic is unidirectional so that only the sources send data. The destinations only send ACKs. The TCP delayed acknowledgement timer is deactivated, and the receiver sends an ACK as soon as it receives a segment. TCP with selective acknowledgments (SACK TCP) is used in our simulations. All link bandwidths are 155.52 Mbps, and peak cell rate

at the ATM layer is 149.7 Mbps. This accounts for a SONET like overhead in the satellite component of the network.

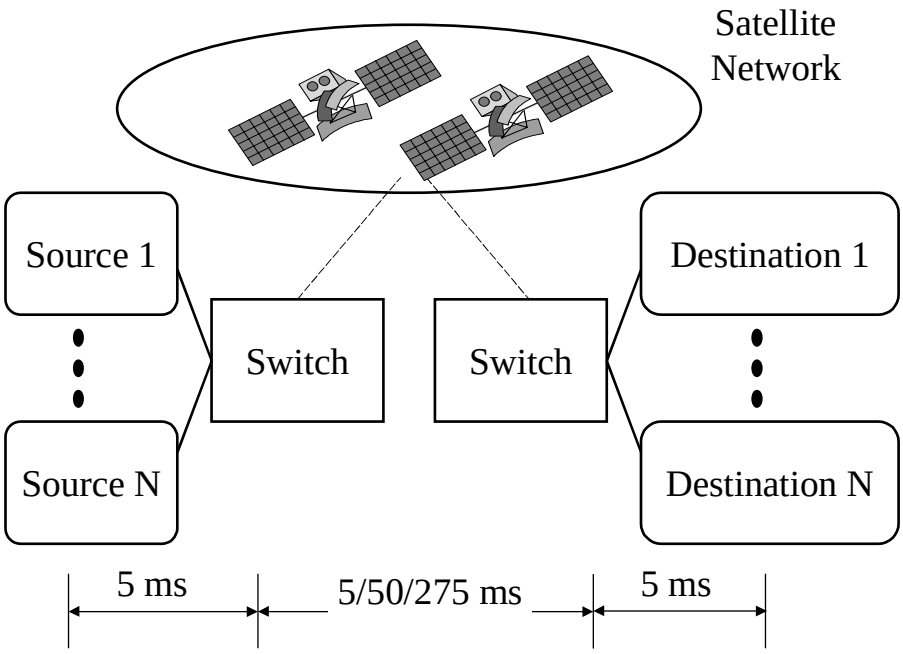


Figure 7: Simulation model for TCP/IP over UBR

The following parameters are used to assess the buffer requirements:

Latency: The primary aim is to study the buffer requirements for long latency connections. A typical latency from earth station to earth station for a single LEO hop is about 5 ms. The latencies for multiple LEO hops can easily be 50 ms or more from earth station to earth station. GEO latencies are typically 275 ms from earth station to earth station for earth stations that are not on the equator. The paper studies these three latencies (5 ms, 50 ms, and 275 ms) with various number of sources and buffer sizes. The link delays between the switches and the end systems are 5 ms in all configurations. This results in round trip propagation delays (RTT) of 30 ms, 120 ms and 570 ms respectively.

Number of sources: To ensure that the recommendations are scalable and general with respect to the number of connections, configurations with 5, 15 and 50 TCP connections on a single bottleneck link are used. For single hop LEO configurations, 15, 50 and 100 sources are used.

Buffer size: This is the most important parameter of this study. The goal is to estimate the smallest buffer size that results in good TCP performance, and is scalable to the number of TCP sources. The values chosen for the buffer size are approximately:

$$Buffer\_size = 2^{-k} \times RTT \times bottleneck\_link\_data\_rate, k = -1..6$$

i.e., 2, 1, 0.5, 0.25, 0.125, 0.0625, 0.031 and 0.016 multiples of the round trip delay-bandwidth product of the TCP connections are chosen. The resulting buffer sizes (in cells) used in the earth stations are as follows:

- *Single LEO:* 375, 750, 1500, 3000, 6000, 12000 (=1 RTT), 24000 and 36000 cells.
- *Multiple LEO:* 780, 1560, 3125, 6250, 12500, 25000, 50000 (=1 RTT), and 100000 cells.
- *GEO:* 3125, 6250, 12500, 25000, 50000, 100000, 200000 (=1 RTT), and 400000 cells.

The plots of the buffer size against the achieved TCP throughput for different delay-bandwidth products and number of sources are shown. The asymptotic nature of this graph provides information about the optimal buffer size for the best performance.

Buffer allocation policy: *Selective drop* is used to fairly allocate switch buffers to the competing TCP connections.

End system policies: SACK TCP [RF2018] is used, for this study. The maximum value of the TCP receiver window is 600000 bytes, 2500000 bytes and 8704000 bytes for single hop LEO, multiple hop LEO and GEO respectively. These window sizes are obtained using the TCP window scaling option, and are sufficient to achieve full utilization on the 155.52 Mbps links. The TCP maximum segment size is 9180 bytes. This conforms to the segment size recommended for TCP connections over long latency connections. The TCP timer granularity is set to 100 ms. This value limits the time taken for retransmissions to multiples of 100 ms. The value is chosen to balance the attainable throughput with the limitations of the TCP RTT measurement algorithm. With large granularity, TCP could wait a long time before detecting packet loss,

resulting in poor throughput. Finer granularity of the retransmission timer leads to false timeouts even with a small variation in the measured RTT values.

Figures 4, 5, and 6 show the resulting TCP efficiencies for the 3 different latencies. Each point in the figure shows the efficiency (total achieved TCP throughput divided by maximum possible throughput) against the buffer size used. Each figure plots a different latency, and each set of points (connected by a line) in a figure represents a particular value of N (the number of sources).

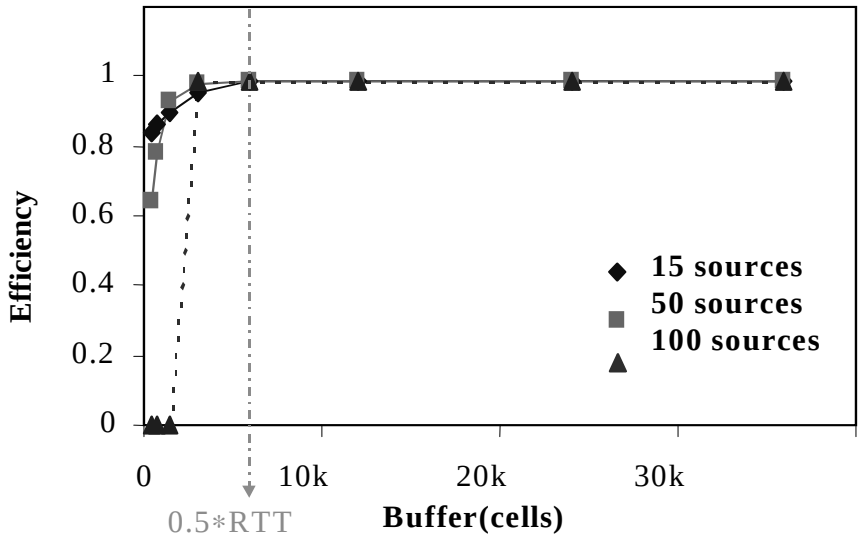


Figure 8 TCP/IP UBR buffer requirements for single hop LEO

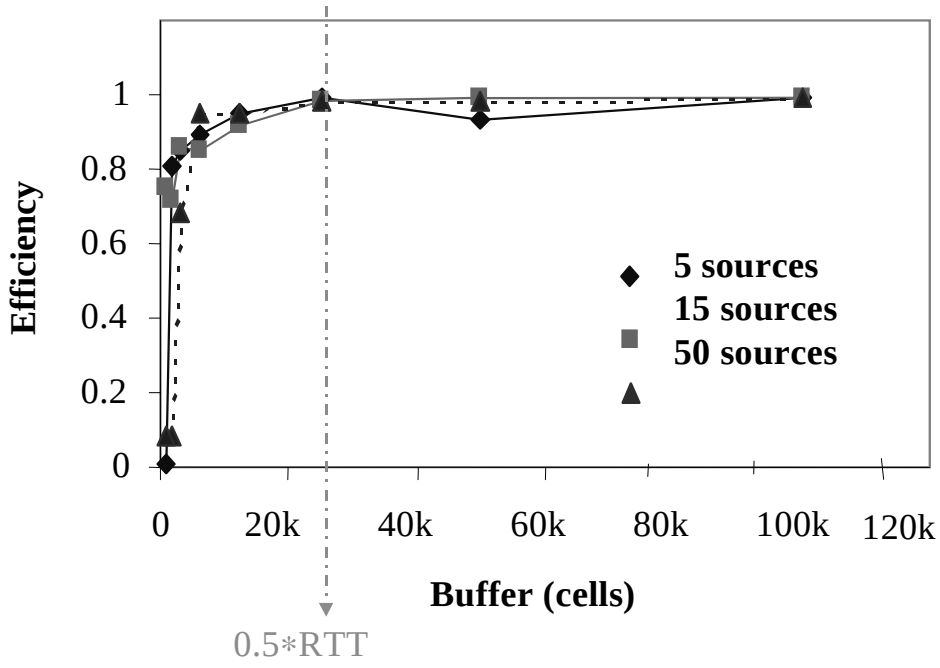


Figure 9 TCP/IP UBR buffer requirements for multiple hop LEO

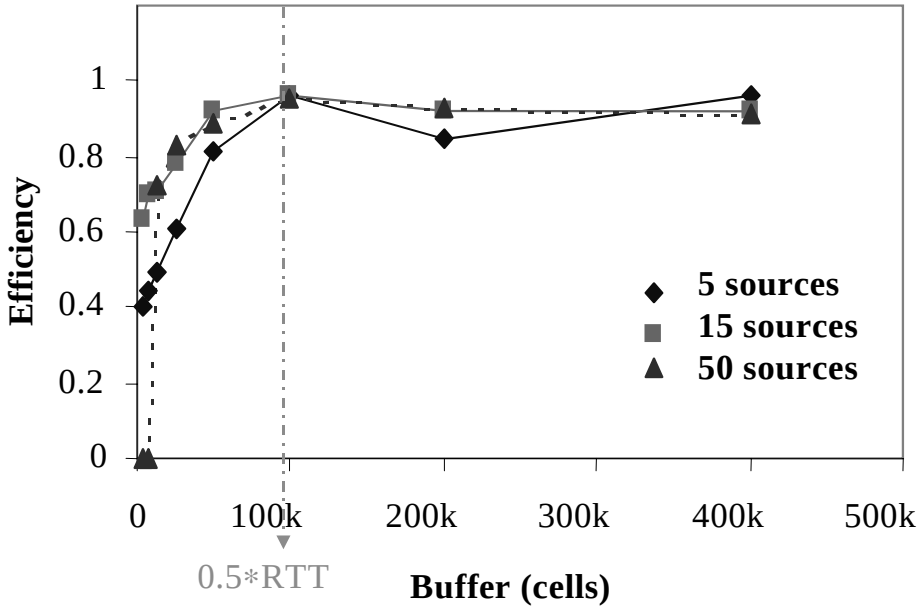


Figure 10 TCP/ IP UBR buffer requirements for single hop GEO

For very small buffer sizes, (0.016RTT, 0.031RTT, 0.0625RTT), the resulting TCP throughput is very low. In fact, for a large number of sources (N=50), the throughput is sometimes close to

zero. For small buffer sizes, the performance of TCP/IP deteriorates with increasing number of sources. This is because more TCP packets are dropped from each connection causing TCP timeout and retransmissions. This results in decreased throughput. For moderate buffer sizes (less then 1 round trip delay times bandwidth), TCP throughput increases with increasing buffer sizes. TCP throughput asymptotically approaches the maximal value with further increase in buffer sizes.

TCP performance over UBR for sufficiently large buffer sizes is scalable with respect to the number of TCP sources. The throughput is never 100%, but for buffers greater than 0.5xRTT, the average TCP throughput is over 98% irrespective of the number of sources. As a result, each queuing point must have sufficient buffers to support one delay-bandwidth product worth of TCP data so that it can ensure minimal loss.

The simulation results show that TCP sources with a good per-VC buffer allocation policy like selective drop, can effectively share the link bandwidth. A buffer size of about 0.5RTT to 1RTT is sufficient to provide over 98% throughput to infinite SACK TCP traffic for long latency networks and a large number of sources. This buffer requirement is independent of the number of sources. The fairness in the throughputs measured by the fairness index is high due to the selective drop policy [KOTA97].

To conclude this section, TCP performance over UBR can be improved by either improving TCP using selective acknowledgments, or by introducing intelligent buffer management policies at the switches. Efficient buffer management has a more significant influence on LANs because of the limited buffer sizes in LAN switches compared to the TCP maximum window size. In long latency networks (WANs), the drop policies have a smaller impact if both the switch buffer sizes and the TCP windows are of the order of the bandwidth-delay product of the network. Also, the TCP linear increase is much slower in WANs than in LANs because the WAN RTTs are higher.

7.6 Guaranteed Frame Rate

Buffer requirements for SACK TCP over UBR with Selective Drop is about 0.5RTT.

With SACK and selective drop, 0.5RTT buffers result in high efficiency and fairness for satellite networks even for a large number of sources.

) service

d packet

marking to provide minimum cell rate guarantees on a per-connection basis. *So far, no studies have been reported to assess the performance of GFR for satellite networks. This is a topic of future study.*

GFR has been recently proposed in the ATM Forum as an enhancement to the UBR service category. Guaranteed Frame Rate will provide a minimum rate guarantee to VCs at the frame level. The GFR service also allows for the fair usage of any extra network bandwidth. GFR requires minimum signaling and connection management functions, and depends on the network's ability to provide a minimum rate to each VC. GFR is likely to be used by applications that can neither specify the traffic parameters needed for a VBR VC, nor have cability for ABR (for rate based feedback control). Current internetworking applications fall into this category, and are not designed to run over QoS based networks. These applications could benefit from a minimum rate guarantee by the network, along with an opportunity to fairly use any additional bandwidth left over from higher priority connections. In the case of LANs connected by ATM backbones, network elements outside the ATM network could also benefit from GFR guarantees. For example, IP routers separated by an ATM network could use GFR VCs to exchange control messages. Figure 11 illustrates such a case where the ATM cloud connects several LANs and routers. ATM end systems may also establish GFR VCs for connections that can benefit from a minimum throughput guarantee.

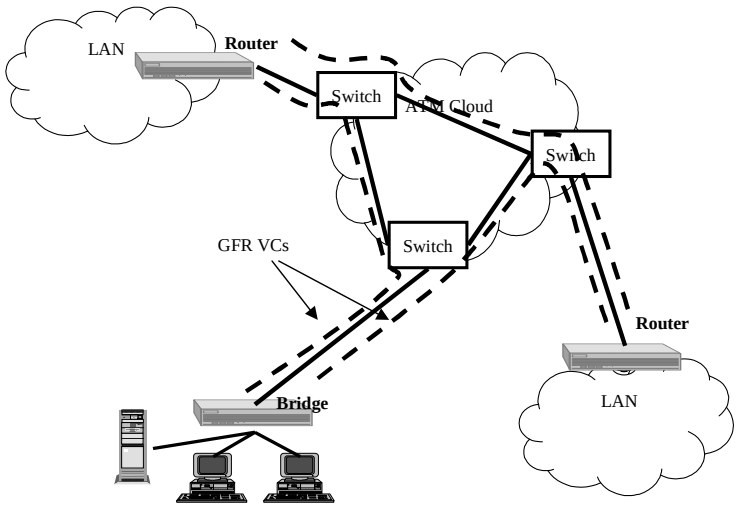


Figure 11 GFR in ATM connected LANs

The original GFR proposals give the basic definition of the GFR service. GFR provides a minimum rate guarantee to the *frames* of a VC. The guarantee requires the specification of a maximum frame size (MFS) of the VC. If the user sends packets (or frames) smaller than the maximum frame size, at a rate less than the minimum cell rate (MCR), then all the packets are expected to be delivered by the network with minimum loss. If the user sends packets at a rate higher than the MCR, it should still receive at least the minimum rate. The minimum rate is guaranteed to the untagged frames of the connection. In addition, a connection sending in excess of the minimum rate should receive a fair share of any unused network capacity. The exact specification of the fair share has been left unspecified by the ATM Forum. Although the GFR specification is not yet finalized, the above discussion captures the essence of the service.

There are three basic design options that can be used by the network to provide the per-VC minimum rate guarantees for GFR -- tagging, buffer management, and queueing:

- **Tagging**: Network based tagging (or policing) can be used as a means of marking non-conforming packets before they enter the network. This form of tagging is usually performed when the connection enters the network. Figure 12 shows the role of network based tagging in providing a minimum rate service in a network. Network based tagging on a per-VC level requires some per-VC state information to be maintained by the network and increases the complexity of the network element. Tagging can isolate conforming and non-conforming traffic of each VC so that other rate enforcing mechanisms can use this information to schedule the conforming traffic in preference to non-conforming traffic. In a more general sense, policing can be used to discard non-conforming packets, thus allowing only conforming packets to enter the network.

- **Buffer management**: Buffer management is typically performed by a network element (like a switch or a router) to control the number of packets entering its buffers. In a shared buffer environment, where multiple VCs share common buffer space, per-VC buffer management can control the buffer occupancies of individual VCs. Per-VC buffer management uses per-VC accounting to keep track of the buffer occupancies of each VC. Figure 12 shows the role of buffer management in the connection path. Examples of per-VC buffer management schemes are Selective Drop and Fair Buffer Allocation. Per-VC accounting introduces

overhead, but without per-VC accounting it is difficult to control the buffer occupancies of individual VCs (unless non-conforming packets are dropped at the entrance to the network by the policer). Note that per-VC buffer management uses a single FIFO queue for all the VCs. This is different from per-VC queuing and scheduling discussed below.

- **Scheduling**: Figure 12 illustrates the position of scheduling in providing rate guarantees. While tagging and buffer management, control the entry of packets into a network element, queuing strategies determine how packets are scheduled onto the next hop. FIFO queuing cannot isolate packets from various VCs at the egress of the queue. As a result, in a FIFO queue, packets are scheduled in the order in which they enter the buffer. Per-VC queuing, on the other hand, maintains a separate queue for each VC in the buffer. A scheduling mechanism can select between the queues at each scheduling time. However, scheduling adds the cost of per-VC queuing and the service discipline. For a simple service like GFR, this additional cost may be undesirable.

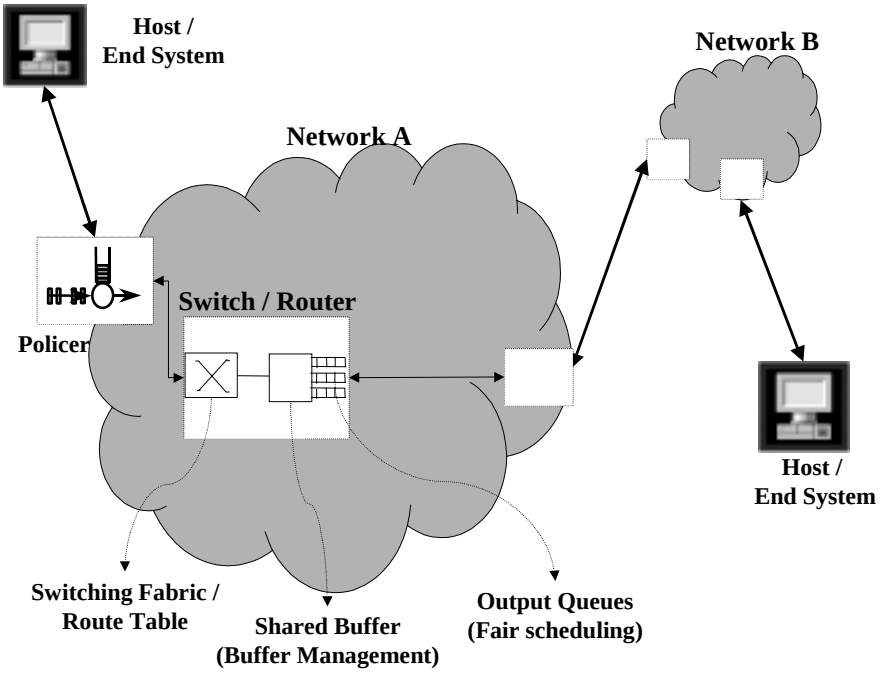


Figure 12 Buffering, Scheduling and Policing in the Network

Several proposals have been made ([BASAK97],[BONAV97]) to provide rate guarantees to TCP sources with FIFO queuing in the network. The bursty nature of TCP traffic makes it difficult to provide per-VC rate guarantees using FIFO queuing. Per-VC scheduling was recommended to provide rate guarantees to TCP connections. However, all these studies were performed at high target network utilization, i.e., most of the network buffers were allocated to the GFR VCs. We show that rate guarantees are achievable with a FIFO buffer for low buffer allocation.

All the previous studies have examined TCP traffic with a single TCP per VC. Per-VC buffer management for such cases, reduces to per-TCP buffer management. However, routers that would use GFR VCs, would multiplex many TCP connections over a single VC. For VCs with several aggregated TCPs, per-VC control is unaware of each TCP in the VC. Moreover, aggregate TCP traffic characteristics and control requirements may be different from those of single TCP streams. Minimum rate allocation to TCP traffic with FIFO buffers is presented in [GOYAL98d].

8 ABR over Satellite

In this section, we present an overview of the ABR service and its implementations on satellite networks. The main focus of this section is on the feasibility of ABR services over

The Guaranteed Frame Rate Service provides minimum rate guarantees to the AAL frames of VCs.

The Guaranteed Frame Rate Service can be used to connect ATM separated IP networks.

ABR mechanisms allow the network to divide the available bandwidth fairly and efficiently among the active traffic sources. In the ABR traffic management framework, the source end systems limit their data transmission to rates allowed by the network. The network consists of switches that use their current load information to calculate the allowable rates for the sources. These rates are sent to the sources as feedback via resource management (RM) cells. The ABR traffic management model is a rate-based end-to-end closed-loop model. There are three ways for switches to give feedback to the sources. First, each cell header contains a bit called Explicit Forward Congestion Indication (EFCI), which can be set by a congested switch. Such switches are called binary or EFCI switches. Second, RM cells have two bits in their payload, called the

Congestion Indication (CI) bit and the No Increase (NI) bit, that can be set by congested switches. Switches that use only this mechanism are called relative rate marking switches. Third, the RM cells also have another field in their payload called explicit rate (ER) that can be reduced by congested switches to any desired value. Such switches are called Explicit Rate switches. RM cells are generated by the sources and travel along the data path to the destination end systems. The destinations simply return the RM cells to the sources. Explicit rate switches normally wait for the arrival of an RM cell to give feedback to a source. However, under extreme congestion, they are allowed to generate an RM cell and send it immediately to the source. This optional mechanism is called backward explicit congestion notification (BECN).

At the time of connection setup, ABR sources negotiate several operating parameters with the network. The first among these is the peak cell rate (PCR). This is the maximum rate at which the source will be allowed to transmit on this virtual circuit (VC). The source can also request a minimum cell rate (MCR) which is the guaranteed minimum rate. The network has to reserve this bandwidth for the VC. During the data transmission stage, the rate at which a source is allowed to send at any particular instant is called the allowed cell rate (ACR). The ACR is dynamically changed between MCR and PCR. At the beginning of the connection, and after long idle intervals, ACR is set to initial cell rate (ICR). A complete list of parameters used in the ABR mechanism is given in **[**TM4096**]**. ABR switches can use the virtual source/virtual destination (VS/VD) feature to segment the ABR control loop into smaller loops. In a VS/VD network, a switch can additionally behave both as a (virtual) destination end system and as a (virtual) source end system. As a destination end system, it turns around the RM cells to the sources from one segment. As a source end system, it generates RM cells for the next segment. This feature can allow feedback from nearby switches to reach sources faster, and achieve hop-by-hop control.

Traffic Management Specifications V 4.0, released in June 1996, specifies various parameters for ABR connections and the rules to be followed by ABR sources and destinations **[**TM4096**]**. An elaborate discussion of ABR parameters, source end system rules and destination end system rules is available in **[**JAIN96**]**. Among the various rules specified for ABR sources, rules 5 and 6 can significantly degrade ABR performance over satellite links if parameters used there do not have appropriate values.

8.2 ABR Source Rules

ABR end systems must follow a set of rules while sending out data and RM cells into the network. Of the source rules, rules 5 and 6 have the most impact on satellite-ATM networks.

8.2.1 ABR Source Rule 5 over Satellite

Rule 5 states that "*Before sending a forward in-rate RM cell, if ACR > ICR and the time T that has elapsed since the last in-rate forward RM cell was sent is greater than ADTF, then ACR shall be reduced to ICR.*"

Here, ACR is the Allowed Cell Rate and ICR is the Initial Cell Rate of the source. Hence, ADTF (ACR Decrease Time Factor) is the maximum time allowed between consecutive forward RM cells before the ACR is reduced to ICR.

The purpose of rule 5 is to solve the problem of ACR Retention. If a source sends an RM cell when the network is not heavily loaded, the source may be granted a very high ACR. The source can then retain that ACR and use it when the network is highly loaded. This may cause switch buffers to overflow. Rule 5 provides a simple timeout solution to this problem with ADTF as the time out value - ACR reduces to ICR whenever the time interval between two consecutive forward RM cells exceeds ADTF **[**JAIN96**]**.

In the case of long delay satellite links, if the ICR is low, starting from ICR and reaching an ACR where the link bandwidth is properly utilized may take a very long time. Hence it is imperative that rule 5 does not get triggered unnecessarily, either because traffic is bursty or the data rate is low. In either case, reaching an optimum ACR from a low ICR may take a long time, during which link utilization will be poor. Hence, triggering of rule 5 should be delayed by keeping a sufficiently high value of ADTF. As a result, the inter-FRM cell time interval lies well below the ADTF value, and the source does not return to ICR. Allowed values for ADTF range between 0.01 to 10.23 seconds with a granularity of 10 ms. This range provides sufficient flexibility in choosing a good value of ADTF for satellite links.

ABR source rule 5: Good ABR throughput over satellite links requires sufficiently high ADTF values.

8.2.2 ABR Source Rule 6 on ABR over Satellite

Consider the following scenarios:

- There is congestion in the network, and the BRM cells, carrying low ER, are stuck in the switch queues. Consequently, sources continue to send cells into the network at their current high ACR, causing further congestion.

- A link is broken and the source is not getting any feedback from the network. With no feedback available, the source continues to pump cells into the network at its current high ACR. All of these cells are lost.

The scenarios, presented above, suggest that the source should decrease its ACR as a preventive measure if it does not get timely feedback from the network. Source Rule 6 for ABR connections requires the sources to do exactly this. This rule states that "*Before sending an in-rate forward RM-cell, and after following source rule 5, if at least CRM in-rate forward RM-cells have been sent since the last backward RM-cell with BN=0 was received, then ACR shall be reduced by atleast ACR×CDF, unless that reduction would result in a rate below MCR, in which case ACR shall be set to MCR.*"

Here, backward RM-cell with BN=0 means an RM cell that was generated by the source and has been turned around by the destination. CRM is the missing RM-cell count that limits the number of forward RM-cells that may be sent in the absence of received backward RM-cells. MCR is the Minimum Cell Rate. Finally, CDF denotes the Cutoff Decrease Factor and controls the decrease in ACR associated with CRM. CDF can have zero or any power of 2 in the range 1/64 to 1 as its value.

Suppose T was the last time when source received a BRM cell from the network as feedback and since then source has sent CRM×Nrm cells. Here, Nrm is the maximum number of cells a source may send for each forward RM cell. Then rule 6 implies that ACR of the source will be reduced by ACR×CDF immediately and for every further Nrm cells source sends before receiving a new BRM cell.

$$ACR = Max(MCR, ACR - ACR \times CDF)$$

This exponential decrease in ACR for every Nrm cells sent, leads to an almost "free-fall" in ACR, as shown in Figure 13 and Table 4.

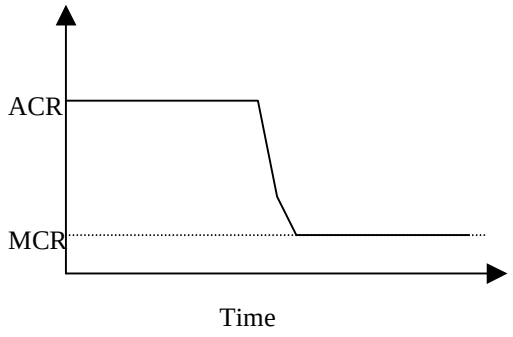


Figure 13 Sudden Drop in ACR with Triggering of Source Rule 6

Table 4 Exponential Decrease in ACR with Triggering of Source Rule 6

ACR	**Number of cells sent since last receiving feedback**
$ACR_{old}(1 - CDF)$	CRM×Nrm
$ACR_{old}(1 - CDF)^2$	(CRM+1)×Nrm
.	.
.	.
$ACR_{old}(1 - CDF)^k$	(CRM + k)×Nrm

This free-fall in ACR continues till a BRM cell is received or ACR reduces to MCR.

Rule 6, once triggered, reduces ACR to MCR quickly unless a BRM is received. Value of CDF (a power of 2 between 1/64 and 1) has little effect in preventing this free-fall in ACR. However rule 6 can be effectively disabled by having a CDF value of 0.

It is clear from the discussion above that the trigger point of rule 6, i.e., the value of the product CRM×Nrm, limits the number of cells from a source that can be "in flight" on a link in the absence of network feedback. Such a situation where no network feedback is available arises during initial startup or when BRM cells are unable to reach the source due to congestion. In the

case of satellite links with long feedback delays, source rule 6 can cause severe reduction in link utilization. This is explained in the next section.

Suppose the satellite link bandwidth (or capacity) is W cells/second. Now, for efficient link utilization, the combined input to satellite link from all sources should be allowed to reach close to W cells/second. If the limit on the value of product CRM×Nrm is not sufficiently high, then it is possible that, on long feedback delay satelite links, a source sends CRM×Nrm cells after the receipt of last BRM cell and before the arrival of the next. In such a situation, rule 6 will be triggered for that source and its ACR will get drastically reduced. This situation can occur with other sources also. Thus, it is very much possible that the combined input to the satellite link from all the sources will never be able to reach to reach the optimum value of W cells/second.

Hence rule 6 may cause inefficient utilization of satellite links if the value of CRM×Nrm is not large enough.

Required values of CRM for efficient GEO link utilization

We have seen that frequent triggering of rule 6 on satellite links will lead to poor utilization of the link. Utilization can be increased by by setting CDF to 0 and disabling rule 6. However, this will make the network susceptible to severe congestion. The solution lies not in disabling rule 6, but in sufficiently delaying its triggering so that efficient link utilization is not compromised.

Efficient link utilization means that sufficient number of cells are 'in flight' so that the link is fully 'filled', i.e., the number of cells in flight is equal to the round trip time (RTT) multiplied by the link capacity.

The product CRM×Nrm specifies the number of cells an ABR source can send at its current ACR starting from the time when it last received a BRM cell. This product should be sufficiently high so that even a single source is able to fill the satellite pipe fully before rule 6 is triggered. This means that the CRM×Nrm value should be at least equal to round trip time (RTT) multiplied by link capacity. In other words,

$$CRM \geq \frac{RTT \times \text{Link Bandwidth}}{Nrm}$$

The value of Nrm can be any power of 2 in the range 2 to 256. Increasing the Nrm value reduces the sensitivity of the source to network conditions, especially at low rates. Hence, Nrm value is generally kept equal to 32, its default value. For a GEO satellite link (550 ms round trip time) with a capacity of 155 Mbps (≈ 365 cells per ms), CRM ≥ 550×365/32 = 6273 (≈ 6k = 6144).

Before August, 1995, TM Specification had allocated 8 bits for CRM thus limiting it to a maximum value of 256. Signaling a CRM greater than 6144 requires at least 13 bits for encoding. For a capacity of 622 Mbps, CRM should be greater than or equal to 24576 which requires at least 15 bits for encoding. For two 622 Mbps satellite hops, CRM should be greater than or equal to 49152 (24576×2) which requires at least 16 bits for encoding.

As a result, the TM Specification V 4.0 has modifications that allow effectively 19 bits for CRM value. In TM Specification V 4.0, CRM is an internal parameter that is derived from a negotiated parameter called Transient Buffer Exposure (TBE). TBE determines the number of cells that a source can transmit before rule 6 is triggered, i.e., TBE essentially equals the product CRM×Nrm. Thus, the relationship between CRM and TBE is given by,

$CRM = \lceil TBE/Nrm \rceil$

TBE gets its name from the fact that it determines the exposure of the switch to sudden traffic transients. It determines the number of cells that may be received at the switch during initial startup or after any long idle period of time. Hence this parameter is negotiated with the network during connection setup based on buffer availability in the network switches. TM Specification V 4.0 sets the size of the TBE parameter to 24 bits. Since Nrm is normally 32, 24-bit TBE allows a 19-bit CRM, which is sufficient for most situations. **[**FAHMY96**]** describes the work that led to setting of TBE size to 24 bits in TM Specification V 4.0.

ABR source rule 6: Good ABR throughput over satellite links requires a high value of TBE so that one round trip times bandwidth worth of cells can be sent into the network without waiting for RM cell feedback.

8.3 ABR Switch Schemes

[To be completed]

8.4 TCP over ABR

[KALYAN97b] provides a comprehensive study of TCP performance over the ABR service category. In the following subsections we present the key issues in TCP over ABR, and highlight their relevance to long delay paths. Most of the discussion assumes that the switches implement a good switch algorithm like ERICA or ERICA+ [KALYAN98b].

8.4.1 Nature of TCP Traffic at the ATM Layer

Data which uses TCP is controlled first by the TCP "slow start" procedure before it appears as traffic to the ATM layer. Suppose we have a large file transfer running on top of TCP. When the file transfer begins, TCP sets its congestion window (CWND) to one. The congestion window increases exponentially with time. Specifically, the window increases by one for every ack received. Over any round trip time (RTT), the congestion window doubles in size. From the switch's point of view, there are two packets input in the next cycle for every packet transmitted in the current cycle (a cycle at a bottleneck is defined as the largest round trip time of any VC going through the bottleneck). In other words, the load (measured over a cycle) at most doubles every cycle. In other words, initially, the TCP load increases exponentially.

Though the application on top of TCP is a persistant application (file-transfer), the TCP traffic as seen at the ATM layer is bursty (i.e., has active and idle periods). Initially, there is a short active period (the first packet is sent) followed by a long idle period (nearly one round-trip time, waiting for an ACK). The length of the active period doubles every round-trip time and the idle period reduces correspondingly. Finally, the active period occupies the entire round-trip time and there is no idle period. After this point, the TCP traffic appears as an infinite (or persistant) traffic stream at the ATM layer. Note that the total TCP load still keeps increasing unless the sources are controlled. This is because, for every packet transmitted, some TCP source window increases by one, which results in the transmission of two packets in the next cycle. However, since the

total number of packets transmitted in a cycle is limited by the delay-bandwidth product, the TCP window increases linearly after the bottleneck is fully loaded. Note that the maximum load, assuming sufficient bottleneck capacity, is the sum of all the TCP receiver windows, each sent at link rate.

When sufficient load is not experienced at the ABR switches, the switch algorithms typically allocate high rates to the sources. This is likely to be the case when a new TCP connection starts sending data. The file transfer data is bottlenecked by the TCP congestion window size and not by the ABR source rate. In this state, we say that the TCP sources are *window-limited*.

The TCP active periods double every round trip time and eventually load the switches and appear as infinite traffic at the ATM layer. The switches now give feedback, asking sources to reduce their rates. The TCP congestion window is now large and is increasing. Hence, it will send data at rate greater than the source's sending rate. The file transfer data is bottlenecked by the ABR source rate and not by the TCP congestion window size. In this state, we say that the TCP sources are *rate-limited*. Observe that UBR cannot rate-limit TCP sources and would need to buffer the entire TCP load inside the network. The minimum number of RTTs required to reach rate-limited operation decreases as the logarithm of the number of sources. In other words, the more the number of sources, the faster they all reach rate-limited operation.

The ABR queues at the switches start increasing when the TCP idle times are not sufficient to clear the queues built up during the TCP active times. The queues may increase until the ABR source rates converge to optimum values. Once the TCP sources are rate-limited and the rates converge to optimum values, the lengths of the ABR queues at the switch will start decreasing. The queues now move over to the source end-system (outside the ATM network).

8.4.2 TCP Performance over ABR

TCP traffic appears as bursty to the ATM Network.

Initially TCP traffic is limited by TCP window sizes (window-limited).

When TCP window size increases, TCP traffic is limited by the network feedback (rate-limited).

uffers to
3R when
uantifies
(through

timeouts) rather than cells (cell loss). If the ABR rates do not converge to optimum values before the cell loss occurs, the effect of the switch congestion scheme may be dominated by factors such as the TCP retransmission timer granularity. Intelligent cell drop policies at the switches can help to significantly improve the throughput.

TCP throughput loss over ABR can be avoided by provisioning sufficient switch buffers. It has been shown that the buffer requirement for TCP over ABR is bounded and small [KALYAN97b]. In particular, the buffer requirement for zero TCP loss over ABR can be bounded by a small constant multiple of the product of the round trip time and bandwidth of the connection. However, note that, even after ABR sources converges to optimum rates, the TCP congestion window can grow till it reaches its maximum (negotiated) value. In such cases, TCP overloads the ABR source and the queues build up at the source end system. If the source queues overflow cell loss will occur, and performance will degrade. In this case, the cell loss occurs outside the ABR network.

The ABR service provides flow control at the ATM level itself. When there is a steady flow of RM cells in the forward and reverse directions, there is a steady flow of feedback from the network. In this state, we say that the ABR control loop has been established and the source rates are primarily controlled by the network feedback (closed-loop control). The network feedback is effective after a time delay. The time delay required for the new feedback to take effect is the sum of the time taken for an RM cell to reach the source from the switch and the time for a cell (sent at the new rate) to reach the switch from the source. This time delay is called the "feedback delay.’’

When the source transmits data after an idle period, there is no reliable feedback from the network. For one round trip time (time taken by a cell to travel from the source to the destination and back), the source rates are primarily controlled by the ABR source end system rules (open-loop control). The open-loop control is replaced by the closed-loop control once the control loop is established. When the traffic on ABR is "bursty” i.e., the traffic consists of busy and idle periods, open-loop control may be exercised at the beginning of every active period (burst). Hence, the source rules assume considerable importance in ABR flow control.

TCP can achieve full throughput over ABR with sufficient buffers in the network.

With limited buffers, buffer management schemes can be used to improve throughput.

8.4.3 Buffer Requirements for TCP over ABR

Most studies for buffer requirements for TCP over ABR over satellite have considered Explicit Rate schemes. In particular, ERICA and ERICA+ have been extensively studied. Emperical and analytical studies have shown that the buffer requirement for TCP over ABR for zero loss transmission is:

$$\text{Buffer} \leq a \times \text{RTT} + b \times \text{Averaging Interval Length} + c \times \text{Feedback delay} \times \text{Link bandwidth}$$

for low values of the coefficients a, b, c and d. This requirement is heavily dependent on the switch algorithm. With the ERICA+ algorithm, typical conservative values of the coefficients are a=3, b=1, and c=1.

The formula is a linear relation on three key factors:

- **Round trip time (RTT)**: Twice the delay through the ABR network or segment (delimited by VS/VD switch(es)).
- **Averaging Interval Length**: A quantity which captures the measurement aspects of a switch congestion control algorithm. Typical measured quantities are: ABR capacity, average queue length, ABR input rate, number of active sources, and VC's rate.
- **Feedback delay**: Twice the delay from the bottleneck to the ABR source (or virtual source). Feedback delay is the minimum time for switch feedback to be effective.

Note that the formula does not depend upon the number of TCP sources. This fact implies that ABR can support TCP (data) applications in a scalable fashion. The buffer requirement is also an indication of the maximum queuing delay through the network. Note that this is a worst case requirement and the average delay is much smaller due the congestion avoidance mechanisms at the ATM layer. As a result, ABR is a better suited for scalable support of interactive applications which involve data large transfers (like web-based downloading etc).

The above formula assumes that the traffic using TCP is a persistant (like a large file transfer). Note that it is possible for TCP to keep its window open for a while and not send data. In the

worst case, if a number of TCP sources keep increasing their TCP windows slowly (during underload), and then synchronize to send data, the queue seen at the switch is the sum of the TCP windows [VAND98].

Variation in ABR demand and capacity affects the feedback given by the switch algorithm. If the switch algorithm is highly sensitive to variation, the switch queues may never be bounded since, on the average, the rates are never controlled. The buffer requirement above assumes that the switch algorithm can tolerate variation in ABR capacity and demand.

Also, in the above formula, it is assumed that the product of the number of active TCP sources times the maximum segment size (MSS) is small compared to the buffer requirement derived. Also note that the buffer requirement is for the ATM switches only. In other words, the queues are pushed by ABR to the edges of the network, and the edge routers need to use other mechanisms to manage the edge queues, which are of the order of UBR queues.

Note also that, under certain extreme conditions (like large RTT of satellite networks) some of the factors (RTT, feedback delay, averaging interval) may dominate over the others (eg: the feedback delay over the round trip time in satellite networks). Another scenario is a LAN where the averaging interval dominates over both RTT and feedback delay. The round trip time for a ABR segment (delimited by VS/VD switches) is twice the maximum one-way delay within the segment, and not the end-to-end delay of any ABR connection passing through the segment. These factors further reduce the buffer requirements in LAN switches interfacing to large networks, or LAN switches that have connections passing through segmented WANs.

Effect of two-way traffic: The above analysis has assumed undirectional TCP traffic (typical of file-transfer applications). We will briefly study the effect of two-way traffic on the buffer requirements. It has been noted that bidirectional traffic complicated TCP dynamics considerably leading to more bursty behavior by TCP. This is called the "Ack Compression'' phenomenon.

Effect of VBR background: The presence of higher priority background traffic implies that the ABR capacity is variable. There are two implications of the variation in capacity: a) the effect on the rate of TCP acks and the window growth, and, b) the effect on the switch rate allocation

algorithm. The VBR ON-OFF times, the feedback delays, and a switch scheme sensitive to variation in ABR load and capacity may combine to create worst case conditions where the ABR queues diverge. However, a scheme that combines accurate measurement with efficient measurement techniques can counter the effects of ON-OFF as well as self-similar VBR background traffic.

The complexity of two-way traffic VBR traffic require a buffer of at least 5×RTT. Note that the effect of the averaging interval parameter dominates in LANs (because it is much larger than RTT or feedback delay). Similarly, the effect of the feedback delay dominates in satellite networks because it can be much smaller than RTT.

Though the maximum ABR network queues are small, the queues at the sources are high. Specifically, the maximum sum of the queues in the source and the switches is equal to the sum of the TCP window sizes of all TCP connections. In other words the buffering requirement for ABR becomes the same as that for UBR if we consider the source queues into consideration. This observation is true only in certain ABR networks. If the ATM ABR network is an end-to-end network, the source end systems can directly flow control the TCP sources. In such a case, the TCP will do a blocking send, i.e., and the data will go out of the TCP machine's local disk to the ABR source's buffers only when there is sufficient space in the buffers. The ABR service may also be offered at the backbone networks, i.e., between two routers. In these cases, the ABR source cannot directly flow control the TCP sources. The ABR flow control moves the queues from the network to the sources. If the queues overflow at the source, TCP throughput will degrade.

Bursty Traffic: Note that the above results apply to the case of infinite traffic (like a large file transfer application) on top of TCP. [VAND98] shows that bursty (idle/active) applications on TCP can potentially result in unbounded queues. However, in practice, a well-designed ABR system can scale well to support a large number of applications like bursty WWW sources running over TCP.

Buffer requirements for zero TCP loss over ABR are small.

8.4.4 TCP over ABR: Switch Design Issues

Some of problems observed by common switch algorithms are discussed below:

- **Out-of-phase effect**: No load or sources are seen in the forward direction while sources and RM cells are seen in the reverse direction.

- **Clustering effect**: The cells from TCP connections typically come in clusters. Hence, the activity of multiple connections is difficult to sense over small averaging intervals, though the corresponding load may be high.

- **Variation in load**: Even an infinite traffic source running on top of TCP looks like a bursty source at the ATM layer. When a number of such sources aggregate, the load experienced at the switch can be highly variant. In such cases, it is possible to have a long period of underload, followed by a sudden burst, which builds queues. As a result the maximum queue may be large even though the utilization/throughput is low. Schemes like ERICA can track the variation in load and filter it, because they use the average load as a metric. However, several schemes use the queue length metric exclusively. Queue length has a higher variation than the average load, and it also varies depending upon the available capacity. Further, a queue length of zero yields little information about the utilization of the link. It has been argued that schemes which look at only the queue length are less susceptible to errors than schemes which use several metrics (like input rate, MACR, number of active sources etc). But, the use of several independent metrics gives more complete information about the system [JAIN91], and variation reduction can be done by using simple averaging techniques.

- **Variation in capacity**: The ABR capacity depends upon the link bandwidth, and the bandwidth usage of the higher priority classes like CBR and VBR, and can exhibit variation accordingly. The effect of ABR capacity variation, when combined with the latency in giving feedback to sources, results in an alternating series of high and low rate allocations by the switch. If the average total allocation exceeds the average capacity, this could result in unbounded queueing delays.

These effects reduce as the network path gets completely filled by TCP traffic, and the ABR closed loop control becomes effective. The switch scheme then controls the rate of the sources. Note that averaging techniques can be used to specifically to counter such conditions, i.e., reduce the error in measurement and handle boundary cases. The residual error even after these modifications manifests as queues at the bottleneck.

A good ABR switch algorithm is needed to counter the effects of variation in load, variation in capacity, out-of-phase effect and clustering effect.

The ERICA+ switch algorithm has been designed to provide good performance in these situations.

ues at the
vided has
also been studied.

ABR sources require one receiver window's worth of buffering per VC to avoid cell loss. The total buffering required for N sources is the sum of the N receiver windows. Note that this is the same as the switch buffer requirement for UBR. In other words, the ABR and UBR services differ in whether the sum of the receiver windows' worth of queues is seen at the source or at the switch.

If the ABR service is used end-to-end, then the TCP source and destination are directly connected to the ATM network. The source can directly flow-control the TCP source. As a result, the TCP data stays in the disk and is not queued in the end-system buffers. In such cases, the end-system need not allocate large buffers. In these end-to-end configurations, ABR allows TCP to scale well.

However, if the ABR service is used on a backbone ATM network (this would be typical of most initial deployments of ABR), the end-systems are edge routers that are not directly connected to TCP sources. These edge routers may not be able to flow control the TCP sources except by dropping cells. To avoid cell loss, these routers need to provide one receiver window's worth of buffering per TCP connection. The buffering is independent of whether the TCP connections are multiplexed over a smaller number of VCs or they have a VC per connection. For UBR, these buffers need to be provided inside the ATM network, while for ABR they need to be provided at the edge router. If there are insufficient buffers, cell loss occurs and TCP performance degrades.

The fact that the ABR service pushes the congestion to the edges of the ATM network while UBR service pushes it inside is an important benefit of ABR for service providers.

Key results in TCP performance over ABR are listed below:

Buffer requirements for TCP at the edge of ABR networks are comparable to UBR buffer requirements.

- *When maximum throughput is achieved, the TCP sources are rate-limited by ABR rather than window-limited by TCP.*
- *When the number of buffers is smaller, there can be a large reduction in throughput even though CLR is very small.*
- *The reduction in throughput is due to loss of time during timeouts (large timer granularity), and transmission of duplicate packets that are dropped at the destination.*
- *When throughput is reduced, the TCP sources are window-limited by TCP rather than rate-limited by ABR.*
- *Switch buffers should not be dimensioned based on the ABR Source parameter TBE. Dimensioning should be based upon the performance of the switch algorithm, and the round trip time.*
- *When ABR capacity is varied, CLR exhibits high variance and is not related to TCP throughput. In general, CLR is not a good indicator of TCP level performance.*
- *Larger buffers increase TCP throughput.*
- *Larger number of window-limited sources increase TCP throughput. This is because, the sum of the windows is larger when there are more sources.*
- *Even when the buffers are small, dropping of EOM cells should be avoided. This avoids merging of packets at the destination AAL5 and improves fairness. When sufficient buffers are provided for ABR, the network drop policy is important mainly at the edge of the ATM network.*

8.5 Virtual Source / Virtual Destination

In long latency satellite configurations, the feedback delay is the dominant factor (over round trip time) in determining the maximum queue length. A feedback delay of 10 ms corresponds to about 3670 cells of queue for TCP over ERICA, while a feedback delay 550 ms corresponds to 201850 cells. This indicates that satellite switches need to provide at least one feedback delay worth of buffering to avoid loss on these high delay paths. A point to consider is that these large queues should not be seen in downstream workgroup or WAN switches, because they will not provide so much buffering. Satellite switches can isolate downstream switches from such large queues by implementing the virtual source/virtual destination (VS/VD) option.

[GOYAL98a] have examined some basic issues in designing VS/VD feedback control mechanisms. VS/VD can effectively isolate nodes in different VS/VD loops. As a result, the buffer requirements of a node are bounded by the feedback delay-bandwidth product of the upstream VS/VD loop. However, improper design of VS/VD rate allocation schemes can result in an unstable condition where the switch queues do not drain.

The paper also presents a per-VC rate allocation mechanism for VS/VD switches based on ERICA+. This scheme retains the basic properties of ERICA+ (max-min fairness, high link utilization, and controlled queues), and isolates VS/VD control loops thus limiting the buffer requirements in each loop. The scheme has been tested for infinite ABR and persistent TCP sources.

VS/VD, when implemented correctly, helps in reducing the buffer requirements of terrestrial switches that are connected to satellite gateways. Without VS/VD, terrestrial switches that are a bottleneck, might have to buffer cells of upto the feedback delay-bandwidth product of the entire control loop (including the satellite hop). With a VS/VD loop between the satellite and the terrestrial switch, the queue accumulation due to the satellite feedback delay is confined to the satellite switch. The terrestrial switch only buffers cells that are accumulated due to the feedback delay of the terrestrial link to the satellite switch.

ABR Virtual Source / Virtual Destination can be used to isolate terrestrial networks from the effects of long latency satellite networks.

References

[AGNE95**]** Agnelli, Stefano and Mosca, Paolo, “Transmission of Framed ATM Cell Streams Over Satellite: A Field Experiment”, IEEE International Conference on Communications, v 3 1995, IEEE, Piscataway, NJ

[AKYL97] Ian F. Akyildiz, Seong-Ho Jeong, “Satellite ATM Networks: A Survey,” IEEE Communications Magazine, July 1997, Vol 5.35. No. 7.

[BASAK97] Debashis Basak, Surya Pappu, "GFR Implementation Alternatives with Fair Buffer Allocation Schemes," ATM Forum 97-0528, May 1997.

[BONAV97] Olivier Bonaventure, "A simulation study of TCP with the proposed GFR service category," DAGSTUHL Seminar 9725, High Performance Networks for Multimedia Applications, June 1997, Germany.

[CHIT94**]** Chitre, D M., Gokhale, D S. Henderson, T. Lunsford, J L. Mathews, N., Asynchronous Transfer Mode (ATM) Operation Via Satellite: Issues, Challenges and Resolutions, International Journal of Satellite Communications. v 12 n 3 May-June 1994

[CLAR81**]** Clark, G. C., and J. B. Cain, Error-Correction Coding or Digital Communications, Plenum Publishing Corporation, New York, 1981.

[CUE95a**]** Cuevas, E. G., Satellite link performance characterization of 2Mb/s IDR channels with Reed-Solomon codec, IEE Conference Publication, n 403/2 1995, IEE, Stevenage, Engl.

[CUE95b**]** Cuevas, E G. Doshi, B. Dravida, S., Performance models for ATM applications over 45 Mb/s satellite facilities, IEE Conference Publication. n 403/1 1995. IEE, Stevenage, Engl.

[ERIC97] S. Kalyanaraman, R. Jain, et. al, “The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks,” Submitted to IEEE/ACM Transactions on Networking, November 1997, http://www.cis.ohio-state.edu/~jain/papers/erica.htm

[FAHMY96**]** Sonia Fahmy, Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal and Fang Lu, “On Source Rules for ABR Service on ATM Networks with Satellite Links,” Proceedings of the First International Workshop on Satellite-based Information Services, November 1996.

[FALL96] Kevin Fall, Sally Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," Computer Communications Review, July 1996.

[FENG] Wu-chang Feng, Dilip Kandlur, Debanjan Saha, Kang G. Shin, "Techniques for Eliminating Packet Loss in Congested TCP/IP Networks," _____.

[FLOYD93] Sally Floyd, Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transaction on Networking*, August 1993.

[GOYAL97a] Rohit Goyal, Raj Jain, Sastri Kota et.al., “Selective Acknowledgments and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks,” Proceedings of IC3N'97, September 1997.

[GOYAL98a] Rohit Goyal, Raj Jain et. al., “Per-VC rate allocation techniques for ABR feedback in VS/VD networks,” Submitted to Globecom'98.

[GOYAL98b] Rohit Goyal, Raj Jain, et. al., "Improving the Performance of TCP over the ATM-UBR Service," To appear in *Computer Communications*, 1998.

[GOYAL98c] Rohit Goyal, Sastri Kota, Raj Jain, et. al, "Analysis and Simulation of Delay and Buffer Requirements of Satellite-ATM Networks for TCP/IP Traffic," Submitted to *Journal of Selected Areas in Communications*, March 1998.

[GOYAL98d] Rohit Goyal, Raj Jain, et. al, "Providing Rate Guarantees to TCP over the ATM GFR Service," Submitted to LCN98, 1998.

[HEIN] Juha Heinanen, Kalevi Kilkki, "A Fair Buffer Allocation Scheme," Unpublished Manuscript.

[HOE96] Janey C. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP,'' Proceedings of SIGCOMM'96, August 1996.

[I35696] ITU-T Recommendation I-356, "B-ISDN ATM Layer Cell Transfer Performance," Geneva, October, 1996.

[IESS308**]** IESS-308, "Performance Characteristics for Intermediate Data Range (IDR) Digital Carriers," Appendix F: "Concatenation of Reed Solomon (RS) Outer Coding with the Existing FEC." IESS-308 (Rev. 6B), December 4, 1992.

[IQTC97] David Lucantoni, Patrick Reilly, "Supporting ATM on a Low-Earth Orbit Satellite System," http://www.isoquantic.com

[IT4B97] ITU-R 4B Preliminary Draft Recommendation, "Transmission of Asynchronous Transfer Mode (ATM) Traffic via Satellite," Geneva, January 1997.

[JACOBS88] V. Jacobson, "Congestion Avoidance and Control," Proceedings of the SIGCOMM'88 Symposium, pp. 314-32, August 1988.

[JAIN91] Raj Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, Wiley-Interscience, New York, NY, April 1991.

[JAIN95**]** Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy and Fang Lu, "Parameter Values for Satellite Links," ATM Forum Trafic Management 95-0972, August 1995.

[JAIN96**]** Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy and Rohit Goyal, "Source Behavior for ATM ABR Traffic Management: An Explanation", IEEE Communications Magazine, November 1996.

[KALYAN97a] Shivkumar Kalyanaraman, R. Jain, et. al., "Performance of TCP over ABR with self-similar VBR video background traffic over terrestrial and satellite ATM networks," ATM Forum 97-0177r2, April 1997.

[KALYAN97b] Shivkumar Kalyanaraman, "Traffic Management for the Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks," PhD Dissertation, The Ohio State University, 1997.

[KALYAN98a] Shivkumar Kalyanaraman, R. Jain, et. al., “Performance and Buffering Requirements of Internet Protocols over ATM ABR and UBR Services,” To appear, IEEE Computer Communications Magazine.

[KALYAN98b] Shivkumar Kalyanaraman, R. Jain, et. al., "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks," Submitted to IEEE/ACM Transactions on Networking, November 1997.

[KOTA97b] Sastri Kota, Jerry Kallaus, Henry Huey, David Lucantoni, “Demand Assignment Multiple Access (DAMA) For Multimedia Services – Performance Results,” Proceedings of Milcom’97, Monterey, CA, 1997.

[KOTA97] Sastri Kota, R. Goyal, Raj Jain, “Satellite ATM Network Architectural Considerations and TCP/IP Performance,” Proceedings of the 3rd K-A Band Utilization Conference, 1997.

[LI96] H. Li, K.Y. Siu, H.T. Tzeng, C. Ikeda and H. Suzuki, "TCP over ABR and UBR Services in ATM," Proc. IPCCC'96, March 1996.

[LIN97] Dong Lin, Robert Morris, "Dynamics of Random Early Detection," Proceedings of SIGCOMM97, 1997.

[LUNS95**]** Lunsford, J. Narayanaswamy, S. Chitre, D. Neibert, M , Link Enhancement for ATM Over Satellite Links, IEE Conference Publication. n 403/1 1995. IEE, Stevenage, Engl.

[MATHIS96] M. Mathis, J. Madhavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgment Options," Internet RFC 2018, October 1996.

[PHAM97] C. Pham, S. Kota, R. Gobbi, “Performance of Concatenated Coding for Satellite ATM Networks,” Document in preparation.

[PONZ97] C. Ponzoni, “High Data Rate On-board Switching,” 3rd Ka-band Utilization Conference, September 13-18, 1997.

[RAMS95**]** Ramseier, Stefan. and Kaltenschnee, Thomas, “ATM Over Satellite: Analysis of ATM QOS Parameters”, IEEE International Conference on Communications. v 3 1995. IEEE, Piscataway, NJ

[RF2018] M. Mathis, J. Madhavi, S. Floyd, A. Romanov, “TCP Selective Acknowledgment Options,” Internet RFC 2018, October 1996.

[ROMANOV95] Allyn Romanov, Sally Floyd, ``Dynamics of TCP Traffic over ATM Networks," *IEEE Journal of Selected Areas In Telecommunications*, May 1995.

[SIU97] Kai-Yeung Siu, Yuan Wu, Wenge Ren, "Virtual Queuing Techniques for UBR+ Service in ATM with Fair Access and Minimum Bandwidth Guarantee," Proceedings of Globecom'97, 1997.

[STAL98] W. Stallings, *High-Speed Networks. TCP/IP and ATM Design Principles*, Prentice Hall, New Jersey, 1998.

[STEV97] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," Internet RFC 2001, January 1997.

[TCPS98] Mark Allman, Dan Glover, “Enhancing TCP Over Satellite Channels using Standard Mechanisms,” IETF draft, February 1998, http://tcpsat.lerc.nasa.gov/tcpsat

[TM4096**]** “The ATM Forum Traffic Management Specification Version 4.0,” ATM Forum Trafic Management AF-TM-0056.000, April 1996.

[WU97] Yuan Wu, Kai-Yeung Siu, Wenge Ren, "Improved Virtual Queuing and Dynamic EPD Techniques for TCP over ATM," Proceedings of ICNP97, 1997.

[WUPI94] William Wu, Edward Miller, Wilbur Pritchard, Raymond Pickholtz, “Mobile Satellite Communications,” Proceedings of the IEEE, Vol. 82, No. 9, September 1994.

[VAND98] Bobby Vandalore, Shiv Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy, "Worst Case Buffer Requirements for TCP over ABR", SICON'98, June 1998.