Internet Draft
Point-to-Point Protocol Extensions WG

Expire in six months

J. Manchester
M. Krishnaswamy
S. Dravida
J. Anderson
B. Doshi
E. Hernandez-Valencia
Lucent Technologies

W. L. Edwards
Sprint Corporation

B. Bharucha
K. Fendick
G. Wetzel
AT&T

PPP over SONET/SDH

<draft-ietf-pppext-pppsonet-scrambler-00.txt>

Status of this Memo

Abstract

This Internet Draft addresses the PPP/SONET interface defined in the IETF RFC 1619 and its application in public Internet backbone networks. By experimental analysis it is found that this SONET interface specification is deficient in providing payload transparency. This

deficiency leads to deleterious effects in providing reliable network operations. A simple SONET payload scrambler enhancement is proposed

to overcome this deficiency. This work has been brought to the ANSI T1X1.5 as an action item for the addition of the RFC 1619 mapping to T1.105 with a scrambler to be determined by T1X1.

We are submitting the same proposal to IETF now with the interest to enhance RFC-1619 accordingly and achieve alignment with ANSI T1 standards. To facilitate this alignment, we have recommended that T1X1 establish a formal liaison with IETF in regard to IP/SONET interface standards and associated IP transmission standards development.

1. Introduction

SONET provides a reliable high speed transmission path for today's growing Internet backbone traffic. RFC 1619 addresses the issues concerning mapping of the PPP packets on to SONET payload envelopes [1], [2]. However no scrambling of the IP payload is specified in RFC 1619. We have found by experimental analysis that this leads to lack of payload transparency that could cause serious damage to the reliable functioning of SONET networks.

We are proposing a scrambler for SONET payloads and use of SONET Path Overheads for maintaining synchronization in the scrambler operation. We have recommended to T1X1.5 that the scrambler enhanced mapping use a different Path Signal Label code (byte C2) from the original proposal in RFC 1619. This will allow network operators to distinguish between scrambled and non-scrambled payloads.

2. Native SONET Scrambler design

One very serious problem with the RFC specification is the assertion that "no scrambling is needed during insertion in the SPE". [2, page 2] It is evident that the decision to not scramble the HDLC delimited PPP packets is derived from errored assumptions about the SONET scrambler. The SONET scrambler was designed for optical transmission of digital signals. "SONET optical interfaces use binary line coding, and therefore must be scrambled to assure an adequate number of transitions (zeros to ones, and ones to zeros) for such purposes as line rate recovery at the receiver,"[3, page 5-6] and the suppression of discrete spectral components that could lower the receiver's signal-to-noise ratio. Use of the SONET scrambler was deemed sufficient for providing payload transparency for multiplexed payloads. However, in the case of non-multiplexed payloads, such as IP or ATM, where the user data occupies a significant portion of the SONET frame, use of the SONET scrambler does not provide sufficient payload transparency.

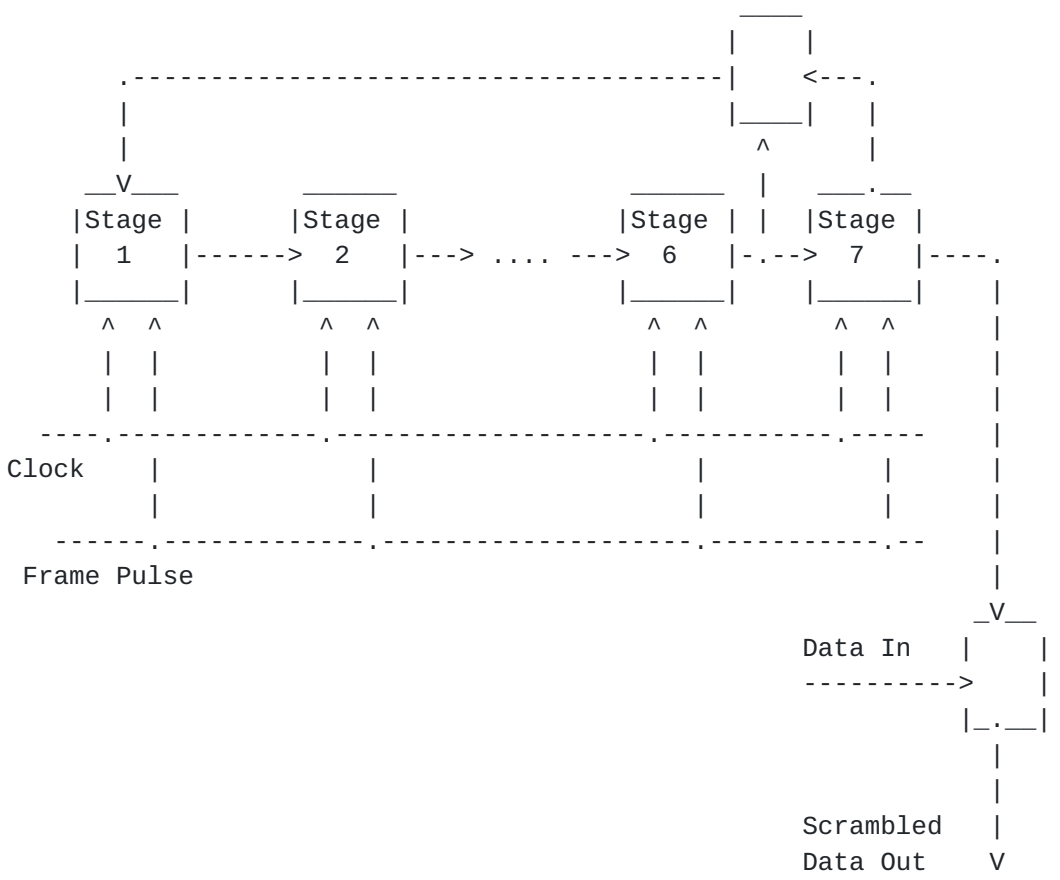


Fig. 1 SONET Scrambler (1 + x**6 + x**7)

To understand the implications of not having sufficient payload transparency, one must examine the SONET scrambler in more detail. The SONET scrambler is a set-reset frame synchronous scrambler with a generating polynomial of 1+x**6+x**7 as shown in Figure 1. The scrambler resets at each SONET frame by setting each of the registers to all ones on the most significant bit of the byte following the STS-1 number N J0/Z0 byte. The framing bytes, and the J0/Z0 bytes in STS-1 through STS-N are not scrambled. A series of shift registers are used with feedback taps coming off of the 6th and 7th registers. These taps are xored for input back into the 1st shift register. This operation produces a pseudo random sequence. As this is a 7th order scrambler, the pseudo random sequence generated repeats itself every (2**7)-1, or 127, bit periods. The pseudo random sequence coming out of the 7th register is xored with the data to be transmitted. This sequence from the 7th register is easily obtainable using a spread sheet application. Thus, a malicious user, armed with knowledge of the xor operation, can, by making a 1/127 assumption as to where his data lands in the SPE, control the output of the SONET

scrambler.

Suppose for example that a malicious user was trying to introduce
a long string of zeros into the public network. They could transmit

an IP datagram that continuously repeats the 127 bit pattern from the 7th register of the SONET scrambler. When the pattern from the 7th register is aligned with the 127 bit pattern from the malicious user, the scrambler will put out all zeros.
The malicious user has no idea where his datagram will land in the SPE. The probability of the repetitive codes in the first row being aligned with the 7th register of the SONET scrambler is 1/127. If the SONET signal is an STS-3c, there will be an 80 bit offset for the transmission of the SONET transport and path overhead. The malicious user will have no control over these fields; however, because 127 is prime and thus has no factors in common with 80, the probability of the repetitive codes matching the output of the 7th register is exactly 1/127 for each new row that the datagram is mapped into [Note 1]. If the assumption is further made, that the user is transmitting to the IP over SONET interfaces via an ethernet interface (which has an MTU of 1500 bytes), then on average, the malicious user only has to transmit 21 [Note 2] datagrams to be reasonably sure that a long string of zeros has been introduced into the network. This long string of zeros is, in the worst case, **2080 bits or 13 microsec for the STS-3c mapping.** This is well within the specification for SONET LOS (Loss Of Signal) and depending on the type of clock and clock recovery circuit may also cause framing and synchronization problems.

[Note 1]
If there were common factors between 127 and 80, the per row probability would be lower because there will be spots in one row whereby if an user's datagram lands in them, it will be impossible for the repetitive codes to be on in the next row. The 1/127 probability actually increases if the Path Overhead starts to float. The path overhead then acts as a second offset and the codes are twice as likely to be on for each row; however, the number of zeroes that can be introduced will be reduced.

[Note 2]
1500 bytes is roughly 6 rows for an STS-3c mapping. Each datagram thus has a 6/127 chance of introducing a long string of zeros. The transmission of 21 datagrams will thus lead to a probability of (6/127)*21 ~= 1.

3. Experimental verification of RFC 1619 inadequacy

In the laboratory we tested out our technical hypotheses using RFC 1619 compliant interfaces and several SONET transport network test equipments. The following is a summary of our conclusions:

A. SONET interfaces that detect LOS in less than 13 microsec are open to a malicious user causing LOS when interconnected to an RFC 1619 compliant interface [Note 3]. Note that the LOS specification is 2.3 to 100 microsec and it is our experience that most SONET interfaces are on the

low end of this detection time as this value is included in the SONET restoration time for automatic protection switching.

B. All SONET interfaces regardless of LOS detection are open to a malicious user causing synchronization, clock, and framing problems when the interface is connected to an RFC 1619 compliant interface. There are no requirements for how long a clock recovery circuit must maintain synchronization, but our experience tells us that most SONET clock recovery circuits are designed to stay in synchronization for ~80 bit periods when there are no bit transitions on the incoming line. After that, the clock will go into holdover and the ability of the clock to maintain synchronization will be dependent on the clock quality (i.e., stratum level).

C. As a result of the above scenario, it is possible for a malicious user to force an interface connected to an RFC 1619 compliant interface to detect a hard failure based on the onset of LOS or LOF (Loss Of Frame). Thus, RFC 1619 compliant interfaces should never be used to provide enhanced services such as protection switching.

It is also important to realize that the issue of unraveling the SONET scrambler came up during the standardization of ATM over SONET interfaces. There a user could only gain 48 bytes of the SONET SPE before there is an interruption from the ATM cell overhead. This was still seen as a problem when analyzed theoretically. Laboratory tests could not be performed at the time because SONET and ATM equipment did not exist. It is therefore simply fair to point out, at the outset, what this work owes to certain contemporary ATM and SONET interface developers. We wish to provide citations to this early outstanding technical work here: [4] [5] [6] [7] [8] [9]. As a result of these contributions, "cell payload scrambling is used to provide security against payload information replicating the frame synchronous scrambling sequence (or its inverse) used at the SONET section layer." [3, page 3-61].

[Note 3]
The malicious datagrams can be transmitted from anywhere in the Internet. The user need not be directly connected to the SONET network.

4. Requirements for SONET Mappings

In November 1988, T1X1 agreed to the following set of mapping guidelines [10]:

Standardized Payload
Significant Network Advantage OR Unique

Payload Transparency for Non-terminated Payloads
Timing Transparency
Minimal Implementation Complexity

Floating/Locked Translation Capability
Mid-Span Meet

A detailed explanation of each of the mapping guidelines from the original T1X1.5 Mapping SWG contribution [10] follows:

Standardized Payload

The structure of any payload to be mapped must be specified in detail and approved by a recognized standard setting organization.

Unique or Significant Network Advantage

The payload must not already have a defined mapping (unique) or the new mapping must have a significant network advantage over the original mapping. Significant NETWORK advantage implies that the assessment of 'significant advantage' takes into consideration the fact that the STS format was developed to allow direct interfaces on many types of transport equipment which are interconnected. Therefore, a mapping which optimizes a particular piece of equipment or application at the expense of the network as a whole is precluded.

Payload Transparency For Non-Terminated Payloads

VT and STS Synchronous Payload Envelopes were developed to allow the transport of payloads by equipment which has no 'knowledge' of the type of payload, and to allow new payloads to be mapped and transported without modification to deployed equipment. New mappings should not compromise this capability.

Timing Transparency

VT and STS Payload Pointers were developed to allow the transport of payloads between equipment which is not phase-aligned nor necessarily exactly frequency synchronous. New mappings should not compromise this capability.

Minimal Transport Delay

VT and STS Payload Pointers also allow the manipulation of Synchronous Payload Envelopes without frame alignment buffers in order to minimize transport delay. New mappings should not compromise this capability.

Minimal Implementation Complexity

The circuitry required to implement a particular mapping should not be unduly complicated.

Performance

The mapping should be such that, after being transported by the
SONET network, the payload can meet all network performance

criteria, such as jitter, specific to that payload.
Floating/Locked Translation Capability

Locked VT Mode mappings must be such that it is possible to translate an STS-1 between Locked and Floating VT Mode. This translation function should meet all other applicable guidelines.

Mid-Span Meet

This mapping should not compromise the capability of providing mid-span meet for that payload. Although a payload that is mapped using Floating VTs is not compatible with the same payload mapped using Lock VTs, this exception is covered by the Floating/Locked Translation Capability.

5. Analysis of RFC 1619 with Respect to Mapping Guidelines

In examining the RFC 1619 PPP/SONET mapping with respect to the T1X1 mapping guidelines, it is clear from the discussion in Section **2 and 3 that the mapping does not provide payload transparency for** non-terminated payloads. This is a very serious problem and cannot be overlooked. To rectify this situation immediately, it has been proposed in T1X1 to enhance the PPP over SONET mapping with the use of a scrambler. (See Appendix A for the scrambler proposed to T1X1)

6. Scrambler Requirements

In the previous sections we demonstrated the need for a scrambler. Scramblers designed to transport high speed data services must meet the following requirements:

A. Robust Performance: The scrambler design must be robust against malicious user attacks. Even if the user gains control of one or more SONET/SDH frames, the user should not be able to negate the scrambler so that bit transparency is lost. This can be accomplished by making the probability of guessing the scrambler state as small as possible.

B. Quick Recovery: The scrambler design should be such that in the event of abnormal situations such as the loss and regaining of physical layer synchronization, the descrambler at the receiver and the scrambler at the transmitter regain synchronization is as short a time as possible.

C. No Error Multiplication: Error multiplication at the lower layer can be damaging to the higher layer applications. If there is error multiplication, certain error protection schemes such as parity checks could be rendered useless. Arithmetic check sums are also generally vulnerable to error multiplication and perform poorly. Even CRC checks

designed at the higher layers to provide a certain minimum Hamming distance begin to lose their power and the effective minimum Hamming distance provided in the presence of error multiplication is reduced.

D. Immunity to Transmission Errors: While quick recovery is desirable
PPP over SONET/SDH October 1997

after abnormal conditions such as loss of signal events, it is also important that the scrambler have immunity to normal conditions such as random background errors. The random errors should not cause the descrambler and scrambler to go out of synchronization.

E. Work within existing Frame Structure: The scrambler should be made to work within the existing SONET/SDH frame structure. There should be no need to introduce new overheads in the SONET/SDH payload.

7. Proposed Scrambler Design

See Appendix A for the scrambler proposed to T1X1.

8. Proposal made to T1X1

This work addressed the PPP/SONET interface defined in the IETF RFC 1619 and its application in public Internet backbone networks. Based on the analysis presented herein, the following proposals were made in T1X1:

A. Reaffirm the SONET mapping criteria and requirements for T1.105 [11] outlined in [10].

B. Enhance the PPP/SONET mapping in [2] with use of a scrambler for the SONET payload and use of Path Overheads for maintaining synchronization in the scrambler operation.

C. Specify the scrambler and the Path Signal Label code. The Path Signal Label code should be different from the original proposal in RFC 1619 for allowing network operators to distinguish between scrambled and non-scrambled payloads.

9. Proposal to IETF

Specifically we propose that future IETF specifications of PPP over SONET/SDH be aligned with T1.105 when this work is completed in T1.

In the interim, we recommend that one of the following is done:

A. Retract RFC 1619 (or)

B. Reissue a new PPP over SONET/SDH RFC with a cautionary note that the failure to scramble the PPP packets can lead to deleterious effects in providing reliable network operations.

10. Security Consideration

This memo is informational, and specifies no protocol for deployment. It highlights specific security vulnerabilities of RFC 1619.

11. Acknowledgments

We would like to thank Tom Bowmaster from Bellcore for his technical and testing assistance.

Grenville Armitage assisted in presenting this information in the IETF Internet Draft format.

12. References

[1] IETF RFC 1661, "The Point-to-Point Protocol (PPP)," W. Simpson - Daydreamer (July 1994).

[2] IETF RFC 1619, "PPP over SONET/SDH," W. Simpson - Daydreamer (May 1994).

[3] Bellcore GR-253-CORE, Issue 2, (December 1995).

[4] T1S1.1/88-498, "Enhancements of SONET Scrambling Capabilities to Carry ATM Cells," Kuo-Hui Liu and William L. Edwards - Pacific Bell (October 1988).

[5] T1S1.1/88-453, "Bit Sequence Independence for ATM Cells," Ralph Ballart - Bellcore (October 1988).

[6] T1X1.5/89-007, "Further Analysis on Bit Transparency Issue," William L. Edwards and Kuo-Hui Liu - Pacific Bell (February 1989).

[7] T1S1.1/89-527, "Distributed Bit Scrambling Method for ATM Cells", **D. Fisher and S. Brueckheimer - STC Technology Limited (September 1989).**

[8] T1S1.1/89-222, "Killer Cells Detector and Solution for Bit Transparency Issue," William L. Edwards and Kuo-Hui Liu - Pacific Bell (May 1989).

[9] T1S1.1/89-163, "Proposed Contribution to CCITT SG XVIII on SDH Mapping for ATM," Jon Anderson - AT&T Bell Laboratories (May 1989).

[10] T1X1.5/88-123, "Payload Mapping Guidelines," Brent Allen - Nortel (November 1988).

[11] ANSI T1.105, "Synchronous Optical Network (SONET) Basic Description including Multiplex Structure, Rates and Formats" (1995).

[12] ITU-T Recommendation G.707, "Network node interface for the synchronous digital hierarchy (SDH)" (3/96).

12. Authors' Address

James Manchester
E-mail: manchester@bell-labs.com
Telephone: +1-732-949-6284
Fax: +1-732-949-3210

Bell Laboratories
Room 2G-527
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Murali Krishnaswamy
E-mail: murali@bell-labs.com
Telephone: +1-732-949-3611
Fax: +1-732-949-3210

Bell Laboratories
Room 2G-527A
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Subrahmanyam Dravida
E-mail: dravida@bell-labs.com
Telephone: +1-732-949-1264
Fax: +1-732-834-5906

Bell Laboratories
Room 3M-337
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Jon Anderson
E-mail: jonanderson@bell-labs.com
Telephone: +1-732-949-0634
Fax: +1-732-949-3210

Bell Laboratories
Room 2G-538
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Bharat Tarachand Doshi
E-mail: bdoshi@bell-labs.com
Telephone: +1-732-949-0823
Fax: +1-732-834-5906

Bell Laboratories
Room 3L-337

101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Enrique Hernandez-Valencia
E-mail: enrique@bell-labs.com
Telephone: +1-732-949-6153
Fax: +1-732-834-5906

Bell Laboratories
Room 3H-313
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

W. L. Edwards
E-mail: texas@sprintcorp.com
Telephone: +1-913-534-5334
Fax: +1-913-534-2526

Sprint Corporation
Mailstop KSOPKB0802
9300 Metcalf Avenue
Overland Park, KS 66212

Behram Bharucha
E-mail: bbharucha@att.com
Telephone: +1-732-949-7989
Fax: +1-732-949-8569

AT&T
Room 1J-301
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Kerry Fendick
E-mail: kfendick@att.com
Telephone: +1-732-949-1243
Fax: +1-732-949-1726

AT&T
Room 1L-425
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Greg Wetzel
E-mail: gfwetzel@att.com
Telephone: +1-732-949-6630

Fax: +1-732-949-1726

AT&T
Room 1L-426
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

Appendix A

Note:
This is a proposal to T1X1 and should not be considered to be
in final form.

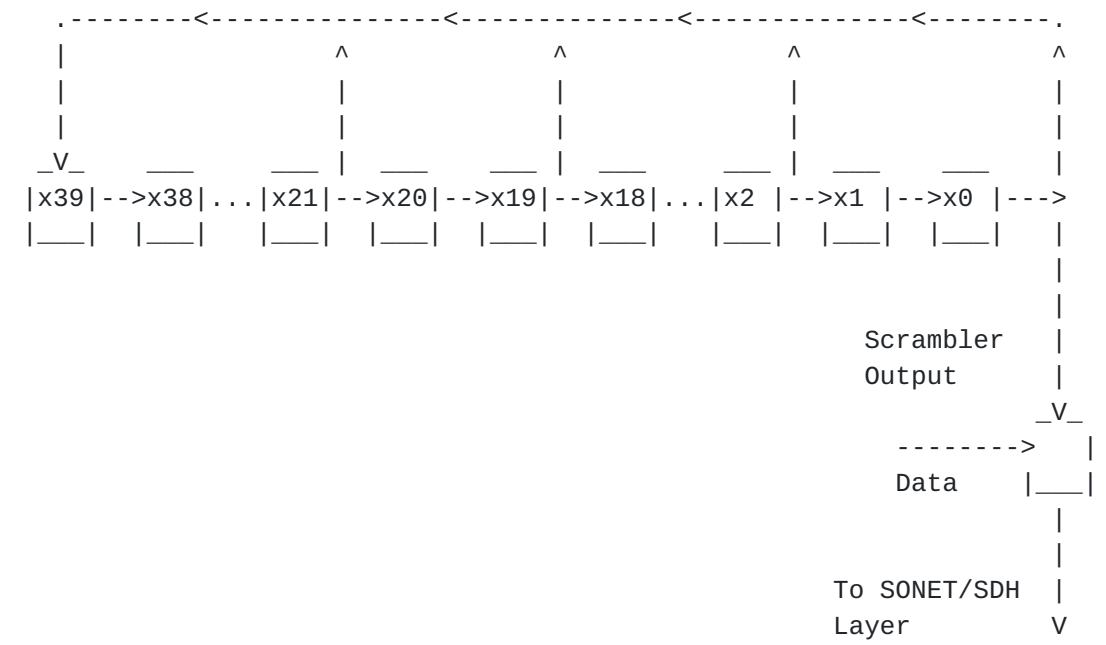


Fig. 2 Proposed Scrambler for PPP over SONET

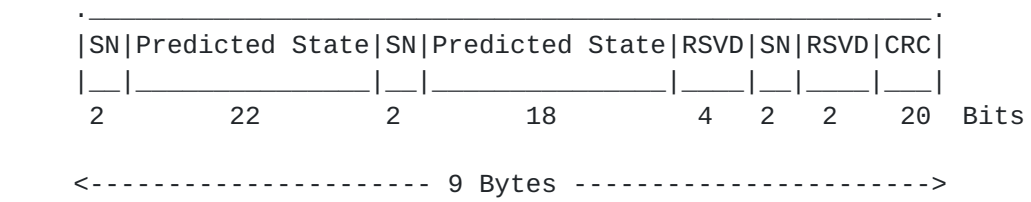
Based on section 6 requirements, a scrambler polynomial has been
of degree 40 has been designed. This scrambler generates
a pseudo-random sequence of period (2**40)-1, that is it repeats
after (10**12) bits. This period is sufficiently long even for
rates up to OC-768. The probability of a malicious user locking
on to the phase of this scrambler is (2 ** -40) or (10 ** -12).
In conjunction with the SONET/SDH set-reset scrambler, the overall
probability of a malicious user causing loss of bit transparency is
(10 ** -14). The scrambler polynomial is
1 + x2 + x**19 + x**21 + x**40 and its shift register implementation**
is shown in Figure 2.

A. Transmitter and Receiver Synchronization.

The proposed scrambler is designed to scramble bits in the SONET/SDH Payload envelope only. The section, line and path overheads are not

scrambled by this scrambler. Therefore the scrambler state is transmitted using bytes available in the Path Overhead (POH). Currently H4, Z3 and Z4 bytes are available. These bytes can be used to transmit the scrambler state so that the descrambler at the receiver can be synchronized with the scrambler at the transmitter.

Since the scrambler state is 40 bits long, 5 bytes are needed to transmit the scrambler state. This can be done by splitting the scrambler state and carrying it in the H4, Z3 and Z4 bytes of multiple frames. The state transmitted by the transmitter should be such that upon its reception, the receiver could load that state into its descrambler and immediately start descrambling. In order to enable this, it is desirable for the transmitter to predict the scrambler state needed at the receiver for immediate descrambling and then place the predicted state in the H4, Z3 and Z4 bytes of the Path Overhead (POH). Since the state is transmitted in multiple frames, the scrambler state would have to be predicted across the number of frames that it takes to transmit the scrambler state. For example, if the scrambler state is transmitted in two successive frames then the scrambler state needs to be predicted by at most two frames. However, the prediction interval is pre-determined and all that is needed is prediction of the current state by a fixed number of bytes. Scrambler state prediction is very simple and can be accomplished in many ways. A straightforward method is to run two scramblers, one at the current state and the second one ahead by the number of bytes in the prediction interval. Another elegant and fast method is to perform prediction using table look-ups. This avoids the need for a serial bit register implementation. Five tables each of 1.2 Kbytes can be used for state prediction. Since the prediction interval is known, the tables are precomputed and stored. The most significant byte (MSB) of the current state indexes the first table and retrieves a 5 byte word, the second most significant byte indexes the second table and retrieves another **5 byte word and so on.** The retrieved 5 byte words are xored to provide the predicted state.



In order to provide immunity from random errors and to provide the ability to quickly recover immediately after abnormal events such

as protection switching, are the predicted state be
covered by a CRC. Furthermore, to leave room for additional
functions that could be added in the future, the

H4, Z3 and Z4 bytes in some frames are reserved for future use. This can be done by introducing a frame sequence number of two bits. The H4, Z3 and Z4 bytes in three consecutive frames can be used for transmitting the predicting scrambler state and the H4, Z3 and Z4 bytes in the fourth frame could be used for some other purpose in the future. The proposed format of the predicted state is shown in Figure 3. In this figure SN stands for sequence number and RSVD is for reserved bits.

At initialization, the transmitter picks a random seed to load the shift registers of the scrambler. The only requirement is that at least one of the 40 bits be non-zero. The transmitter then picks the corresponding predicted state and forms the predicted state field as shown in Figure 3 and hands it down to the SONET/SDH physical layer. The SONET/SDH physical layer transports the predicted state in three consecutive frames. At the receiver, the CRC is checked and if it passes the predicted state is loaded into the descrambler and descrambling starts immediately.

Under normal conditions, the receiver checks to see if the CRC passes. If the CRC does not pass then the receiver ignores the predicted state. If the CRC passes and the predicted state does not match the current state at the receiver, then the receiver loads the predicted state into the descrambler and continues to descrambler. The chance of loading an errored state is 1 in (10**6).

In the worst case, it will take six frame times (750 microsec) to regain scrambler synchronization under abnormal conditions such as the loss and recovery of physical layer synchronization.