TECHNICAL REPORT

DSL Forum TR-034

Alternative OAM Communications Channel Across the U interface

May 2000

Abstract:
This document describes a method for an ATU-C to provision and retrieve remote information across the U interface. This method is proposed as an alternative to the optional method specified by ITU-T Recommendation G.997.1 (i.e. SNMP over clear eoc).

1 Introduction

A common set of objects are defined to manage an ADSL interface (see references 1 through 6). Included in this set of objects are a small number of ATU-R characteristics which can not be derived by the ATU-C using mechanisms provided by the physical layer (see references 7 through 9). ITU-T G.997.1 specifies the use of an optional OAM communications channel consisting of SNMP over clear eoc. In order to reduce complexity this document proposes a simpler alternative to SNMP over clear eoc as the basic mechanism to retrieve this information across the U interface. Since the proposal is based upon the use of the standard transaction oriented eoc provided by the physical layer, which is unidirectional (ATU-C is the master and ATU-R the slave), this mechanism only supports the ATU-C setting or getting ATU-R parameters.

2 Alternative OAM Channel

The physical layer interface specifications provide comprehensive eoc register read and write facilities. In addition two registers (registers 4 and 5) are reserved for vendor discretionary use. This proposal consists of an implementation agreement, which defines the use of one of these two registers. The other register remains available for vendor discretionary use. A mechanism is also defined which identifies the ATU-R as supporting this protocol. Alternatively the vendor ID, version number and serial number could be used to determine the capability of the ATU-R.
The remainder of this section describes the implementation of the Eoc Register Protocol (ERP).

2.1 Protocol

2.1.1 Transport of Messages

Use eoc read / write procedures and eoc register 5 as provided by the physical layer specifications (see reference 8 section 9.2.5.3 Message / data-response protocol state and section 9.2.4 Data registers in the ATU-R)[1] . As per the definition of the eoc, integrity of the message transfer is guaranteed by the eoc protocol.

2.1.2 Message Rate

The raw eoc channel rate may be up to approximately 15 kbps. Since each message is sent multiple times and acknowledged (multiple times) the message rate is reduced by a factor of 3 on an error-free line. In addition the read and write procedures are inherently half duplex (request – reply – request – reply, etc) so the rate further reduces by half. When operating in some reduced overhead modes available bandwidth is further decreased by a factor of two as eoc and aoc are sent on alternate frames. Note that the eoc rate is always independent of the line rate. The available message rate is therefore expected to be approximately 1 kbps.

2.1.3 Message set

Through eoc register 5 the following operations are supported :

Message	Value	Description
Identify	0	Used to determine if the ATU-R supports this method of information exchange.
Get	1	Used to retrieve ATU-R information.
Set	2	Used to send information to the ATU-R.
Reserved	3-255	reserved for future use

[1] There are corresponding sections in both reference 7 and 9 that describe identical functionality.

Each of these operations is implemented by a specific series of multi-byte eoc register read and write actions which when taken together comprise the atomic ERP operation. Each ERP operation must be complete before another ERP operation is begun. Should another operation begin before the current one is complete, then the first ERP operation is aborted. However, other eoc actions (that do not effect eoc register 5) can occur while an ERP operation is in progress.

The specific sequence for each operation follows:

2.1.3.1 Identify

The Identify operation determines if the ATU-R supports this method of information exchange. The following steps are required to perform this operation:

Step 1

EOC read register 5

If the response is UTC (unable to comply) then ERP is not supported
(optionally save the value of register 5)

Step 2

EOC Write register 5	Value
	byte 1
	0

If the response is UTC then ERP is not supported

Step 3

EOC read register 5

If the response is 0 then ERP is not supported
(optionally restore the value saved in step one to register 5 with an EOC register 5 write operation)

ERP is supported if the response to the read is as follows:
1st byte = 0x45 (E)
2nd byte = 0x52 (R)
3rd byte = 0x50 (P)
4th byte = 0x01 (version number)
This document specifies ERP version number 1 therefore the 4th byte should be coded as 01.

The Identify operation should be performed before any other ERP operation is performed to a particular ATU-R after the U interface initializes.

2.1.3.2 Get

The Get operation is used to retrieve ATU-R information from 'virtual registers'. A virtual register is a data structure associated with a particular element of ATU-R information that can be retrieved or modified by the ATU-C using the ERP. The virtual registers and their structure are defined in section 2.1.4.

Step 1

EOC Write register 5	Value	
	byte 1	byte 2

	1	virtual reg number

Step 2

EOC read register 5

returns

single byte or multi-byte return of data structure defined for this specific virtual register. If the virtual register is not implemented it is assumed that the ATU-R will return UTC to the eoc read request

Note that a **Get** of **virtual register 0** should result in the same response as the **Identify** operation.

2.1.3.3 Set

The Set operation is used to send information to 'virtual registers' in the ATU-R. A virtual register is a data structure associated with a particular element of ATU-R information that can be retrieved or modified by the ATU-C using the ERP. The virtual registers and their structure are defined in section 2.1.4.

Step 1

EOC Write register 5	Value					
	byte 1	byte 2				
	2	virtual reg number				
			1 to N bytes	containing	values to insert	in virtual register

It is assumed that the eoc write will return UTC if the virtual register is undefined or the virtual register is defined as read only.

Note that no virtual registers are currently defined for **Set** operations.

2.1.3.4 Block Get / Set

This optional procedure specifies the mechanism to get or set multiple virtual registers as a block. The operation begins as per a normal get or set operation with the first or "base" virtual register specified as per 2.1.3.2 or 2.1.3.3. Once all bytes of the "base" fixed length virtual register are read or written then the virtual register number is auto-incremented to "base + 1". Subsequent eoc reads or writes would continue as if they had been preceded by the respective ERP get or set operation sequence. In this way multiple successive virtual registers may be read or written while only requiring the first virtual register address to be explicitly specified. As per single virtual register get and set operations this block operation can be interrupted by any other eoc action which does not affect eoc register 5.
This procedure is optional in both the ATU-C and ATU-R. An ATU-R that does not implement this procedure should reply EOD (end of data) when more bytes are read or written than the length defined for the "base" virtual register. EOD should also be returned when the block operation has auto-incremented to an unimplemented virtual register.

2.1.4 Virtual registers

The following virtual registers are defined for ERP version 1 :

Virtual Register #	Use	Length	Description
0	Read (R)	4 bytes	Identification register (1) ms byte – 0x45 (E) 2nd byte – 0x52 (R)

			3rd byte – 0x50 (P) 4th byte – 0x01 (version number)
1	R	32 bytes	All cell counters (2, 3, 4) Interleave path - HEC violation count - HEC total cell count - User total cell count - Idle cell bit error count Fast path - HEC violation count - HEC total cell count - User total cell count - Idle cell bit error count
2	R	32 bytes	SNR per bin for bins 0 to 31 (5, 6)
3	R	32 bytes	SNR per bin for bins 32 to 63
4	R	32 bytes	SNR per bin for bins 64 to 95
5	R	32 bytes	SNR per bin for bins 96 to 127
6	R	32 bytes	SNR per bin for bins 128 to 159
7	R	32 bytes	SNR per bin for bins 160 to 191
8	R	32 bytes	SNR per bin for bins 192 to 223
9	R	32 bytes	SNR per bin for bins 224 to 255
10	R	32 bytes	Attenuation per bin for bins 0 to 31 (5, 7)
11	R	32 bytes	Attenuation per bin for bins 32 to 63
12	R	32 bytes	Attenuation per bin for bins 64 to 95
13	R	32 bytes	Attenuation per bin for bins 96 to 127
14	R	32 bytes	Attenuation per bin for bins 128 to 159
15	R	32 bytes	Attenuation per bin for bins 160 to 191
16	R	32 bytes	Attenuation per bin for bins 192 to 223
17	R	32 bytes	Attenuation per bin for bins 224 to 255
18-255	Read/Write	1 byte	unimplemented virtual registers (4)

Notes

1. virtual registers shall be read most significant byte first
2. virtual registers are sent sequentially as follows; HEC violation count on the Interleaved path is sent first followed by HEC total cell count and so on. Each individual counter is 4 bytes and is returned msb first. The counters are defined in ITU-T G.997.1. When retrieved via this interface these counters are to be rolling counters reset to zero upon transceiver initialization.
3. eoc data read procedures allow multi-byte transfers to be aborted by the ATU-C. When operating in single latency modes where only the interleave path is active (eg. G.992.2), cell counter retrieval may be aborted after the 1st 4 counters are retrieved.
4. unimplemented virtual registers should return UTC. Unsupported sub-registers or elements of a particular virtual register structure should return 0
5. bin 0 is returned first
6. SNR is an estimate of the ratio of signal to noise in this bin independent of signal constellation loading. Each octet is coded as an unsigned integer, ranging from 0 to 127, corresponding to a 0 to 63.5 dB SNR (0.5 dB steps).
7. Attenuation is an estimate of the attenuation across the line from the ATU-C transmitter to the ATU-R receiver. It is coded as an unsigned integer, ranging from 0 to 255, corresponding to a 0 to 127.5 dB attenuation (0.5 dB steps).

3 References

1. ADSL Forum, TR-005, ADSL Network Element Management
2. ADSL Forum, TR-006, SNMP-based ADSL Line MIB
3. ADSL Forum, TR-024, DMT Line Code Specific MIB
4. ITU-T, G.997.1, Physical Layer Management for Digital Subscriber Line (DSL) Transceivers
5. IETF, RFC-2662, Definitions of Managed Objects for the ADSL Lines

6. IETF, draft-ietf-adslmib-adsl2-01.txt, Definitions of Supplemental Managed Objects for ADSL Lines
7. T1.413-1998, Asymmetric Digital Subscriber Line (ADSL) Metallic Interface
8. ITU-T, G.992.1, Asymmetrical Digital Subscriber Line (ADSL) Transceivers
9. ITU-T, G.992.2, Splitterless Asymmetrical Digital Subscriber Line (ADSL) Transceivers

Editors Note: this page will not form part of the final technical report.

Contributory Material to this Technical Report

The following ADSL Forum contributions have been made on this technical report and their comments incorporated:

1. 99-146, ATU-R information exchange across the U interface. Chuck Storry, Newbridge, May 1999.
2. 99-180, 3COM Straw Ballot Vote on WT-040 "Proposal for Alternative OAM Communications Channel across the U Interface", Peter Silverman, 3Com, September 1999
3. 99-189, Ericsson Straw Ballot Vote on WT-040 revision 1 "Proposal for an Alternative AOM Communications Channel across the U Interface", Lars Johansson, Ericsson, September 1999
4. 99-315, "TI Comments on ADSL Forum WT-040", Ben Wiseman and Gail Cone, Texas Instruments, November 1999

WT-040 Revision History

Revision	Date Issued	Editor	Comments
1	June 25, 1999	Chuck Storry	Initial release of WT-040 incorporating comments from 99-146 and Vienna meeting review
2	September 1, 1999	Chuck Storry	incorporate comments from Hawaii meeting contributions
3	November 18, 1999	Chuck Storry	added block virtual register get and set