Mapping of RPR over Sonet/SDH

Portland IEEE 802.17 Meeting – July 2001

Italo Busi – Alcatel

Mark A. Bordogna – Agere

Robin Olsson – Vitesse

Harry Peng – Nortel

Table of Contents

Introduction

Scope of the presentation

- Analyze the possible alternatives to map RPR frames over a Sonet/SDH container
  - HDLC/PPP – as defined in IETF RFC 1661, RFC 1662 and RFC 2615
  - GFP – as defined in T1X1 and ITU-T G.gfp
- Compare the two options

- Analyze the encapsulation issues in IEEE 802.17
  - IEEE 802.17 standard should specify also the encapsulation method(s) supported as part of the Sonet/SDH PHYs

Functionality of the encapsulation mechanism

- Sonet/SDH has a synchronous, octet-oriented interface
  - The Sonet/SDH layer must be fed with a continuous and synchronous octet stream in the egress direction
  - A continuous and synchronous octect stream is received in the ingress direction
- The IEEE 802.17 MAC has an asynchronous, frame-oriented interface
- The encapsulation should adapt these two kinds of interface
  - It should provide a method for frame delineation in the received direction
  - It should provide a method for coding the silence times
  - It should provide a method for map different frame types over the same Sonet/SDH path
  - It should be robust enough to acceptable bit errors
  - It should not be vulnerable to malicious users

HDLC/PPP Encapsulation

HDLC/PPP Encapsulation – a brief summary

1 Byte	1 Byte	1 Byte	2 Bytes	Variable Length	2/4 Bytes	1 Byte
Flag	Address	Control	Protocol	Payload	FCS	Flag

- **Flag** – Fixed to the ‘0111 1110’ (0x7E) value
  - It is used to delineate the frame as well as an inter-frame filler
- **Address** – Fixed to the ‘1111 1111’ (0xFF) value
  - If a different value is received, the frame is discarded
- **Control** – Fixed to the ‘0000 0011’ (0x03) value
  - If a different value is received the frame is discarded
- **Protocol** – It indicates which kind of frame is carried in the payload area
  - If an unexpected or unsupported value is received the frame is discarded
- **Payload** – It represents the RPR MAC frame
- **FCS** – It is the CRC-16 or CRC-32 calculated on all the frame, excluding flags

Frame Delineation and Rate Adaptation

- The flag sequence is used to delineate the frame
  - The first non-flag octet following a sequence of one or more flag octets is considered the first octet of a new frame
  - The end of the frame is detected when the next flag octet is received
  - Silence between two frames are filled by transmitting more flag octets. At least one flag octet is sent between two consecutive frames
- There can not be flag octets inside the frame in the transmitted octet stream
- An octet stuffing procedure is used, based on the escape code '0111 1101' (0x7D), as part of line coding – inflation factor
  - Each flag emulating octet in the payload is substituted by the following couple of octets: 0x7D, 0x5E
  - Each escape code in the payload is substituted by the following couple of octets: 0x7D, 0x5D
  - The opposite operations are done during reception

GFP Encapsulation

GFP Encapsulation – current status

General GFP Frame

2 Bytes	PLI
2 Bytes	cHEC
2 Bytes	Type
2 Bytes	tHEC
Variable Length	Extension Header (3 types)
Variable Length	Payload
4 Bytes	FCS (Optional)

Null Extension header

for point-to-point configurations dedicated to one client signal

Extension header for Linear frame

1 Bytes	CID
1 Bytes	Spare
2 Bytes	eHEC

Extension header for Ring frame

1 Bytes	DP	SP
1 Bytes	Spare1	
1 Bytes	Spare2	DE+Cos
1 Bytes	TTL	
6 Bytes	Destination MAC	
6 Bytes	Source MAC	
2 Bytes	eHEC	

- **IEEE 802.17 needs only (?) the GFP frame encapsulation with null extension header as an alternative to the HDLC/PPP encapsulation**

GFP Encapsulation fields

- **PLI** – PDU Length Indicator
  - It is used to delineate the frame
- **Type** – It indicates the type of encapsulated frame (similar to the Protocol field), the type of extension header and the presence/absence of the client FCS trailer
- **CID** – Channel Identifier used to multiplex different client signals over the same Sonet/SDH path

Frame Delineation and Rate Adaptation

- Using the cHEC as the HEC on the PLI it is possible to identify a candidate start of a frame. The PLI value indicates where is the start of the next frame.
  - After having received two consecutive and correct candidate frames, the alignment state is entered
  - The frames are delineated based on the PLI values
  - The cHEC code is able to correct single-bit errors in the PLI and cHEC fields
  - Alignment is lost when two consecutive frames experience single-bit errors or when a frame experience a multi-bit error in the PLI and cHEC fields
- Silence between two consecutive frames are filled by transmitting idle frames

Idle GFP Frame

2 Bytes	PLI = 0
2 Bytes	cHEC

HDLC and GFP Comparison

GFP and HDLC – a comparison

- GFP is more efficient than HDLC
  - It has no inflation factor but a fixed overhead, almost equal to the minimum overhead in HDLC → traffic management and QoS control is much more easy
- GFP is more robust than HDLC
  - A single bit-error in the PLI or cHEC field does not cause loss of alignment, while with HDLC a single bit-error in the flag causes loss of alignment
  - Malicious users can not degrade the service by sending ad-hoc formed frames
- GFP scales very well to higher bit rates
  - ASIC designs are not impacted by the data rate expansion on a byte level
  - It is also supported by WDM/OTN interfaces
- HDLC has been an established method to map packets over Sonet/SDH interfaces while GFP is a new encapsulation method

Conclusions

Requirements

- IEEE 802.17 should support the GFP encapsulation for Sonet/SDH ring interfaces
  - Ask ITU-T and T1X1 to allocate a Type code representing GFP frames, with a null extension header
- The support of GFP does not exclude supporting also HDLC – the need to support both, as well as any interoperability issue, is left for open discussion