The Generic Framing Procedure (GFP): An Overview

Enrique Hernandez-Valencia, Lucent Technologies
Michael Scholten and Zhenyu Zhu, AMCC

ABSTRACT

The Generic Framing Procedure (GFP) is a recently standardized traffic adaptation protocol for broadband transport applications. It provides an efficiency and QoS-friendly mechanism to map either a physical layer or logical link layer signal to a byte-synchronous channel. It also supports basic client control functions for client management purposes. This article presents a brief overview of GFP.

INTRODUCTION

Transmission rates in the backbones of the public transport networks will continue to rise with each new innovation in electro/optical transceiver/ transponder technology. As technology matures, existing LAN, MAN, and WAN data networking solutions become candidates for integration into the existing transport network infrastructure, preferably under a common data transport framework. Over the last few years, a need has arisen for a simple traffic adaptation mechanism that can be used to integrate the current diverse spectrum of physical and data link layer formats into the common public transport network infrastructure. The desired traffic adaptation mechanism must be simple enough to scale gracefully with the ever-increasing transmission rates in the core of the transport networks. However, it must be flexible enough to accommodate diverse data transmission requirements: from stringent delay and throughput constraints in storage networking, to differentiated quality of service (QoS) handling in LAN interconnect, to best effort delivery service for Internet browsing.

The Generic Framing Procedure (GFP) is a simple but flexible traffic adaptation mechanism specifically designed to transport either block-coded or packet-oriented data streams over a byte-synchronous communications channel. GFP generalizes the error-control-based frame delineation scheme successfully employed in asynchronous transfer mode (ATM) [1] to both fixed and variable length data. Unlike frame delineation mechanisms based on codeword delineation patterns, GFP does not require special preprocessing of the client's byte stream. Rather than relying on embedded data/control bits such as in 8B/10B and 64B/66B encoding [2, 3], or delineation flags such as in HDLC framing [4], GFP relies on the length of the current payload and an error control check for frame boundary delineation. Successful validation of these two pieces of information, conveyed in the GFP frame header, is used to determine proper data link synchronization and the number of bytes to the next incoming frame.

By facilitating the processing of arbitrary blocks of bytes at a time, GFP substantially reduces processing requirements for data link mappers/demappers. By exploiting the low bit error rate performance in modern fiber-based communications media for the data link synchronization logic, GFP further decreases receiver logic. This reduced implementation complexity makes GFP particularly suitable for high-speed transmission links such as point-to-point synchronous optical network/synchronous digital hierarchy (SONET/SDH links) [5, 6], wavelength channels in an optical transport network (OTN) [7], or even dark fiber applications.

GFP allows the implementation of multiple transport modes that may coexist within the same transport channel. One mode, referred to as Frame-Mapped GFP, is optimized for packet switching environments where resource management functions are delegated to the native data clients. This is the transport mode used for native Point-to-Point Protocol (PPP), IP, multiprotocol label switching (MPLS), or Ethernet traffic. A second mode, referred to as Transparent-Mapped GFP, is intended for delay-sensitive storage area network (SAN) applications that require bandwidth efficiency and transparency to the line code data. This is the transport mode used for Fibre Channel, FICON, and ESCON traffic. The current GFP specification is a result

0163-6804/02/$17.00 © 2002 IEEE

A need emerged for a standard based mechanism to accommodate the diverse datalink transport technologies prevalent in the enterprise market over the existing public transport infrastructure. GFP soon found applications over SONET and SDH as well as the emerging Optical Transport Network technology.

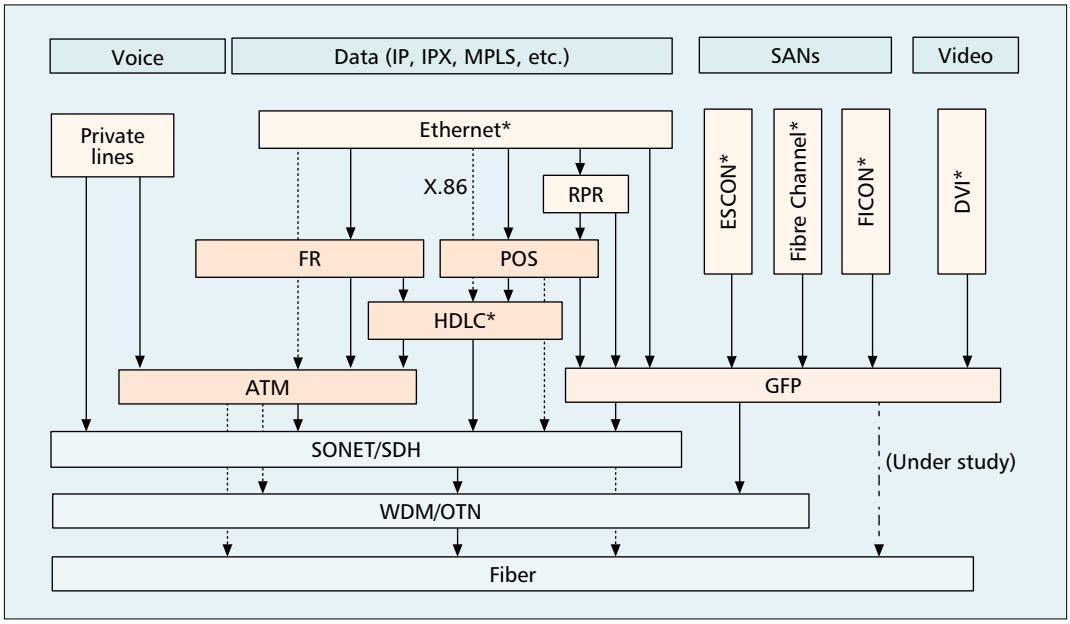

Figure 1. *Adapting voice, data, storage, and video traffic over the public transport network infrastructure. (* These traffic types may also run directly over fiber.)*

of a joint standardization effort in both American National Standards Institute (ANSI) Committee T1X1.5 and International Telecommunication Union — Telecommunication Standardization Union (ITU-T) Study Group 15 (Question 11). ITU-T Recommendation G.7041 [8] reached consent in October 2001. The ANSI specification was ratified in January 2002.

The remainder of this article starts with a short historical background on the development of the GFP specification, highlighting key market drivers and development requirements. The article continues with a brief description of basic frame structures and procedures. Sample applications of GFP adaptation modes are discussed in detail in other articles in this issue.

WHY GFP?

GFP finds its roots in the data-driven traffic growth of the late '90s. The initial applications were envisioned as a transport solution for data-centric traffic over readily available dark fibers [8] from recently deployed dense wavelength-division multiplexing (DWDM) systems. Later on, a need emerged for a standard-based mechanism to accommodate the diverse data link transport technologies prevalent in the enterprise market over the existing public transport infrastructure. GFP soon found applications over SONET and SDH as well as the emerging OTN technology. Various factors have contributed to these developments.

A TRANSPORT LAYER PERSPECTIVE

Although there has long been a desire to extend the reach of SONET/SDH-based transport networks to the very edge of corporate networks, three major concerns have hindered the widespread acceptance of SONET/SDH technology as the preferred transport vehicle for data traffic into the WAN. First, most commercial SONET/SDH add/drop multiplexers (ADMs) and broadband cross-connect systems (BXCs) had been heavily optimized to transport voice traffic, including support of timing and protection requirements, over the transport requirements of other traffic types. Although such optimizations were not harmful to packet switching technologies being deployed in the long-haul backbones (e.g., frame relay and ATM), the access model was burdensome to the bulk of LAN and MAN technologies typically deployed in enterprise networks (Ethernet, FDDI, ESCON, etc.). Second, the SONET/SDH management and operational framework had traditionally presumed direct end-user access to SONET/SDH channels (at the SONET/SDH path level). The higher cost structure of such an access model was inconsistent with the low-cost transport needs of the enterprise networking market. Third, even though it is possible to transport LAN technologies over the public SONET/SDH-based transport network infrastructure by defining a suitable mapping of the LAN bitstream onto the SONET/SDH payload area, both the rigidity of the readily available SONET/SDH channel sizes and the lack of a common adaptation procedure of the native data streams into the SONET/SDH path further slowed a more proactive deployment. Lastly, a native packet-oriented transport mechanism would enable alternative path protection and sharing options for both linear and ring configurations.

The first two issues can easily be handled by the adoption of a hybrid ADM and BXCs model that incorporates both SONET/SDH and packet-based access interfaces into a single system as well as support for flexible provisioning of traffic resiliency options. This approach preserves a well-known service interface toward the end user while placing the adaptation of the native bit-

stream into the SONET/SDH payload inside the hybrid transport system, rather than within the end user's customer premises equipment (CPE). Projects under the Data Aware initiative in ANSI and ITU-T directly addressed the open technical concerns. The introduction of inverse multiplexing facilities (the ability to bundle multiple independent lower-rate channels to create a logical higher-rate channel), referred to as *virtual concatenation* of synchronous payloads, and the specification of dynamic link capacity and adjustment schemes (LCAS) [9] to dynamically and on demand provision and reconfigure time-division multiplexed (TDM) channels to fit end-user needs, addressed granularity concerns with SONET/SDH (and OTN) transport. The final step, a common adaptation mechanism, is provided by GFP.

A Services Layer Perspective

Figure 1 illustrates a simplified taxonomy of the transport options for end-user applications such as voice, data, storage, and video traffic over the public network infrastructure. Frame relay, ATM, and packet over SONET/SDH (POS) represent the dominant service layer technologies for data delivery to small office/ home office (SOHO) and large corporations. HDLC is the traffic adaptation mechanism (to a bit/byte-synchronous physical transmission channel) in native frame relay and POS services. Until recently there has been limited support for connectivity services for protocols for storage networking or video over the traditional service technologies. When handled at the physical level (layer 1) Ethernet and SAN protocols such as Fibre Channel, ESCON, and FICON have traditionally been transported over the public network infrastructure by means of proprietary solutions that either optically extend the reach of the native signal or mimic an electrical "repeater" function.

Given the widespread availability of inexpensive 10/100/1000 Mb/s Ethernet interfaces for CPE switches/routers, the growing need to improve data center/SAN interconnectivity, as well as the recent additions of virtual-LAN-based virtual private networking (VPN) and QoS capabilities via IEEE 802.1Q/p, there is renewed interest in a common QoS-friendly standard-based mechanism to transport IP, Ethernet, and SAN traffic over both TDM and WDM networks. Although ATM and HDLC are the most common mechanisms employed to adapt native data traffic to the public transport network infrastructure, both have their own limitations as the base for such a simple common adaptation solution. In the case of ATM, it provides both transport and service layer integration that is far more than required to support simple connectivity services of interest to service providers. The large ATM cell tax is also an issue when bandwidth efficiency is a concern. HDLC has been the technology of choice for best effort wide-area services, but it has long been distrusted as a reliable broadband transport mechanism for value-added connectivity services. For a more in-depth discussion of the relevant trade-offs, see a companion article on GFP applications in this issue.

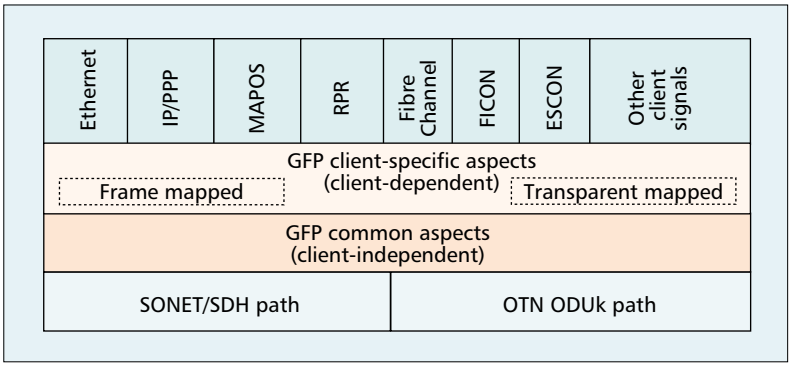

Figure 2. *GFP relationship to client signals and transport paths.*

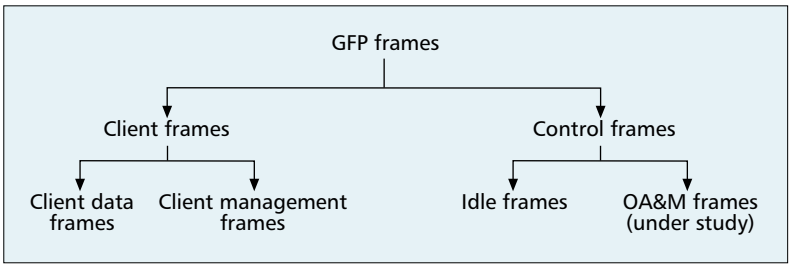

Figure 3. *GFP frame types.*

The GFP Solution

GFP provides a flexible encapsulation framework that supports either a fixed or variable length frame structure. As opposed to HDLC-like framing, GFP does not rely on a byte stuffing mechanism to delineate protocol data units (PDUs). Instead, GFP uses a variation of the HEC-based self-delineation technique. To accommodate variable lengths PDUS, an explicit payload length indicator is provided in the GFP frame header. Thus, the GFP PDU size can be fixed to a constant value (to provide a TDM-like channel), or changed from frame to frame (to allow easy extraction of the encapsulated PDUs). The explicit frame size indication also bounds the duration of the frame boundary hunt process, which is critical for data link synchronization. This client adaptation feature supports full encapsulation of the variable-length user PDU, hence obviating the need for segmentation/ reassembly functions, or frame padding to fill unused payload space. GFP also segregates error control between the GFP adaptation process and the user data. Error control segregation allows the delivery of corrupted payloads to be handed back to the intended receiver for further processing. This is a convenient feature for the transport of audio and video streams where corrupted information is preferred to no information at all.

Functionally, GFP consists of both common and client-specific aspects. Common aspects of GFP apply to all GFP adapted traffic and cover functions such as PDU delineation, data link synchronization and scrambling, client PDU multiplexing, and client-independent performance monitoring. Client-specific aspects of GFP cover issues such as mapping of the client PDU into the GFP payload, client-specific per-

Error control segregation allows the delivery of corrupted payloads to be handed back to the intended receiver for further processing. This is a convenient feature for the transport of audio and video streams where corrupted information is preferred to no information at all.

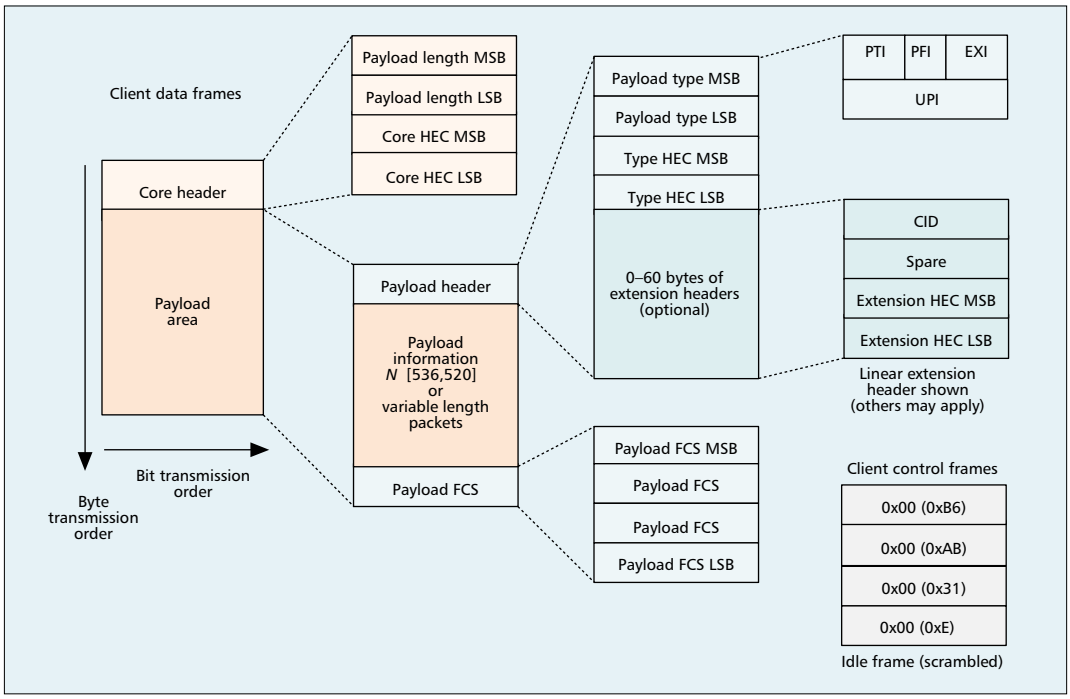

Figure 4. *GFP frame structure.*

formance monitoring, and operations, administration, and maintenance (OA&M). This is illustrated in Fig. 2.

GFP Frame Structure

From a functional standpoint, two basic GFP frame types are currently defined: GFP client frames and GFP control frames. Two types of GFP client frames are defined: client data frames (CDFs) and client management frames (CMFs). GFP client data frames are used to transport client data. GFP client management frames are used to transport information associated with the management of the client signal or GFP connection. This functional decomposition is illustrated in Fig. 3.

GFP Core Header

The GFP core header is intended to support GFP-specific (rather than client-specific) data link management functions (Fig. 4). The core header allows GFP frame delineation independent of the content of the higher-layer PDUs, which is a variation of the HEC-based frame delineation mechanism defined for ATM. The GFP core header is 4 bytes long and consists of two fields:

Payload Length Indicator (PLI) Field: A two-byte field indicating the size in bytes of the GFP payload area. It indicates the beginning of the next GFP frame in the incoming bitstream as an offset from the last byte in the current GFP core header. PLI values in the range 0–3 are reserved for GFP internal usage and are referred to as GFP control frames. All other frames are referred to as GFP client frames.

Core HEC (cHEC) Field: A two-octet field containing a cyclic redundancy check (CRC-16) sequence that protects the integrity of the core header. The cHEC sequence is computed over the core header bytes using standard CRC-16. The CRC-16 enables both single-bit error correction and multibit error detection. Single-bit error correction is advised, particularly over high bit error rate links.

GFP Payload Area

The payload area covers all remaining bytes of the GFP frame following the GFP core header. This variable length area may include from 0 to 65,535 octets. It is generally intended to support client-specific aspects of GFP, such as clients PDUs (layer 2 of the Open Systems Interconnection, OSI, Reference Model), link layer codewords (layer 1), or GFP client management information. Structurally, the payload area consists of two mandatory components, a payload header and a payload information field, and a third optional component, the payload FCS field. The format and functions of these components are further described in this section.

Payload Header — The payload header is a variable-length area, 4–64 octets long, intended to support data link management procedures specific to the transported client signal. The payload header contains two mandatory fields, the type field and an accompanying type header error correction (tHEC) field. The tHEC protects the integrity of the type field. Optionally, the payload header may include an additional variable number of subfields, referred to as a group as the extension header. The presence of the extension header and its format, and the presence of the optional payload FCS are specified by the type field. The tHEC protects the integrity of the type field.

GFP Payload Type Field — The payload type field is a mandatory 2-octet field of the payload header that indicates the content and format of the GFP payload information. The type field dis-

tinguishes between services in a multiservice environment. The type field consists of:

Payload Type Identifier (PTI): A 3-bit subfield that identifies the type of GFP client frame. Two kinds of user frames are currently defined: user data frames and client management frames.

Payload FCS Indicator (PFI): A 1-bit subfield indicating the presence or absence of the payload FCS field.

Extension Header Identifier (EXI): A 4-bit subfield identifying the type of extension header GFP. Three kinds of extension headers are currently defined: a null extension header, a linear extension header, and a ring extension header (currently under study).

User Payload Identifier (UPI): An 8-bit field identifying the type of payload conveyed in the GFP payload information field. UPI values for client data and client management frames are specified below. The UPI is set according to the transported client signal type. Defined UPI values for client data frames include:

- Frame-mapped Ethernet
- Frame-mapped PPP (including IP and MPLS)
- Transparent-mapped Fibre Channel
- Transparent-mapped FICON
- Transparent-mapped ESCON
- Transparent-mapped Gb Ethernet

Transparent mappings for 10/100 Mb/s Ethernet, digital video broadcast, and Infiniband are also under consideration.

Type HEC (tHEC) Field: A 2-octet field that contains an International Organization for Standardization (ISO) CRC-16 sequence to protect the integrity of the type field. The tHEC sequence is computed over the core header bytes using standard CRC-16. As with the cHEC, CRC-16 enables both single-bit error correction and multibit error detection. Single-bit error correction is advised, particularly over high bit error rate links.

GFP Extension Headers — GFP also supports a flexible (payload) header extension mechanism to facilitate adaptation of GFP for use with diverse transport mechanisms. The payload extension header is a 0–60-byte extended field (including the extension headers HEC field, eHEC) that supports technology-specific data link headers such as virtual link identifiers, source/destination addresses, port numbers, class of service, eHEC, and so on. The type of the extension header is indicated by the content of the EXI bits in the type field of the payload header. Three extension header variants, null, linear, and ring, are currently defined to support client-specific data over logical ring or logical point-to-point (linear) configurations.

Null Extension Header: When the EXI is set to this value, there is no extension header present. This is the default case when the entire GFP payload is dedicated to the transport of a single payload type

Linear Extension Header: A 2-octet extension header that supports sharing of the GFP payload across multiple clients in a point-to-point configuration. The linear extension header is intended for scenarios where several independent links are being aggregated onto a single transport path. The linear extension header consists of an 8-bit channel ID (CID) field, used to indicate one of 256 communications channels at a GFP termination point and an 8-bit spare field reserved for future use.

Ring Extension Header: The current proposal considers an 18-octet extension header that supports sharing of the GFP payload across multiple clients in a ring configuration. A similar functionality is envisioned in the IEEE 802.17 resilient packet ring (RPR)-based medium access control (MAC) for packet ring functions. In the RPR case, the MAC PDU would be conveyed using a null extension header instead.

Extension HEC (eHEC) Field: This a mandatory 2-octet field containing an ISO CRC-16 check sequence that protects the integrity of the contents of the extension. CRC-16 enables both single-bit error correction and multibit error detection.

Payload Information Field — The payload information field contains the framed PDU. This variable length field may include from 0 to 65,535 – X, octets, where X is the size of the payload header (including the extension header, if present) and the payload FCS field (if present). The client user/control PDU is always transferred into the GFP payload information field as an octet-aligned packet stream. The payload adaptation procedures are described later.

Payload Frame Check Sequence (FCS) — This is an optional 4-octet-long frame check sequence. It contains a CRC-32 check sequence that protects the contents of the GFP payload information field. The payload FCS is generated using the CRC-32 generating polynomial (ISO 3309). A value of 1 in the PFI bit within the type field indicates the presence of the payload FCS field. Unless otherwise stated, corrupted GFP frames are passed to a client adaptation process for local handling according to client-specific rules.

GFP Client-Independent Processes

GFP supports six basic procedures common to all payloads: frame delineation, client/frame multiplexing, header/payload scrambling, and client management. These procedures are overviewed in this section.

GFP Frame Delineation

The GFP transmitter and receiver are designed to operate asynchronously. The GFP transmitter inserts GFP frames on the physical link according to the bit/byte alignment requirements of the specific physical interface (e.g., SONET/SDH, OTN, or dark fiber). The critical function of the GFP receiver is to identify the correct GFP frame boundary at the time of link initialization and also after link failures or packet delineation loss events. This function is performed in a three-state process:

Hunt State: The base state when the link is initialized or after GFP receiver failures are detected. The receiver "hunts" for the next GFP frame using the current four octets of data. If the computed cHEC matches the value in the cHEC field, the receiver tentatively assumes that

The current proposal considers an 18-octet extension header that supports sharing of the GFP payload across multiple clients in a ring configuration. A similar functionality is envisioned in the IEEE 802.17 Resilient Packet Ring-based MAC for the packet ring functions.

Typically a GFP client-specific source adaptation process would send periodic far-end CSF indications upon detection of a failure/degradation. The GFP client-specific sink adaptation process should clear the defect condition either after failing to receive a number of consecutive CSF indications, or after receiving a valid GFP User Frame.

it has identified the frame boundary; otherwise, it shifts forward by 1 bit/byte and checks again.

Pre-Sync State: Intermediate state after a candidate GFP frame is identified. The transmitter waits for the next candidate GFP frame based on read PLI field value. If *N* consecutively GFP frames are detected, the receiver transitions to the Sync state. Core header error correction is disabled to minimize false detection events.

Sync State: The regular operational state. The receiver examines the PLI field, validates the incoming cHEC field, extracts the framed PDU, and then rolls over to the next GFP frame. Core header single-bit error detection and correction is enabled.

The HEC-based frame delineation procedure permits sophisticated traffic engineering, flexible QoS-aware routing, better partitioning of SONET/SDH bandwidth, and multiservice integration, either at the GFP layer (via header extension options with the GFP payload header) or via native layer 2 (e.g., Ethernet) or layer 3 (e.g., IP/MPLS) mechanisms.

Core Header Scrambling

The core header is always scrambled on transmission (via a exclusive OR operation) with a well-known Barker-like pattern, 0xB6AB31E∅. Core header scrambling ensures high bit transition density during idle data link operations.

Payload Area Scrambling

Scrambling of the GFP payload area is required to provide security against payload information replicating the scrambling word (or its inverse) from a frame-synchronous scrambler such as those used in the SONET line layer (SDH RS layer) or in an OTN OPUk channel. All octets in the GFP payload area are scrambled using a $1 + x^{43}$ self-synchronous scrambler. As in ATM, scrambling is enabled starting at the first transmitted octet after the cHEC field, and disabled after the last transmitted octet of the GFP frame (and the state retained).

Frame Multiplexing

GFP frames from multiple GFP processes (idles, client data frames, client management frames) are multiplexed on a frame-by-frame basis. The expectation is that client data frames are always sent with preference over client management frames. When there are no other GFP frames available for transmission, GFP idle frames must be inserted. This provides a continuous stream of frames for mapping into an octet-aligned physical layer.

Client Multiplexing

GFP supports client multiplexing capabilities via the GFP linear and ring extension header formats. The choice of scheduling algorithms for distinct client frame types is currently outside the scope of the GFP specification. For instance, multiple clients of the same type (e.g., Gigabit Ethernet clients) may easily be multiplexing via a round-robin scheduling. Class-based queuing may be used for multirate multiplexing. However, there is the expectation that a build-out buffer will be provided to smooth out any jitter introduced by the client multiplexing stage.

Client Management

GFP provides a generic mechanism to propagate client-specific source adaptation information, such as performance monitoring, OA&M information, or embedded management channels. This mechanism uses a common GFP client frame type referred to as client management frames (CMF). Currently, the only client-specific facility defined is related to the propagation of client interface failure conditions, referred to generally as client signal fail (CSF).

Client Signal Fail — CSF is a message that may be sent from the GFP source adaptation process to the far-end GFP sink adaptation process upon failure/degradation detection in the ingress client signal. Detection rules for client signal failure events are by definition client-specific, and recommendations are provided for the various clients supported by GFP. The CSF indication is a special type of GFP client frame consisting only of a payload header and no payload information field. Two generic types of failure defects can be reported:

- Loss of client signal (e.g., loss of light)
- Loss of client character synchronization

Typically a GFP client-specific source adaptation process would send periodic far-end CSF indications upon detection of a failure/degradation. The GFP client-specific sink adaptation process should clear the defect condition after either failing to receive a number of consecutive CSF indications, or receiving a valid GFP user frame.

Handling of incomplete GFP frames at the onset of a CSF event should be consistent with the GFP error handling procedures. Client-specific suggestions are provided in the specification [8].

GFP Client-Specific Processes

There are two modes of client signal payload adaptation defined for GFP:

- Frame-mapped GFP (GFP-F) applicable to most packet data types
- Transparent-mapped GFP (GFP-F) applicable to 8B/10B coded signals

Frame-mapped GFP payloads consist of variable length packets. In this mode, each received client signal frame is mapped in its entirety into one GFP frame. Examples of such client signals include Gigabit Ethernet and IP/PPP.

For transparent-mapped GFP, a number of client data characters are character mapped into efficient block codes for transport within a GFP frame. The payload consists of *N* 67-byte [536,520] *superblocks*, where each superblock is constructed of eight 65B blocks followed by a 16-bit CRC-16 superblock checksum.

Below is a short discussion of GFP-F and GFP-T adaptation processes. A summary comparison of these two mechanisms is provided in Table 1.

Frame-Mapped GFP

Frame mapping of native L2 payloads into GFP is intended to facilitate the transport of arbitrarily coded client signals for scenarios that require packet-level handling of incoming PDUs. Examples of such client signals include IEEE 802.3 Ethernet MAC frames, PPP/IP packets, or any

Frame-mapped GFP	Transparent-mapped GFP
Variable length GFP frames.	Fixed length GFP frames.
1-to-1 mapping of data packets to GFP frames.	*N*-to-1 mapping of client "characters" to GFP frames.
Point-to-point, multipoint, packet aggregation, or resilient packet ring network topology.	Primarily point-to-point topology using virtual concatenation.
Requires MAC awareness to terminate client signal and pass only data packets.	Only 8B/10B PHY layer terminated; MAC not required terminating higher-layer protocol.
Data only passed in 8B format.	Data and control compressed using 64B/65B recoding.
Channel-associated control possible using GFP Control Frames.	Channel-associated control possible using GFP control frames.
Client LOS, loss of sync, or code violations may be communicated to far end based on L2/L3 fault management rules.	Defines client-specific mechanisms for communicating LOS, loss of sync, or code violations to far end.
Client egress action due to SONET/SDH signal failure based on L2/L3 FM rules.	Defines client egress action due to SONET/SDH signal failure.

Table 1. *Frame-mapped GFP vs. transparent-mapped GFP.*

HDLC framed PDU. Here, the transmitter encapsulates an entire frame of the client data into its own GFP frame. The adaptation procedure applies only to layer 2 PDUs rather than the physical layer client signal (e.g., client data and control characters). Frame multiplexing is supported with frame-mapped GFP. Frame-mapped GFP uses the basic frame structure of a GFP client frame, including the required payload header. The payload FCS is optional. A typical GFP-F frame (assuming a null extension header) is depicted in Fig. 5a.

Transparent (8B/10B) Mapped GFP

Transparent mapping of 8B/10B payloads into GFP is intended to facilitate the transport of 8B/10B block-coded client signals for scenarios that require very low transmission latency. Examples of such client signals include Fibre Channel, ESCON, FICON, and Gigabit Ethernet. Rather than buffering an entire frame of client data into its own GFP frame, the individual characters of the client signal are demapped from the client block codes and then mapped into periodic fixed-length GFP frames. The mapping occurs regardless of whether the client character is a data or control character, which thus preserves the client 8B/10B control codes. Frame multiplexing is not precluded with transparent GFP.

The transparent GFP client frame uses the same structure as the frame-mapped GFP, including the required payload header. The payload FCS is optional. A typical GFP-T frame is depicted in Fig. 5b.

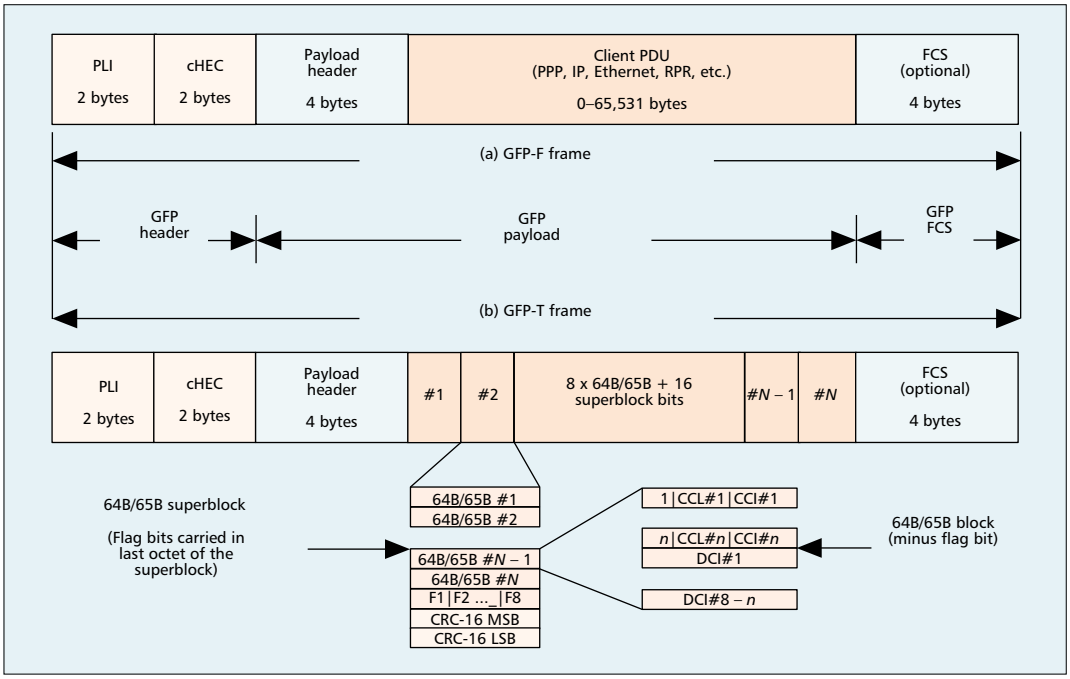

Figure 5. *Adapting traffic via GFP-F and GFP-T.*

Transparent mapping of 8B/10B payloads into GFP is intended to facilitate the transport of 8B/10B block-coded client signals for scenarios that require very low transmission latency. Examples of such client signals include Fibre Channel, ESCON, FICON, and Gigabit Ethernet.

Client data rate	Client signal	VC path size	Minimum superblocks/GFP frame
160 Mb/s	ESCON	VC-3-4v	1
425 Mb/s	Fibre Channel	VC-4-3v	13
850 Mb/s	Fibre Channel / FICON	VC-4-6v	13
1000 Mb/s	Gigabit Ethernet	VC-4-7v	95
1700 Mb/s	Fibre Channel	VC-4-12v	13

NOTE: The minimum number of superblocks here assumes a Null Extension Header and no optional payload FCS.

Table 2. *VCG Sizes to transport various clients.*

The first step in the client adaptation process is decoding the physical layer of the client signal. For 8B/10B line codes, the received 10-bit character is decoded into its original 8-bit value, if it is an 8B/10B data codeword (or data codeword indicator, DCI), or into a control character if it is an 8B/10B control codeword. 8B/10B control codewords are mapped into one of the 16 possible 4-bit control code indicators (CCIs) for the 8-bit control characters available in transparent GFP.

The second step is to map the decoded 8B/10B characters into a 64-bit/65-bit (64B/65B) block code. A bit of the 65-bit block, the flag bit, indicates whether the block contains only 64B/65B 8-bit data characters (DCIs) or client control characters are also present in the block. (Flag bit = 0 indicates data octets only; flag bit = 1 indicates at least one control octet in the block.) Client control characters, which are mapped into 8-bit 64B/65B control characters, are located at the beginning of the 64-bit block payload if they are present in that block. The first bit of the 64B/65B control character contains a last control character (LCC) flag bit which indicates whether this control character is the last one in this block (LCC=0), or there is another control character in the next octet (LCC = 1). The next three bits contain the control code locator (CCL), which indicates the original location of the 8B/10B control code character within the sequence of eight client characters contained in the block. The last 4 bits, the control code indicator (CCI), give the 4-bit representation of the 8B/10B control code character.

The third step creates a [536,520] superblock. The superblock structure groups eight consecutive 64B/65B blocks into a single transport unit. The flag bits of each 64B/65B block are placed together in a single octet at the end of the superblock. This arrangement keeps the superblock structure octet-aligned, which simplifies implementation and monitoring. A CRC-16 that supports multibit error detection and single-bit error correction is added to protect the integrity of the superblock.

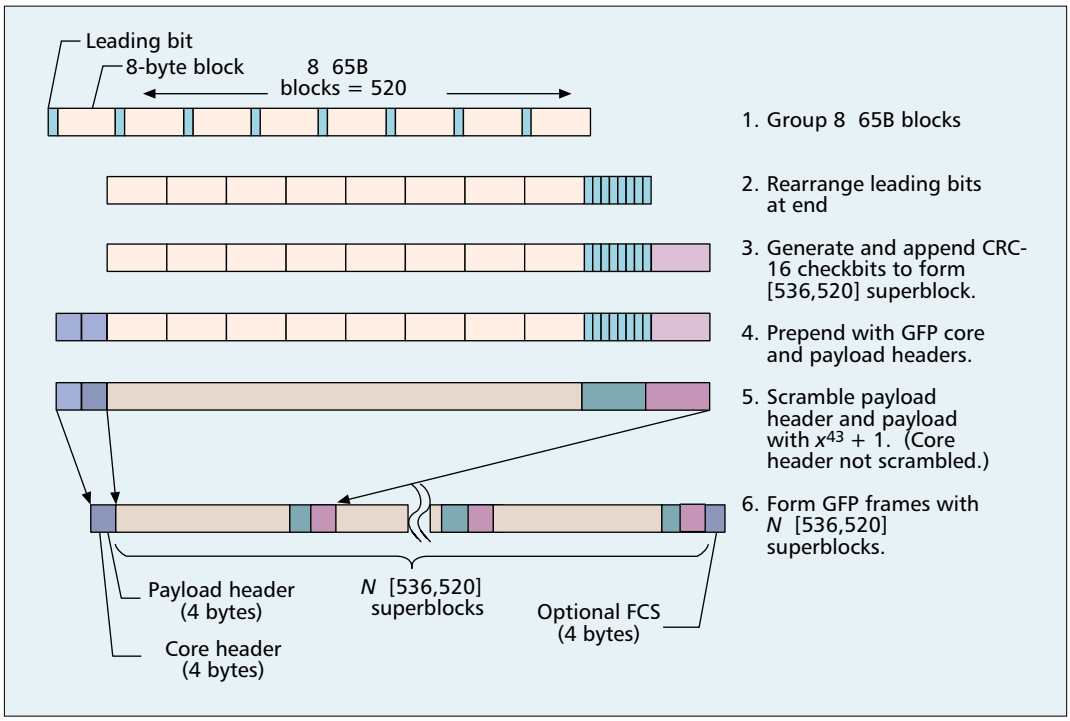

Figure 6. *Construction of [536,520] superblocks from 8 64B/65B-encoded blocks.*

The GFP-T frame generation procedure is illustrated in Fig. 6.

To successfully transport transparent-mapped client signals, the transport path must provide sufficient bandwidth to accommodate the recoded client signal and GFP frame overhead. Typically, transparent-mapped GFP frames are expected to be mapped into a "right-sized" virtually concatenated group (VCG) of SONET/SDH paths, where right-sized means the smallest possible VCG based on the client line rate. Table 2 lists the VCG sizes expected to be typically used for GbE, Fibre Channel, and ESCON. This table also indicates the minimum number of [536,520] superblocks that must be mapped into each GFP frame. If fewer superblocks are mapped into each GFP frame, the GFP frame bit rate will exceed the payload capacity of the VCG, and client data will be lost.

Since client and SONET clocks are asynchronous, it is impossible to perfectly match GFP frame bit rate to payload data rate. In addition, some spare capacity may be desired to transport client management frames. As a result, a mechanism is needed to fill spare payload capacity when insufficient client characters are available and no client management frames are waiting to be sent. GFP provides two such rate adaptation mechanisms: GFP idle frames and 65B_PAD characters.

CONCLUSIONS

GFP is a low-complexity adaptation mechanism for data client signals into a byte-synchronous communications channel. The adaptation mechanism uses pointer and header CRC to delineate encapsulated PDUs of fixed or variable length. Support is provided for the direct mapping of either multipoint or point-to-point data client signals. Clients can be either physical layer signals or logical data link signals. These characteristics make GFP particularly suitable for data adaptation over SONET/SDH as well as the next generation of optical transport networks.

REFERENCES

[1] ITU-T Rec. I.432.1, "B-ISDN User-Network Interface — Physical Layer Specification," 1993.

[2]ISO/IEC 8802-3: 2000 (E), "Information Yechnology — Telecommunications and Information Exchange Between Systems — Local and Metropolitan Area Networks — Specific Requirements — Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," 2000.

[3] P802.3ae/D3.1, "Draft IEEE Supplement to ISO/IEC 8802-3 (IEEE 802.3) — Media Access Control (MAC) Parameters, Physical Layer, and Management Parameters for 10Gb/s Operation."

[4] ISO/EIC 3309:1991(E), "Information Technology – Telecommunications and Information Exchange between Systems — High-Level Data Link Control (HDLC) Procedures — Frame Structure," 4th ed., 1991.

[5] "American National Standard For Telecommunications — Synchronous Optical Network (SONET): Physical Interfaces Specifications", ANSI T1.105.06, 2000.

[6] ITU-T Rec. G.707, "Network Node Interface for the Synchronous Digital Hierarchy (SDH)," 1996.

[7] ITU-T Rec. G.709, "Interfaces for the Optical Transport Network (OTN)," 2001.

[8] J. Carlson, P. Langner, J. Manchester, and E. Hernandez-Valencia, "The Simple Data Link (SDL) Protocol," IETF RFC 2823, May 2000.

[9] ITU-T Rec. G.7041/Y.1303, "Generic Framing Procedure (GFP)," 2001.

[10] ITU-T Draft Rec. G.7042/Y.1305, "Link Capacity Adjustment Scheme (LCAS)," 2001.

BIOGRAPHIES

ENRIQUE HERNANDEZ-VALENCIA [M] (enrique@lucent.com) is a distinguished member of technical staff at Lucent Technologies Bell Laboratories. He received his B.Sc. degree in electrical engineering from the Universidad Simon Bolivar, Caracas, Venezuela, and his M.Sc. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena. Since joining Bell Laboratories in 1987, he has worked in the research and development of high-speed data communications protocols and systems. He is a member of the ACM and Sigma Xi.

MICHAEL SCHOLTEN (mscholte@amcc.com) is a senior principal engineer at Applied Micro Circuits (AMCC) functioning as a senior systems engineer. He is responsible for specifying new products within AMCC's Digital Product Division. He represents AMCC in T1X1 standards committees and consults with AMCC's representatives to ITU-T, OIF, IEEE, and other standards organizations. Previously he has worked as a software engineer, software development manager, product development manager, program manager, and systems engineer in various optical telecommunications companies. He is a graduate of Michigan Technological University.

ZHENYU ZHU is a senior staff engineer at AMCC functioning as a senior systems engineer. He is responsible for specifying new products within AMCC's Digital Product Division. Previously he has worked as system engineer, ASIC design engineer, and network engineer in Lucent Technologies, Siemens Mobile Communications, and other communications companies. He holds degrees from Lehigh University, Chinese Academy of Science, and Shanghai Jiao Tong University. His interests include optical communications, data networks, wireless communications, and communication/networking ASIC implementations.

GFP is a low-complexity adaptation mechanism for data client signals into a byte synchronous communications channel. The adaptation mechanism uses a pointer and header CRC to delineate encapsulated PDUs of fixed or variable-length.