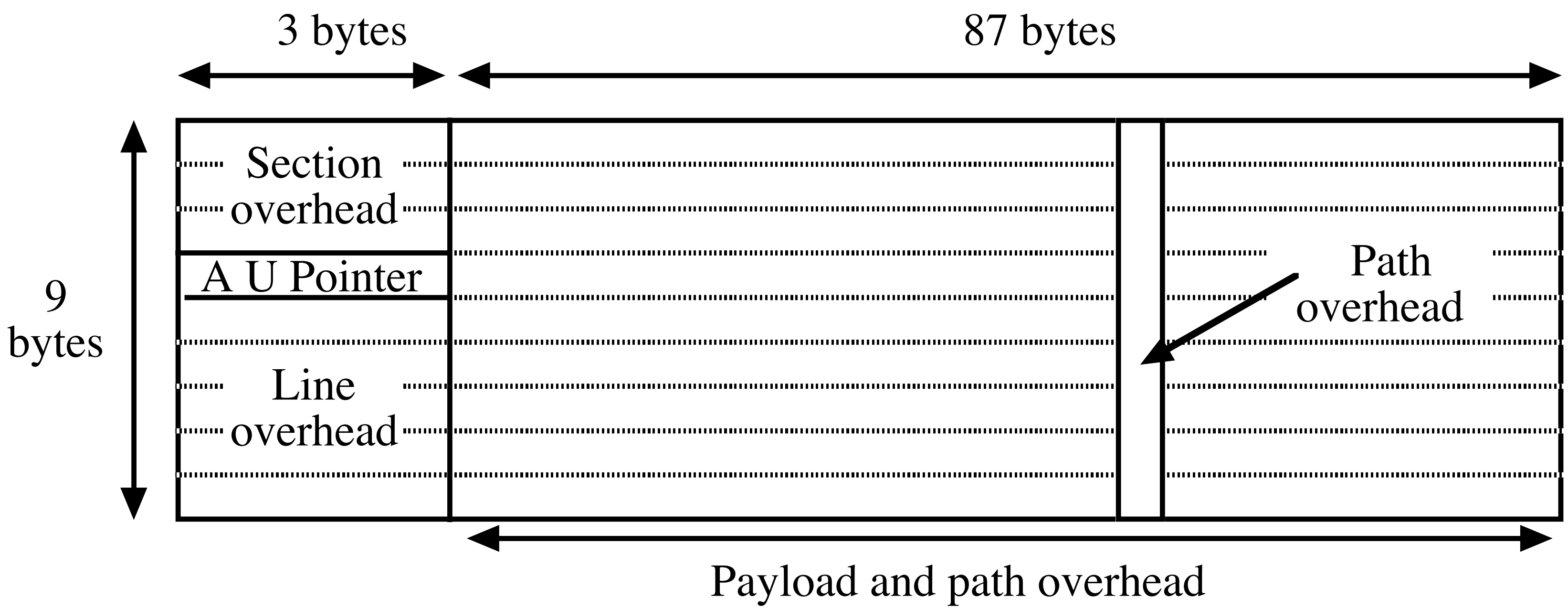
SDH and SONET

- The ITU-T (was CCITT) specifies a Synchronous Digital Hierarchy, or SDH, based on a 125μs frame (8,000 frames/second).
- The basic STM-1 frame has 9 rows of 270 columns with the bytes transferred in "raster scan" (left to right, top to bottom).
- Higher speed STM-*n* frames are formed by byte-interleaving *n* of the basic STM-1 frames.
- A 1.544 Mbps (North American T-1) source maps into groups of 27 bytes (25 T-1 data bytes + 2 overhead).
- A 2.048 Mbps stream (CCITT E-1) maps into groups of 36 bytes (32 E-1 data bytes + 4 overhead).

SONET STS-1 Frame

- The basic frame is the SONET (Synchronous Optical Network) frame based on 9×(3+87) = 810 octets with 8000 frames per second giving a line speed of 51.840 Mbit/s.



SONET STS-1 frame

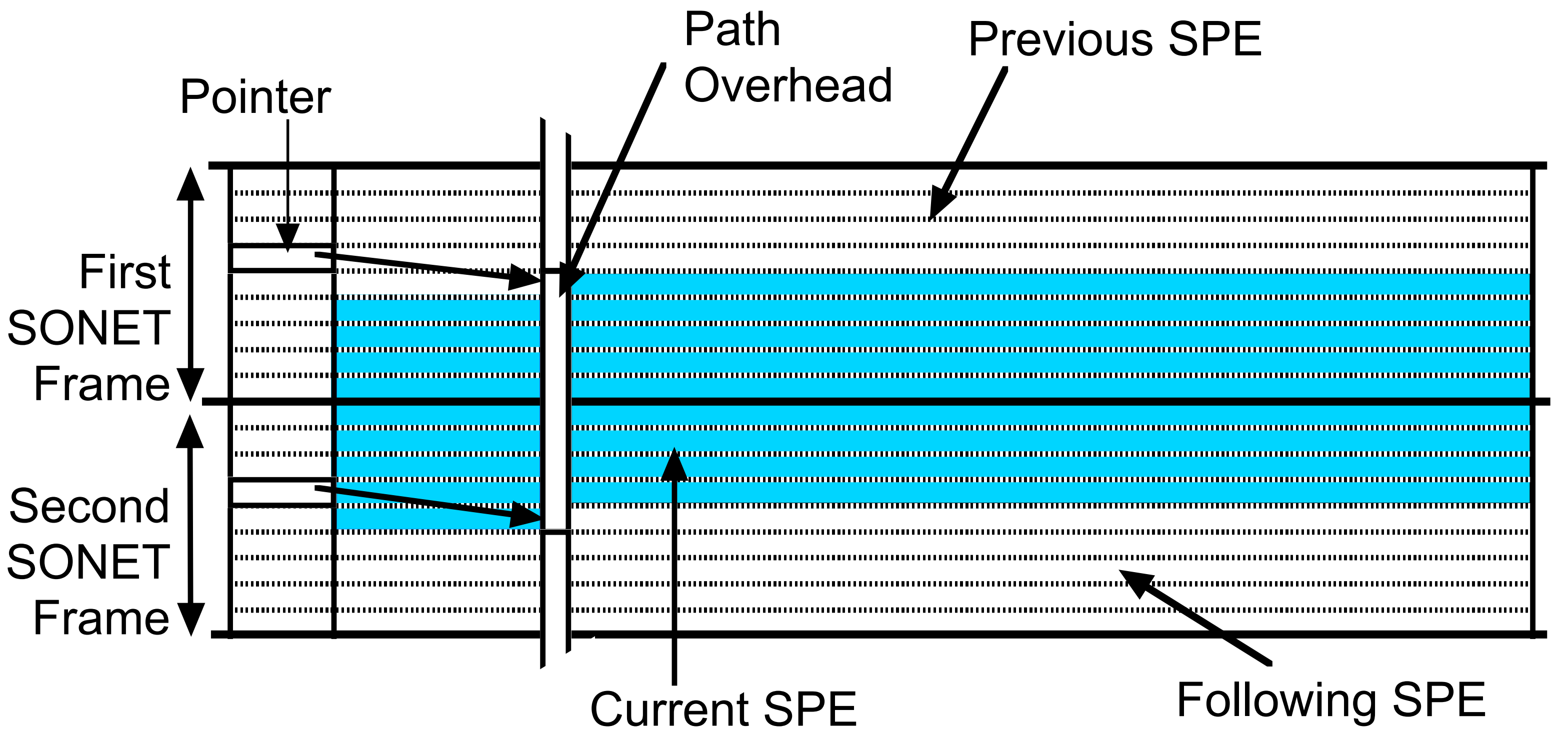
- The "section", "line" and "path" recognise levels in the SONET transmission hierarchy.
- Section and line deal with physical transmission of SONET frames and include considerable system and maintenance communication.
- The user payload (SPE – Synchronous Payload Environment) floats within the SONET frame. Usually an SPE will straddle two SONET frames, with its starting position indicated by the H1 and H2 octets of the Line Header.
- User circuits are time-multiplexed, each to the same position or positions in the SPE.
- Higher rates are obtained by interleaving *n* STS-1 frames to form an STS-*n* frame, still with 8000 STS-*n* frames per second (125µs period).
- The STS-3 frame is the basis of the ITU-T SDH transmission hierarchy.

Section Overhead	**Framing A1 = 0xF6**	**Framing A2 = 0x28**	**STS-ID C1**	Trace J1	Framing	For frame synchronisation
	BIP-8 B1	Orderwire E1	User F1	**BIP-8 B3**	STS-ID	identifies frames in MPX block
	DataCom D1	Data Com D2	Data Com D3	**Signal C2**	BIP	Byte parity over previous frame
Line Overhead	**Pointer H1**	**Pointer H2**	**Pointer H3**	Path Status G1	Orderwire	maintenance phone
	BIP-8 B2	APS K1	APS K2	User F2	Data Com	System Communication
	DataCom D4	Data Com D5	Data Com D6	**Multiframe H4**	Pointer	locates user SPE in SONET frame
	DataCom D7	Data Com D8	Data Com D9	Growth Z3	APS	protection, etc
	DataCom D10	Data Com D11	Data Com D12	Growth Z4		
	Growth Z Z1	Growth Z Z2	Orderwire E2	Growth Z5		

Section Overhead — Path Overhead

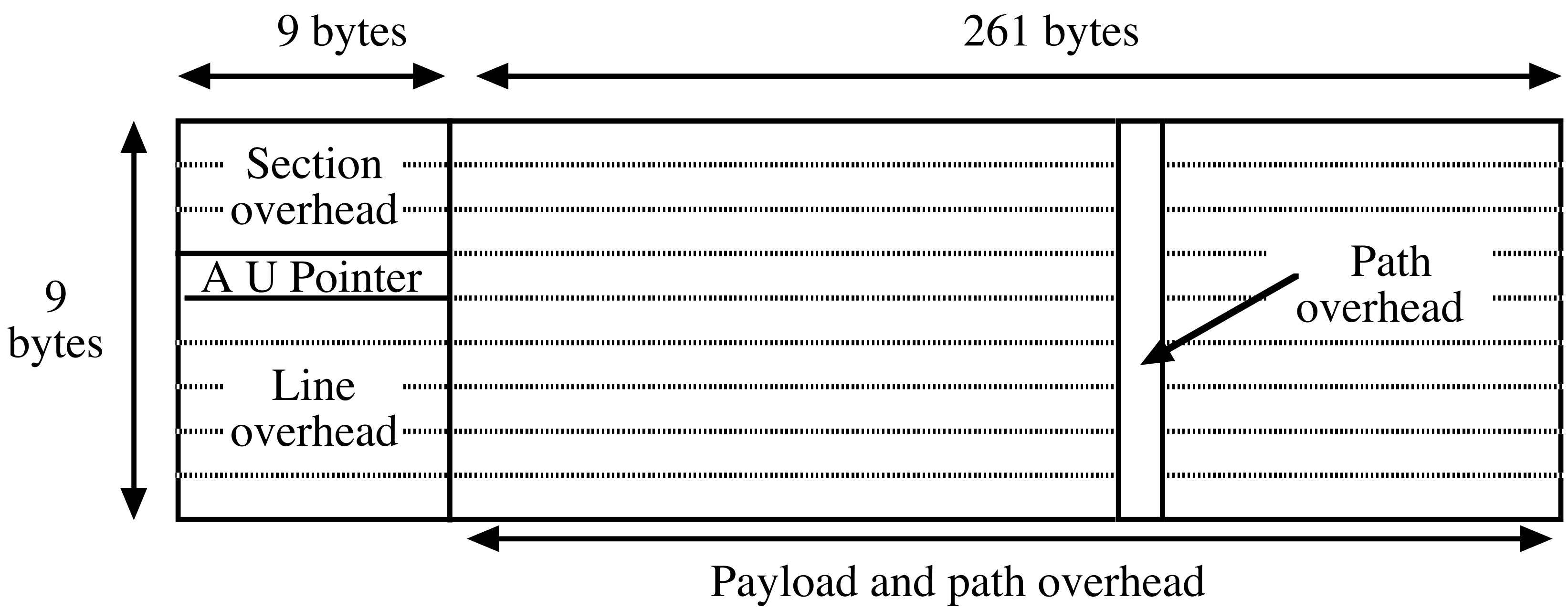
STS-1 Frame: overhead layouts

The actual data frame (the Synchronous Payload Envelope, or SPE; Administrative Unit, AU, in ITU terms) “floats” within and across transmission frames (SDH or SONET frames) to accommodate clock variations between the sender and the frame.



- The AU Pointer of the section header gives the byte offset within the physical frame of the start of the data frame (the first byte of the Path Overhead)
- The H1 and H2 Header bytes contain the pointer value, which can adjust by ±1 in each frame (or remain unchanged).
- If the pointer adjusts by +1, it means that the Synchronous Payload Envelope (SPE) has slipped *later* by 1 byte. This leaves a "hole" in the SONET frame, which is assumed to be in the SPE byte immediately after the H1, H2, H3 bytes.
- If the pointer adjusts by –1, the SPE has slipped to one byte *earlier* in the SONET frame. One byte is "squashed" out of the STS frame and is placed in the H3 byte of the Pointer.

SDH Frame format



S T M - 1 frame

This frame has a 3-fold interleave of each row with bytes sequentially from 3 multiplexed sub-frames. The first row starts out as (hexadecimal values)–

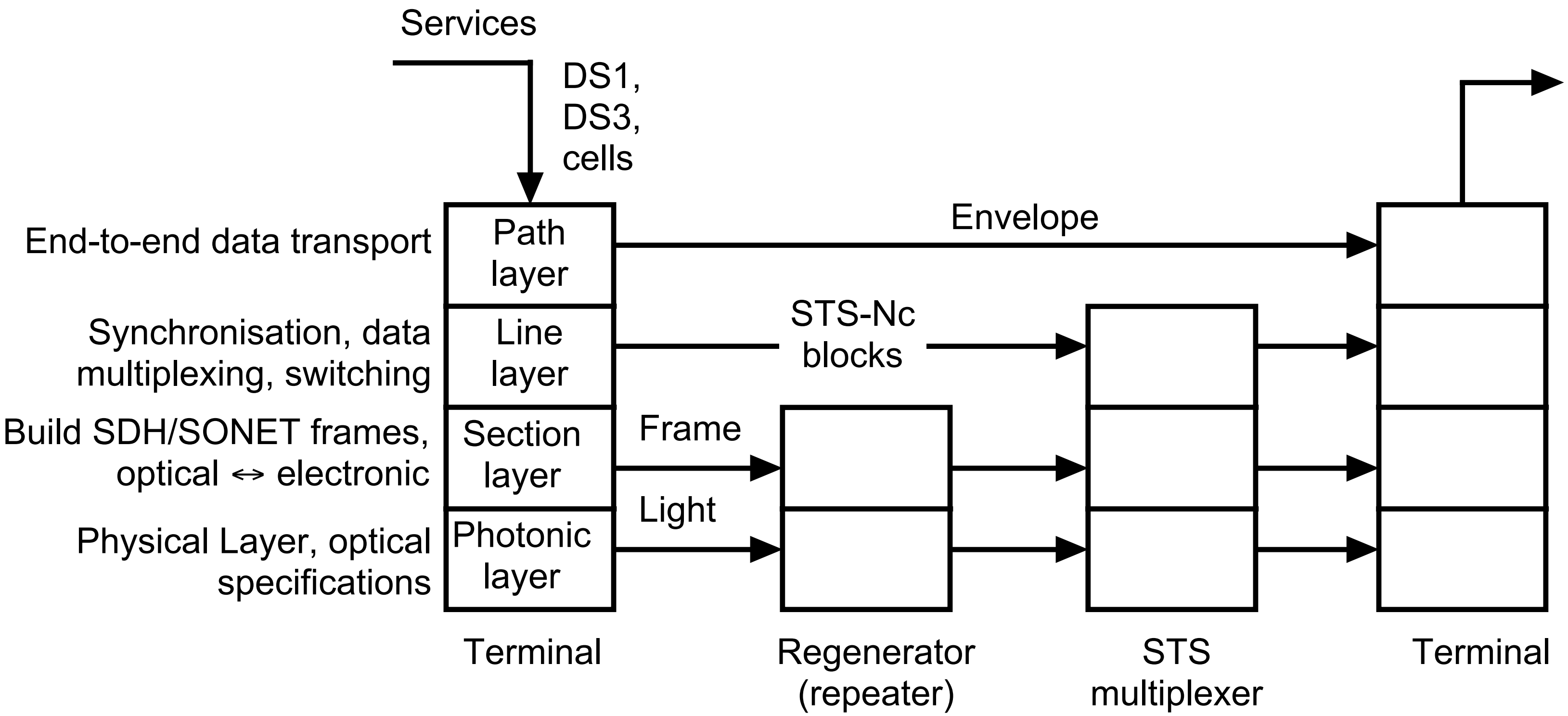
F6	F6	F6	28	28	28	01	02	03	ch0	ch1	ch2	ch0	ch1	ch2	…

The defined transmission speeds for the North American (SONET) standards and the ITU (CCITT or ITU-T) standards are shown in the table.

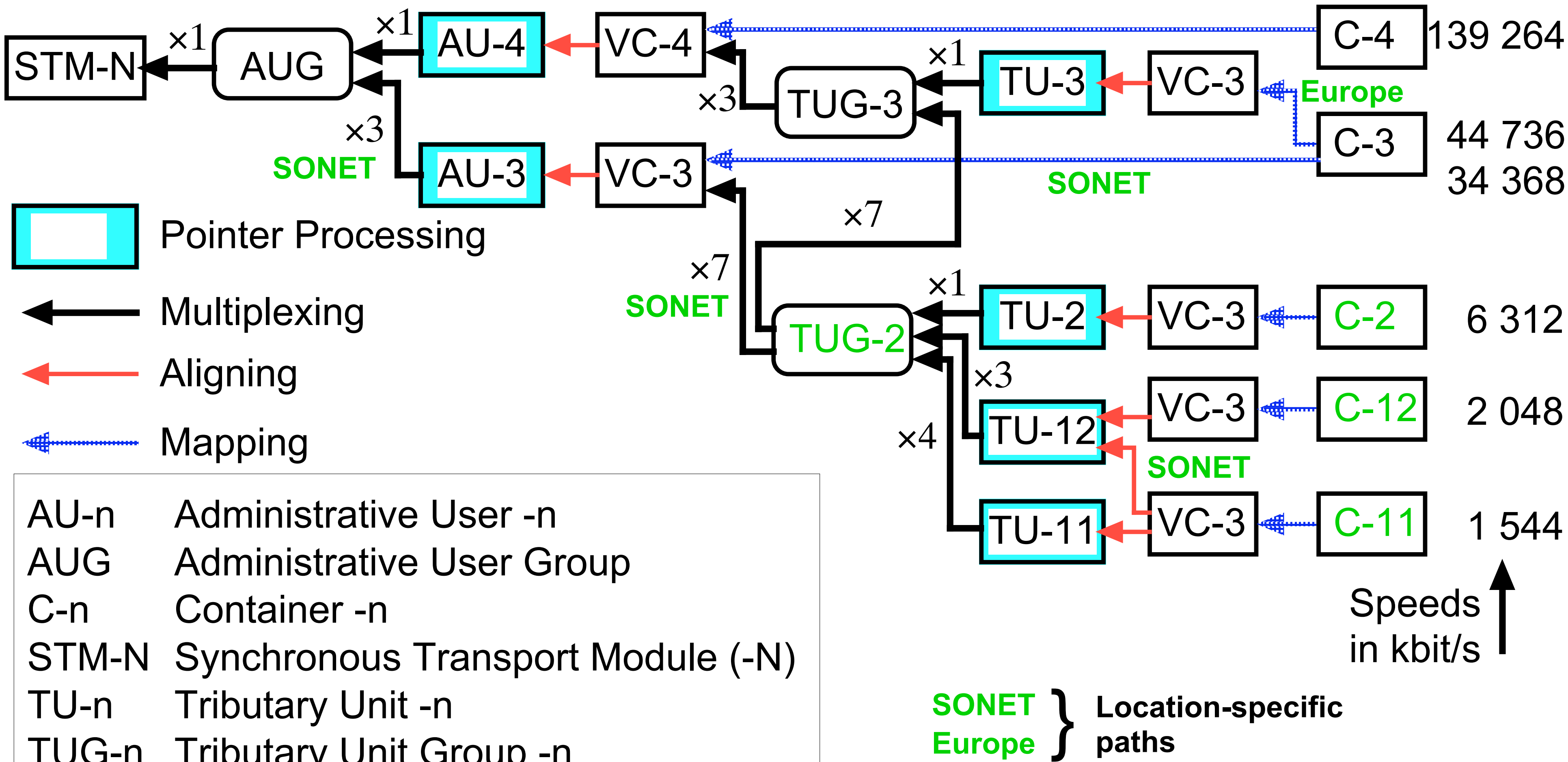
SONET designation	"Optical" designation	SONET rate Mbps	ITU designation	ITU rate Mbps
STS-1	OC-1	51.84		50.112
STS-3	**OC-3**	**155.52**	**STM-1**	**150.336**
STS-9	OC-9	466.56	STM-3	451.008
STS-12	**OC-12**	**622.08**	**STM-4**	**601.344**
STS-18	OC-18	933.12	STM-6	902.016
STS-24	OC-24	1,244.16	STM-8	1202.688
STS-36	OC-36	1,866.24	STM-12	1804.032
STS-48	**OC-48**	**2,488.32**	**STM-16**	**2405.376**
STS-192	**OC-192**	**9,953.28**	**STM-64**	**9621.504**

High capacity transmission rate assignments.

Nine bytes of each frame are interpreted by repeaters, 18 by multiplexers and 9 bytes (one column within the frame) by the end-to-end path logic.



SDH Multiplexing structure



AU-n	Administrative User -n
AUG	Administrative User Group
C-n	Container -n
STM-N	Synchronous Transport Module (-N)
TU-n	Tributary Unit -n
TUG-n	Tributary Unit Group -n
VC-n	Virtual Container -n

PPP on SDH

To avoid the overheads of using ATM as an intermediate step, we can put IP packets directly into an SDH frame.

See **RFC 2615** *PPP over SONET/SDH* and

RFC 1661 *PPP in HDLC-like Framing*.

The IP packet is checksummed, byte-stuffed, encapsulated and finally scrambled before being placed in the SDH frame.

Flag	Address	Control	Protocol	Info	Pad	FCS	Flag
01111110	11111111	00000011	16 bits	n bytes	m bytes	32 bits	01111110

Processing is performed in the order

IP → PPP → FCS generation → Byte stuffing
→ Scrambling → SONET/SDH framing.

Byte stuffing uses a Control Escape octet **01111101** (**0x7d**). The data byte is preceded by the Escape code and is then Exclusive-ORed with 0x20.

Only 2 values are byte-stuffed.

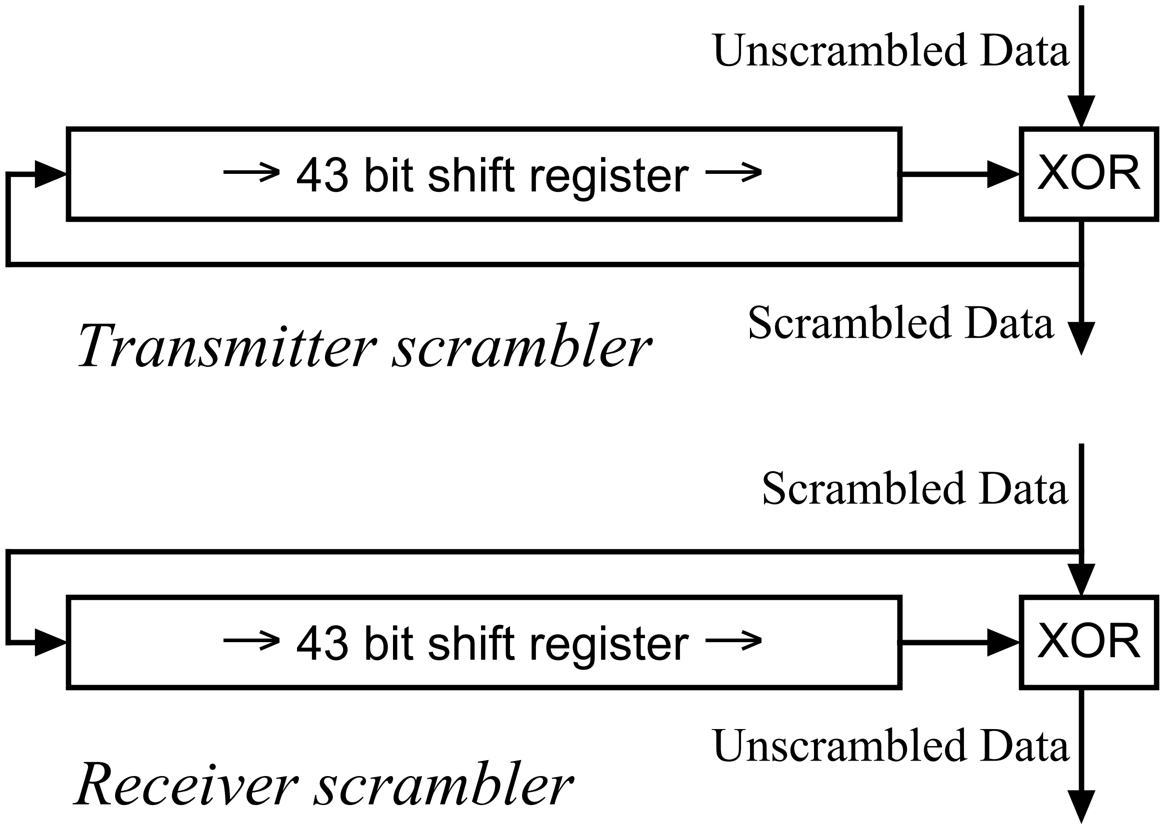
input	stuffed pair		meaning	binary stuffed result
0x7e	**0x7d**	**0x5e**	Flag Sequence	**01111101 01011110**
0x7d	**0x7d**	**0x5d**	Control Escape	**01111101 01011101**

Flag	Address	Control	Protocol	Info	Pad	FCS	Flag
01111110	11111111	00000011	16 bits	n bytes	m bytes	32 bits	01111110

- Flag (start and end) must be **`01111110`**
- Address MUST be **`11111111`**
- Control MUST be **`00000011`** (Unnumbered Information)
- Protocol is **`0x0021`** for IP (see RFC 1340 Assigned Numbers)
- The final stream is located by row within the SONET STS-SPE (or SDH Higher order VC), with the H4 (Payload Header) indicator zero.
- The Path Signal Label (C2 of the Payload Header) is 0x16 if the payload is scrambled and 0xCF with no scrambling.

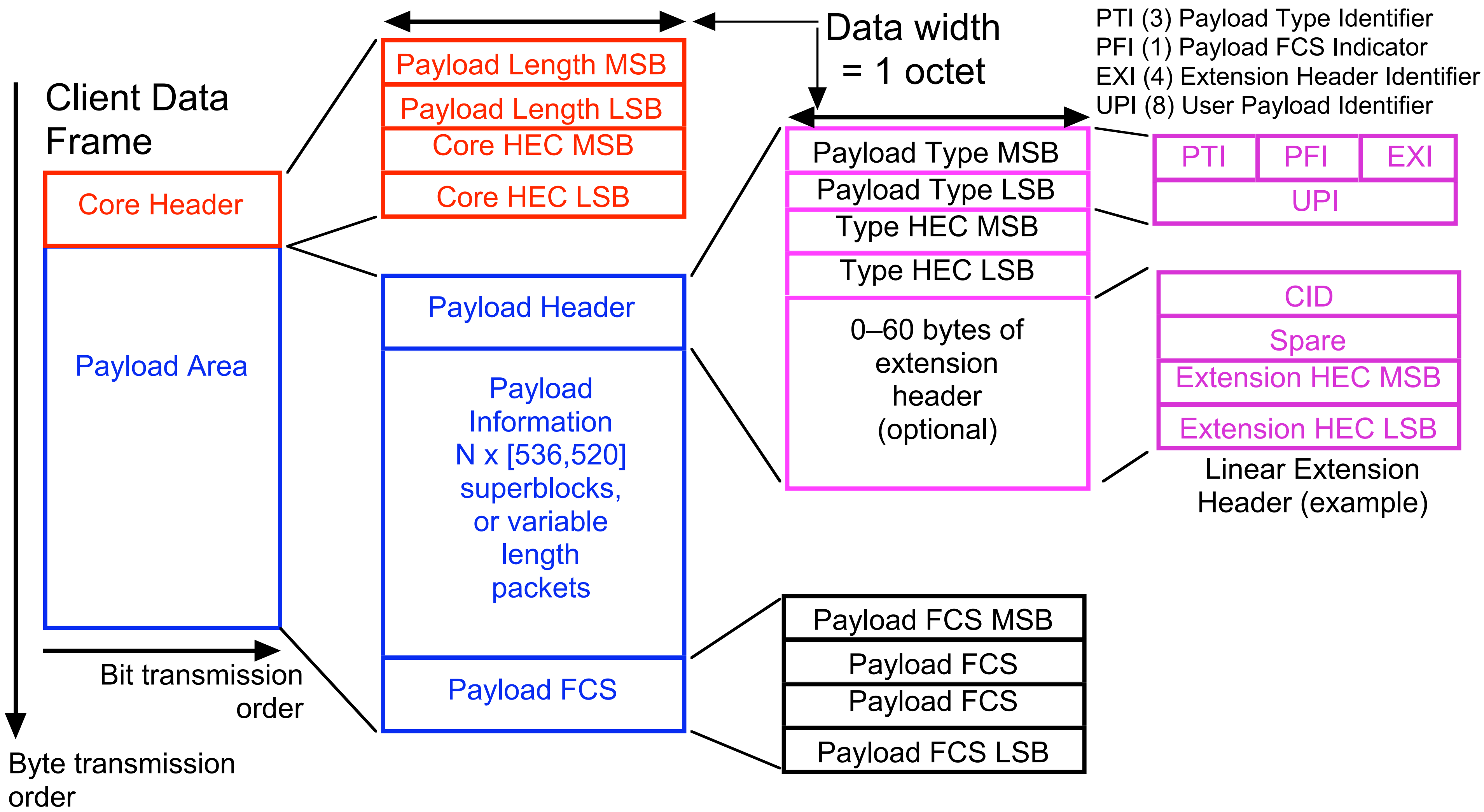
PPP over SDH/SONET, scrambling

Scrambling uses the generator polynomial $x^{43}+1$. The scrambler runs continuously, ignoring the overhead bytes.



Generic Framing Procedure (GFP)

- GFP is a new technique for carrying general protocols over SDH or SONET without the overheads and complexities of ATM.
- GFP provides full bit transparency
- GFP needs no special framing codes (HDLC flags, 8B10B K*xx*.*y*, etc)
- GFP is designed to support embedded or tunnelled protocols.



Generic Framing Procedure – Framing Structure

GFP Core Header

- At the lowest or core level a payload is preceded by a checksummed payload length, PLI – Payload Length Indicator.
- Much as with ATM, the receiver looks for a sequence of 4 octets satisfying a standard CRC-16 checksum. (With such small data the CRC-16 can do some error correction as well as error detection.)
- The first 2 octets are taken as a length indicator; the receiver expects to find another PLI after that distance.

Payload Length MSB
Payload Length LSB
Core HEC MSB
Core HEC LSB
Payload

GFP Payload

- The GFP payload has a payload header (checksummed), an optional extension header (also checksummed, cf Fibre Channel), the payload information and an optional payload checksum.
- The minimal message is very simple, with overheads added only as needed.

GFP Payload Header

PTI (3)	PFI (1)	EXI (4)	Type, FCS ind, Ext Hdr
UPI			User Payload ID
Type HEC MSB			16 - bit
Type HEC LSB			checksum

- PTI (3 bits) The type of frame. Now just user data and client management
- PFI (1 bit) Payload FCS indicator. = 1 if Payload FCS is present
- EXI (4 bits) Extension Header Identifier. Gives type of Extension Header (if present)
- UPI (8 bits) User Payload Identifier. Gives type of client data

GFP Payload

There are two modes for adapting client payloads into a GFP payload

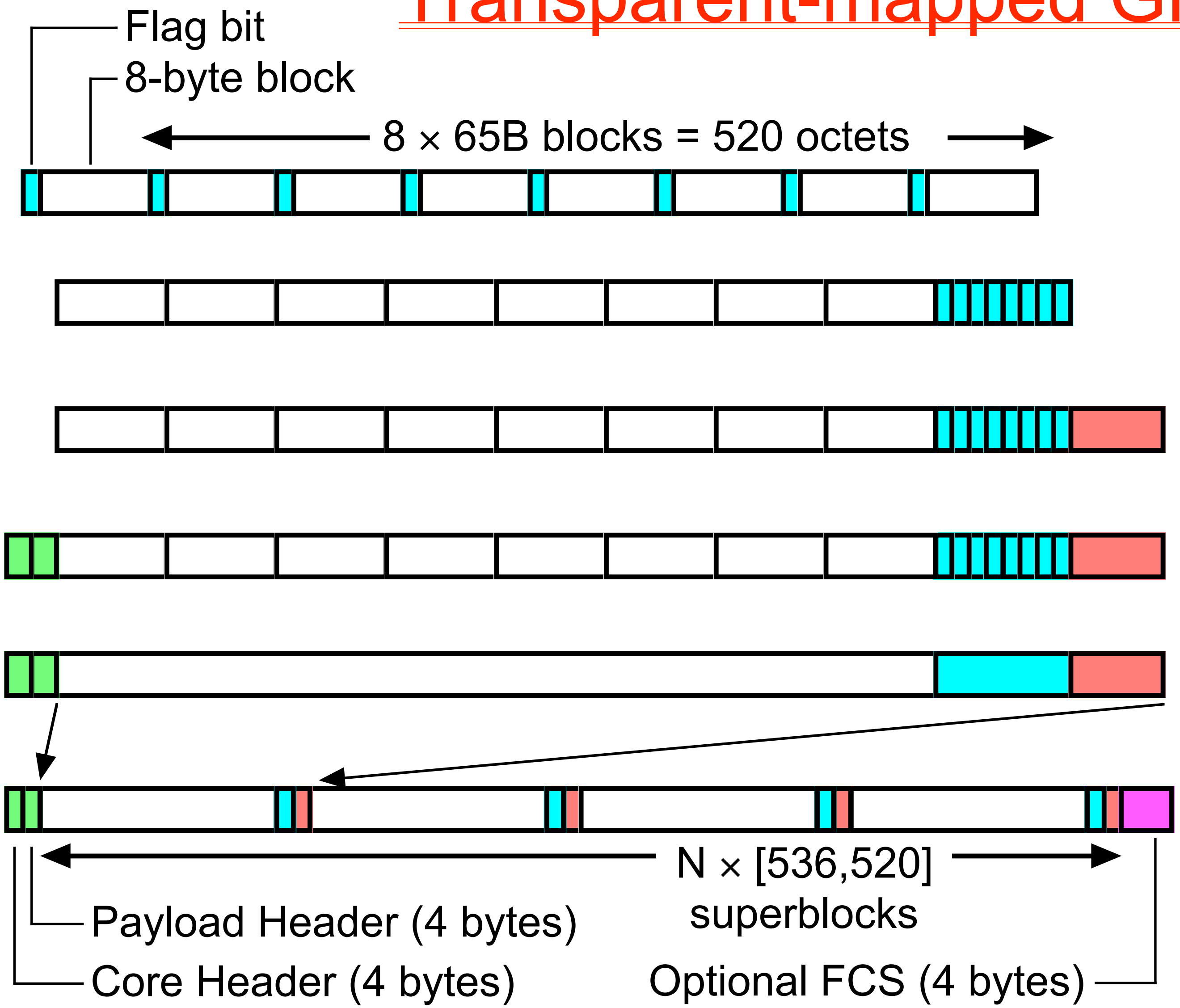
- Frame-mapped GFP (GFP-F) handles variable length packets with only 8-bit data, such as IP, or any HDLC framed PDU such as Frame Relay. The client frame is just inserted as the payload. The Payload Header structure is as given before, with the FCS optional.

PLI	cHEC	Payload header	Client PDU (PPP, IP, Ethernet, etc)	FCS (optional)
2 bytes	2 bytes	4 bytes	0 – 65,531 bytes	4 bytes

- Transparent-mapped GFP (GFP-F) encodes 8B/10B signals, where the 8-bit data is supplemented by "non-data" codes.
- The decoded codewords are grouped in 8s (64 bits) and each 64 bit block has a flag bit set if it contains a non-data code, giving a 64B/65B code.
- 8 of these blocks are grouped, with flag bits following and a 16-bit checksum added to form a [536,520] superblock.
- One or more superblocks form the the payload of the GFP frame

PLI	cHEC	Payload header	Client PDU recoded 8B/10B codewords	FCS (optional)
2 bytes	2 bytes	4 bytes	N x [536,520] superblocks	4 bytes

Transparent-mapped GFP structure



1. Group 8 × 65B blocks
2. Flag bits to end
3. Add CRC-16 to form [536,520] superblock
4. Add GFP core and payload headers
5. Scramble payload & header with $x^{43}+1$.
6. Form GFP frames with N × [536,520] superblocks.

Non-Data encoding, 1 (p 90 ff)

The rules for handling non-data depend on several features –

- Each 8-octet coding block can contain no more than 8 non-data codes
- The position of each non-data code can be specified by a 3-bit value
- As only 12 non-data codes are allocated, their coding needs only 4 bits
- Only 1 flag bit is needed to signal that a block has at least one control code
- The 8 flag bits for a 64-octet superblock can be held in a “flag byte”.

Non-Data encoding, 2

- If a block has non-data codes, bring all those codes to the front of the block and all data codes to the end, preserving the original order of each.
- Any block with 3 non-data codes and 5 data codes would be encoded

1 aaa C1	1 bbb C2	0 ccc C3	D1	D2	D3	D4	D5

- Each non-data cell –
  - starts with 1 bit, except for the last one.
  - has a 3-bit code (aaa, bbb, ccc) giving its original position in the block
  - has a 4-bit code (C1, C2, C3) giving its value (equivalent to K*xx*.*y*).
- This information allows the original block to be regenerated with only 1 bit of overhead.
- A CRC-16 error code is added as decoding is very sensitive to bit errors.

Virtual Concatenation

- The synchronous Digital Hierarchy is designed for fixed bandwidth allocations, seldom changed (some examples say weeks to reallocate bandwidth on a transcontinental link!)
- Computer data needs bandwidth to be allocated and released much more quickly
- Virtual concatenation defines a whole series of smaller paralleled circuits that can be multiplexed to give a higher overall bandwidth.
- Virtual concatenated paths do not even have to travel together, but can be split among different routes, recombining at the receiver.

Link Capacity Adjustment Scheme (LCAS)

- Virtual Concatenation is fine, but it can be difficult to change capacity, “on the fly”, for an active circuit.
- LCAS facilitates this adjustment, first exchanging LCAS control messages and then signalling through the Path Header H4 byte to announce when changes will take effect.