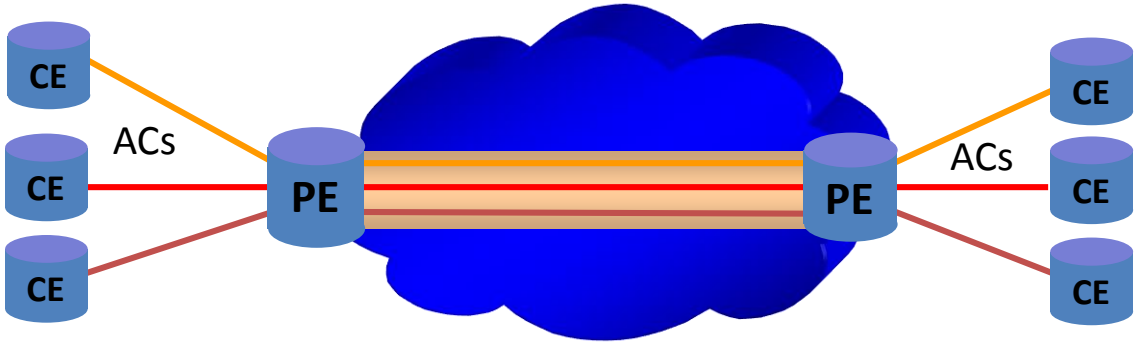


Pseudowires

Communications services

There are many kinds of customer traffic
(voice, video, file-transfer, BE data, web browsing, etc.)

Historically, **S**ervice **P**roviders built networks and sold *services*
optimized for a specific traffic type or types

Thus, we ended up with (too) many communications protocols

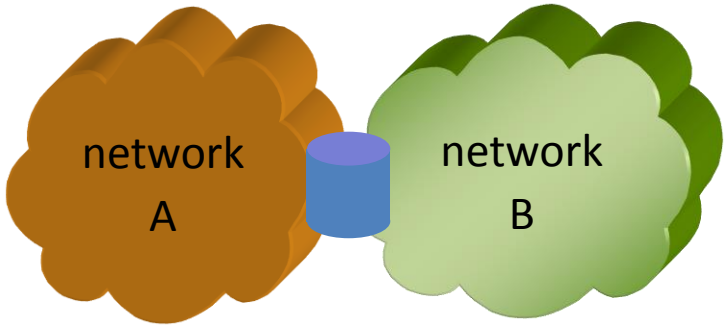
- IPv4, IPv6, Ethernet, MPLS, ATM, frame relay,
- E1 , E3, T1, T3, SDH, OTN, CPRI, PPP,
- fiber channel, Controller Area Network, Profinet, …

However, SPs with one type of network infrastructure
want to fully exploit it to carry all types of network traffic

Today, SPs prefer to run a single **P**acket **S**witched **N**etwork

Interworking

As transport using different network protocols
are often sold as different *services* (although the customer shouldn't care)
one frequently needs to connect networks using different protocols



This connection is called *interworking*

The protocol converter goes by various names :

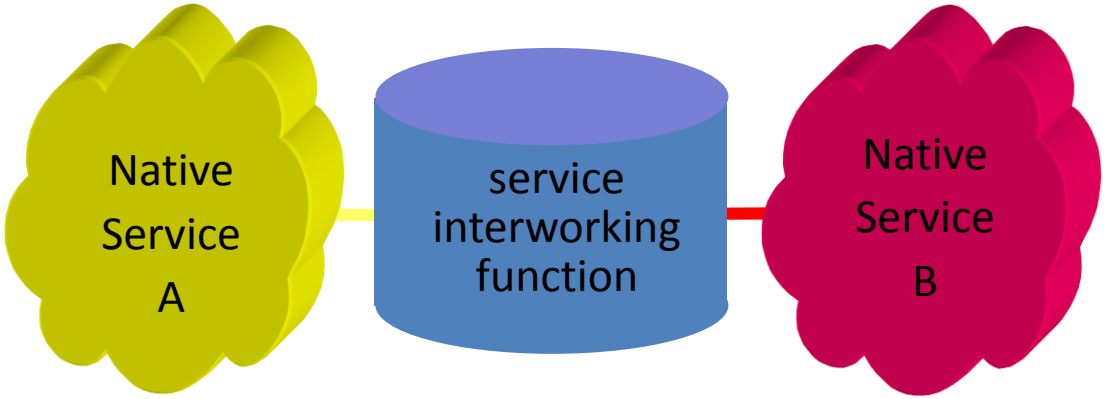
- interworking function (IWF)
- gateway (GW)
- edge device (e.g., PE)

Service Interworking

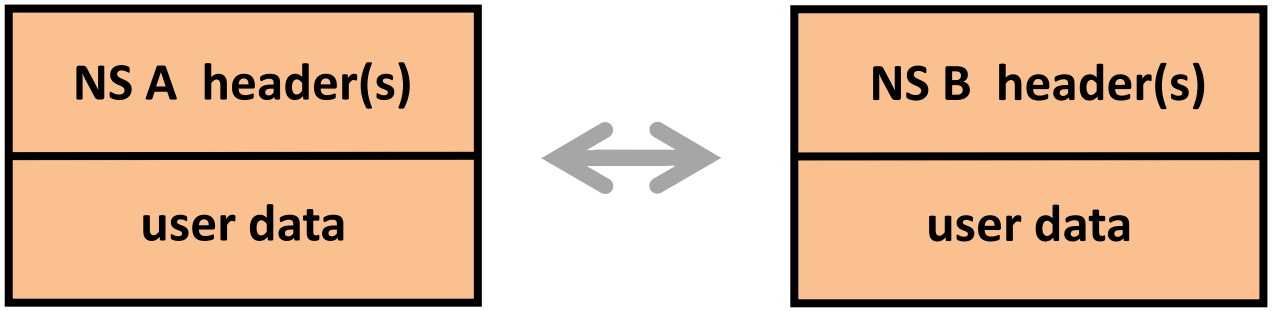
One way to connect two different networks

is by full conversion of the ***N**ative **S**ervice* formats

This is called *service interworking*



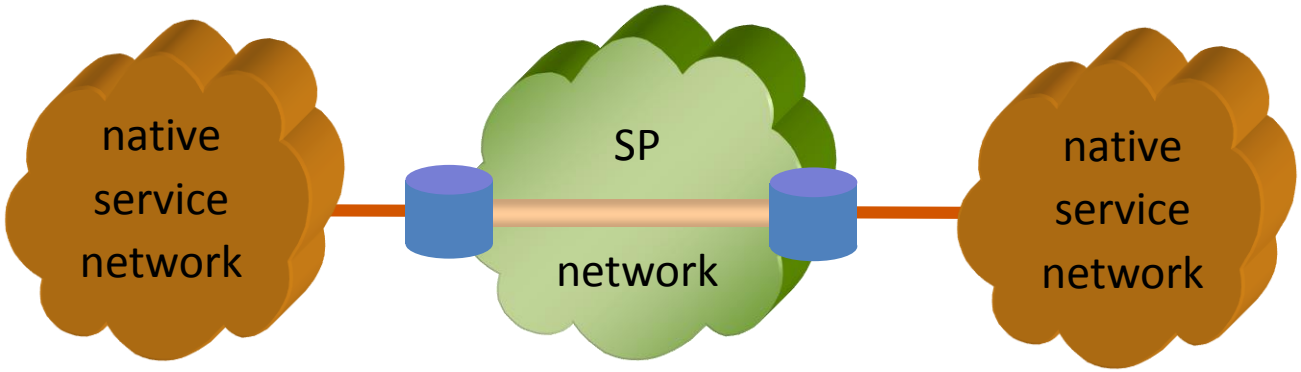
The service interworking IWF fully terminates the native services



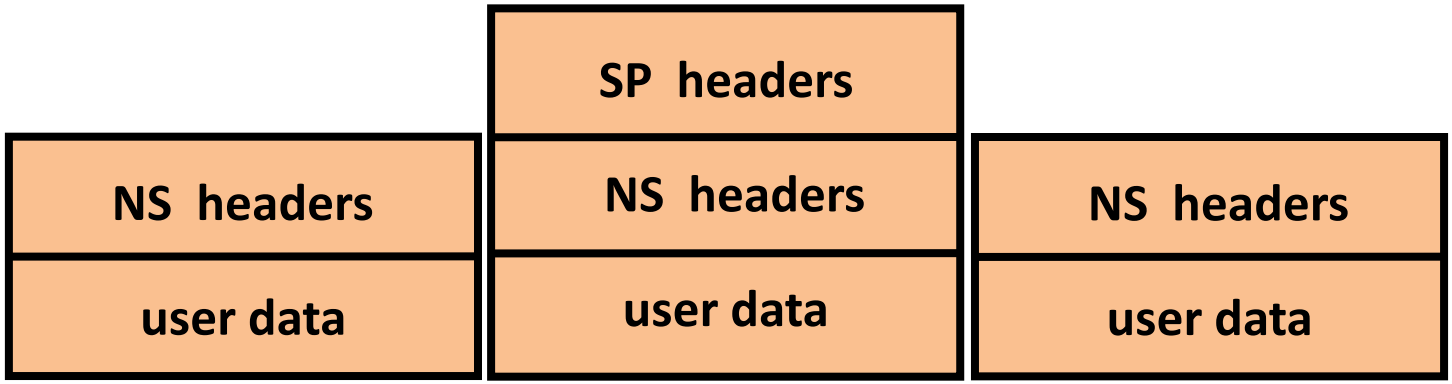
Tunneling

Network interworking (tunneling) is a simpler alternative that is applicable when both end-points sit on the same NS

For example, if we wish to interconnect two Ethernet LANs using an MPLS infrastructure network



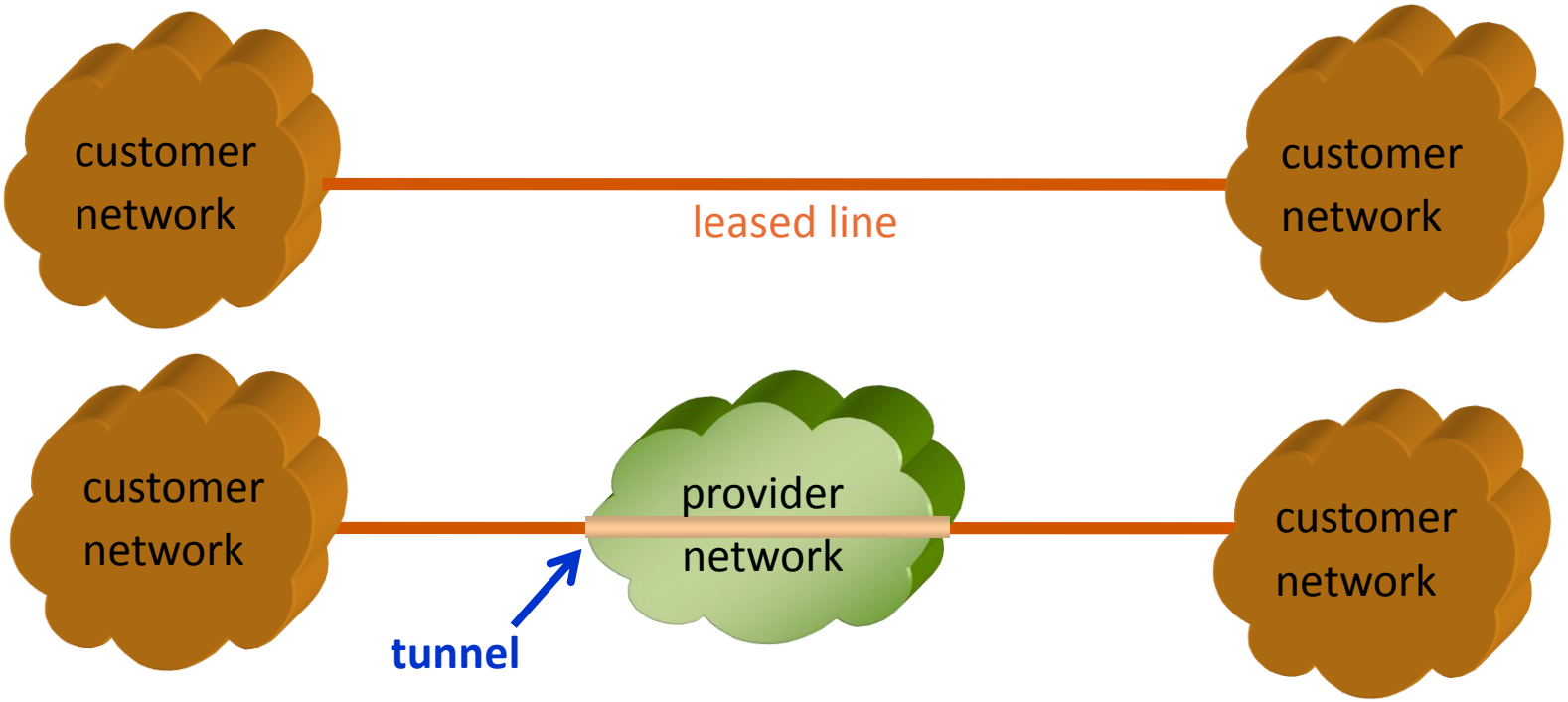
Note that the native service protocol is not terminated



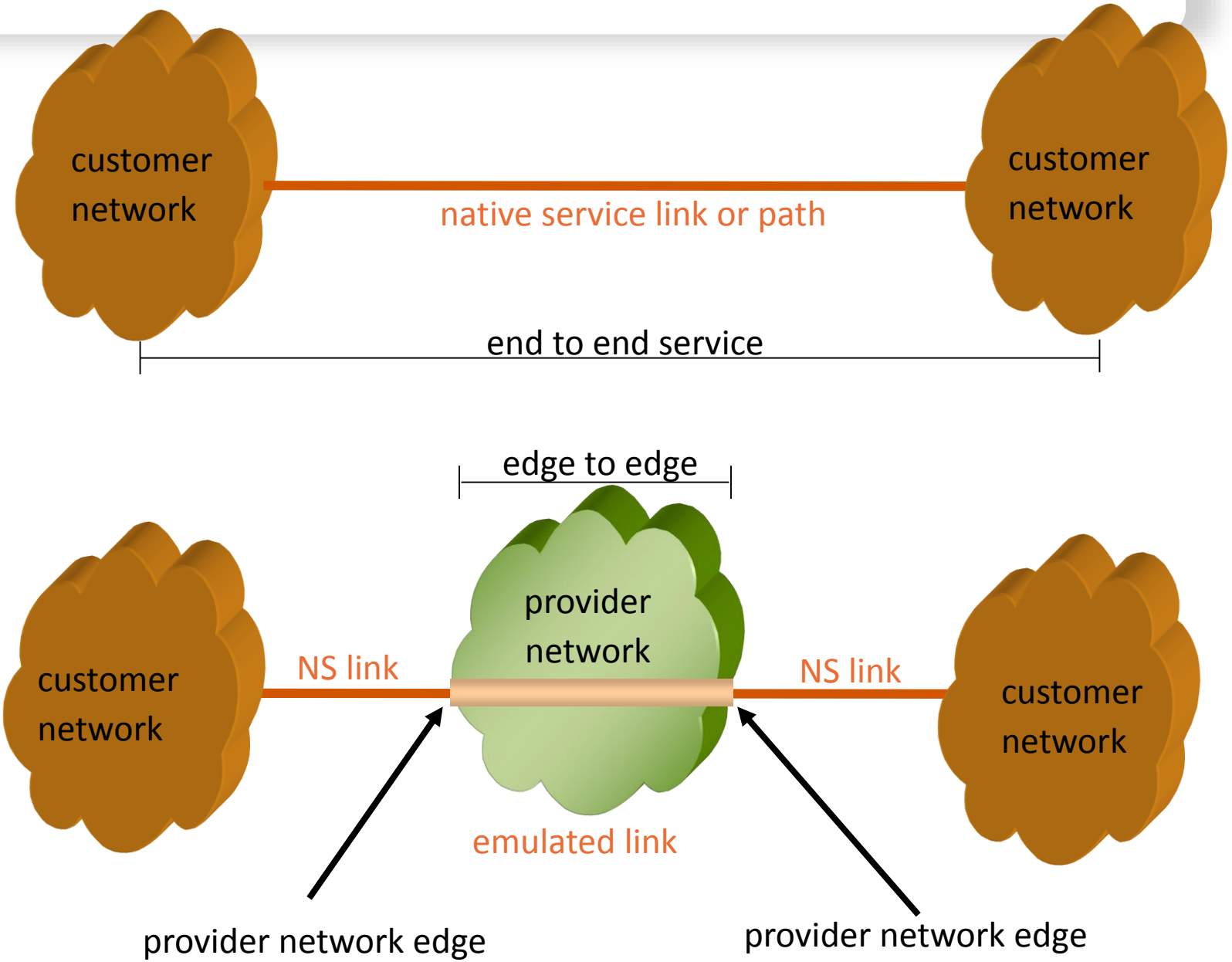
Emulation (wire service)

Since both ends of a tunnel are of the same native service
the tunnel seems to be transparent (a *virtual wire*)

In fact, the *goal* of network interworking (wire) service
is to *emulate* the native service
i.e., to be as *transparent* as possible
however, the emulation will not always be perfect !



Emulation Edge to Edge



Pseudowire

a Packet Switched Network (PSN) is
  - a network that forwards packets
  - e.g., IPv4, IPv6, MPLS, Ethernet*

a pseudowire (PW) is a mechanism to *tunnel* through a PSN

PWs have been defined for :
- MPLS
- L2TPv3 over IP

TDM PWs have also been defined for :
- Ethernet (only TDM PWs)
- UDP/IP (only TDM PWs)

PWs enable exploitation of a single converged network

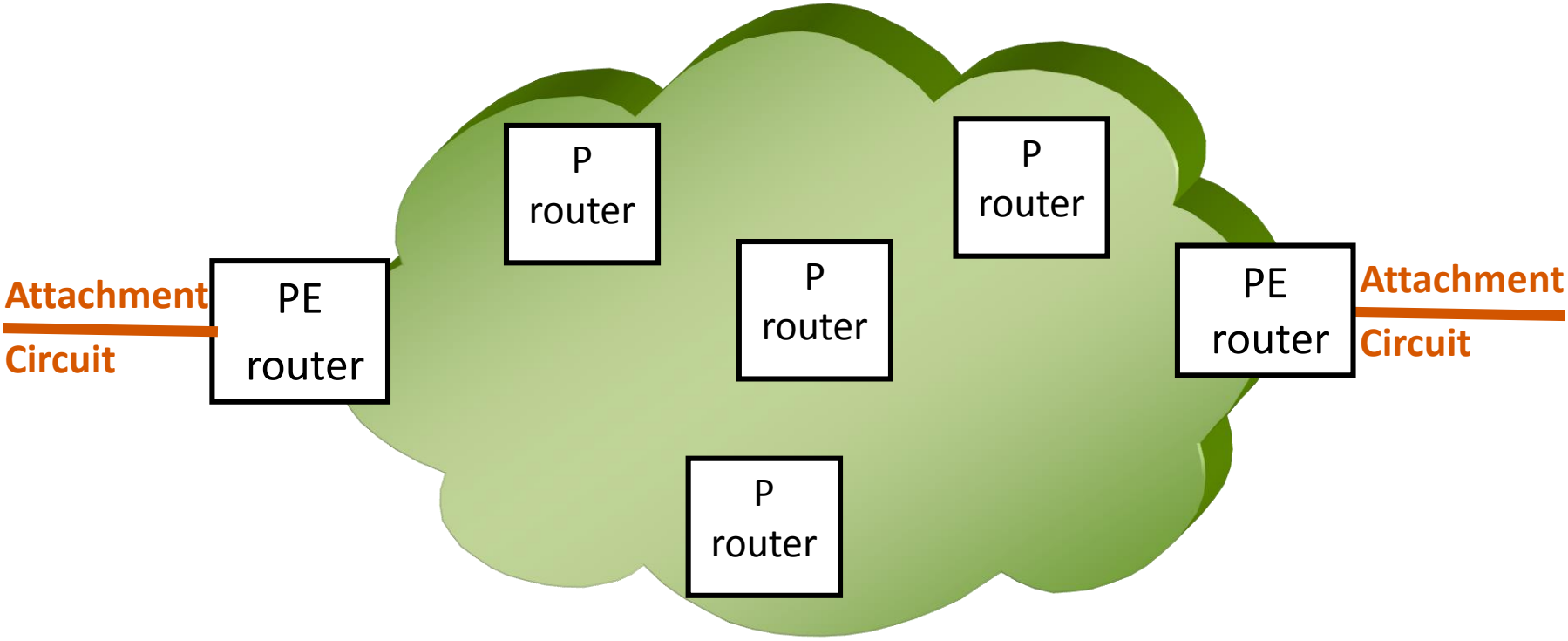
PWs are bidirectional (unlike MPLS LSPs)

PW architecture is an extension of VPN architecture

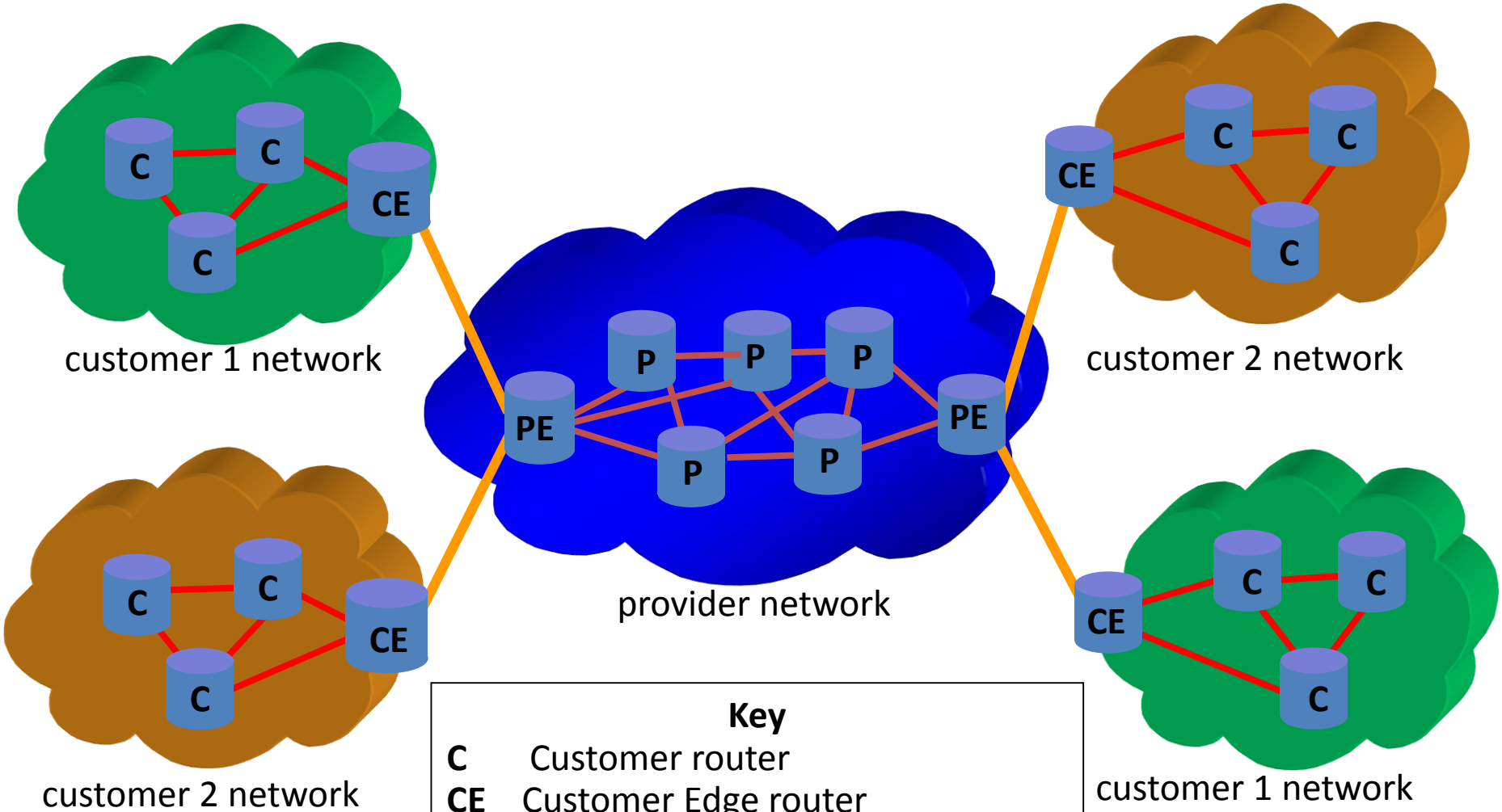
Provider Network Architecture

a provider network is composed of:

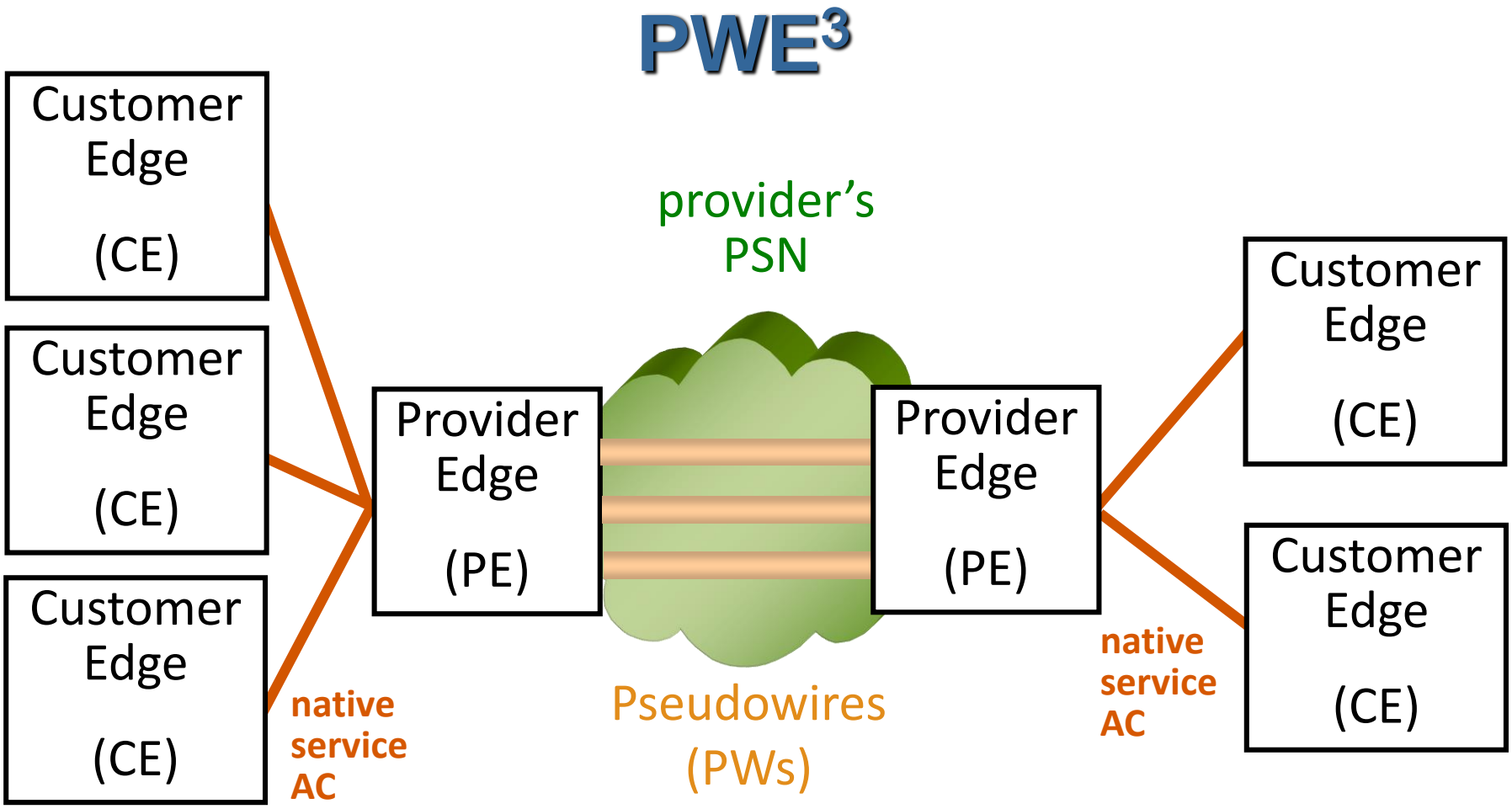
- **P**rovider routers or switches (P routers)
- **P**rovider **E**dge routers or switches (PE routers)



VPN architecture



Pseudowire Emulation Edge to Edge



Some PWE3 principles

- Edge-to-edge emulation and maintenance of PWs
  - tunnel creation and placement out of scope
  - PSN is responsible for differentiation between PWs
    e.g., in MPLS there is a PW label, in UDP/IP a PW port number
- Network interworking, not service interworking
- Native service emulation need not be perfect
  e.g., timing of a recovered TDM PW may not be the same as the timing of a native TDM circuit
  - imperfection documented in applicability statements
- Must not exert controls on underlying PSN
  - but diffserv, RSVP-TE can be used
- Must not redefine native service functionalities
  - use **N**ative **S**ervice **P**rocessing functions
- The PWE3 encapsulation consists of a 4-byte **C**ontrol **W**ord
  - initially each native service encap defined a different CW
  - RAD once had its own TDMoIP CW
  - perhaps my most important contribution to PWE3 was the single CW format
- Like MPLS packets, PW packets are not self-describing

MPLS PWs

Basic idea behind MPLS PWs :

LSRs forward based on *top of stack* label

LSR don't look inside the packet (e.g., for an IP address)

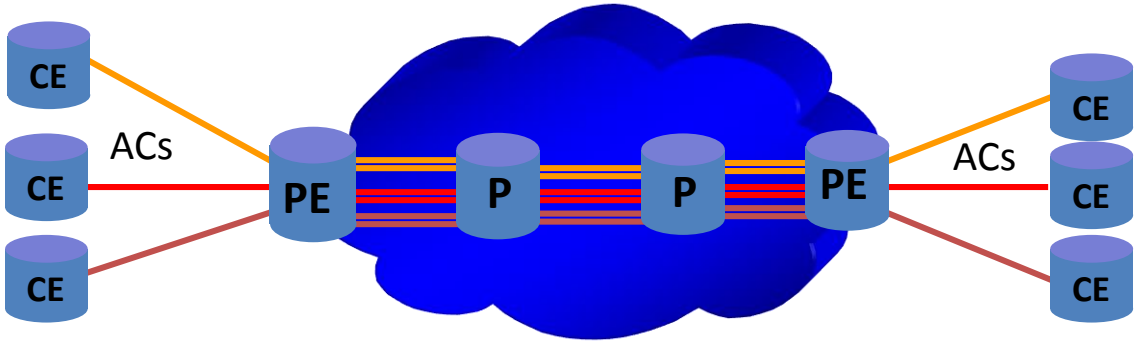
So we can carry anything inside the MPLS packet – not only IP !

The IETF PWE3 WG focused on MPLS (and the L2TPext WG on L2TPv3)

Why ?

- Emulated services often have QoS and TE requirements
  - IP is basically a "best effort" service (RSVP extensions not prevalent)
  - MPLS can provide TE guarantees (via RSVP-TE)
- IP provides no standard "bundle" multiplexing method
  - UDP/TCP ports provide *application* multiplexing
  - RTP uses ports in a nonstandard way
  - L2TP includes a multiplexing mechanism
  - MPLS label stack provides natural multiplexing method

Simplistic MPLS solution



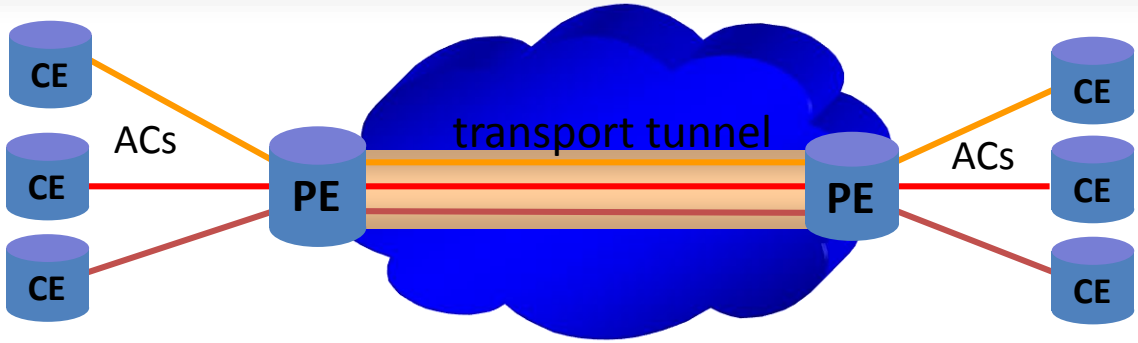
Each customer network mapped to pair of (unidirectional) LSPs

Each native packet/frame encapsulated with MPLS label

Scaling problem:

- requires huge number of LSPs in provider network overburdening P-router LFIBs
- P-routers need to be aware of customer network details

(Martini) Pseudowires



Transport MPLS tunnel set up between PEs

Multiple PWs may be set up inside tunnel

PW label is **B**ottom **o**f **S**tack (S=1)

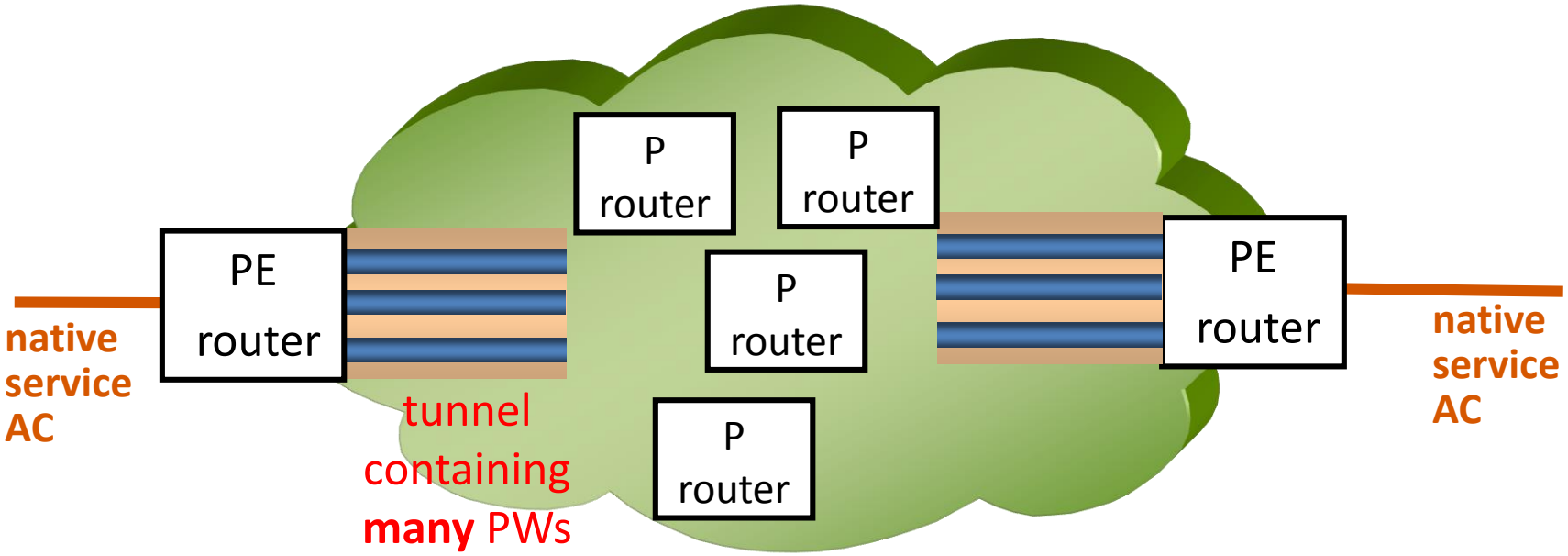
Native packet/frame encapsulated with 2 labels

P-routers are completely unaware of individual PWs

MPLS (tunnel) label
PW (inner) label
payload

	inner label	outer label(s)	MPLS-f
Translations:	interworking label	transport label(s)	ITU-T
	PW label	tunnel label(s)	IETF

Martini PWs (cont.)



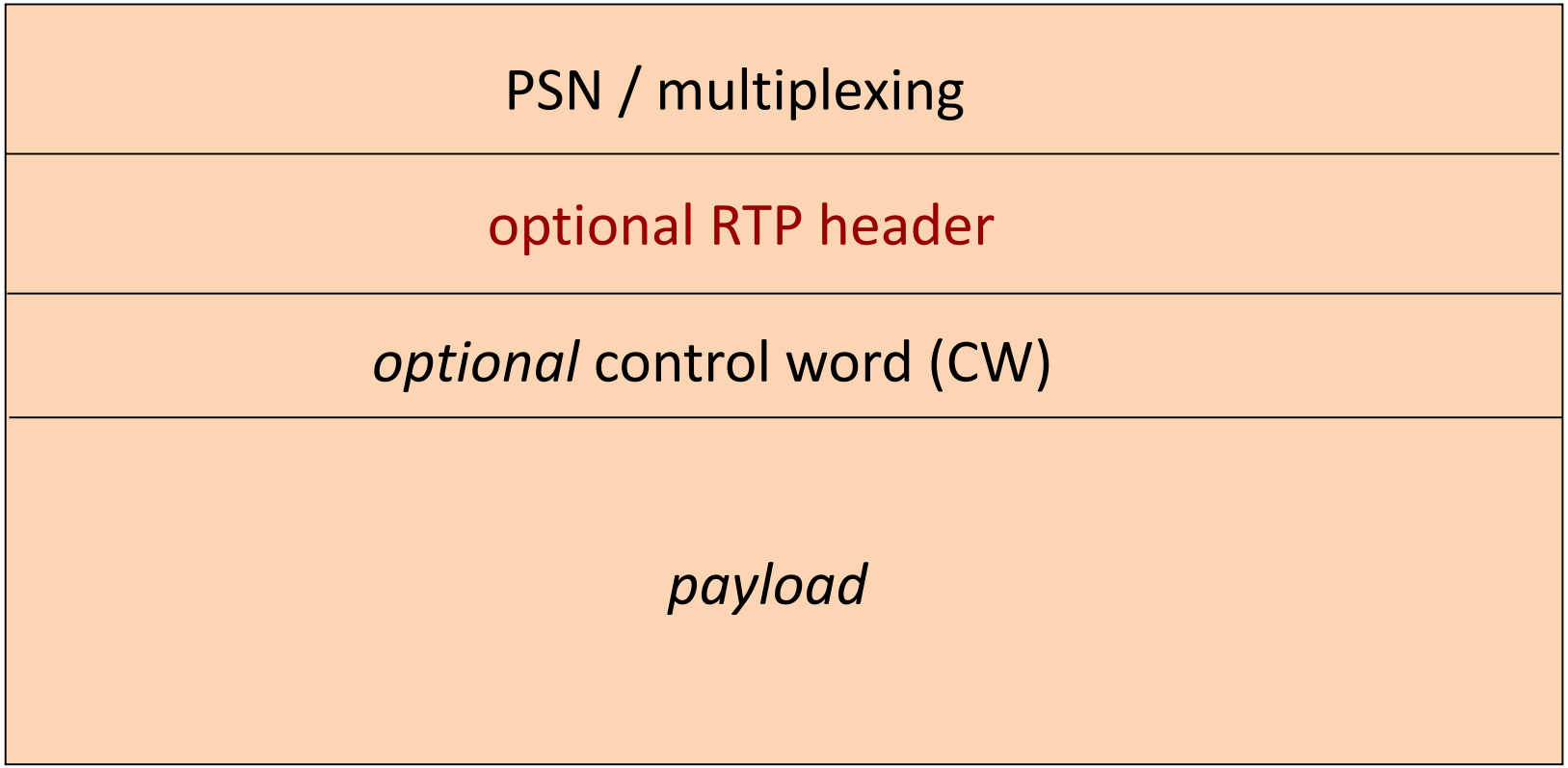
P routers use the tunnel label to forward the PW packet to the egress PE

The inner label is never used as an MPLS label

- no forwarding decisions based on it
- only used by PE to connect to the correct native service port

But this changes in **M**ulti**S**egment PWs

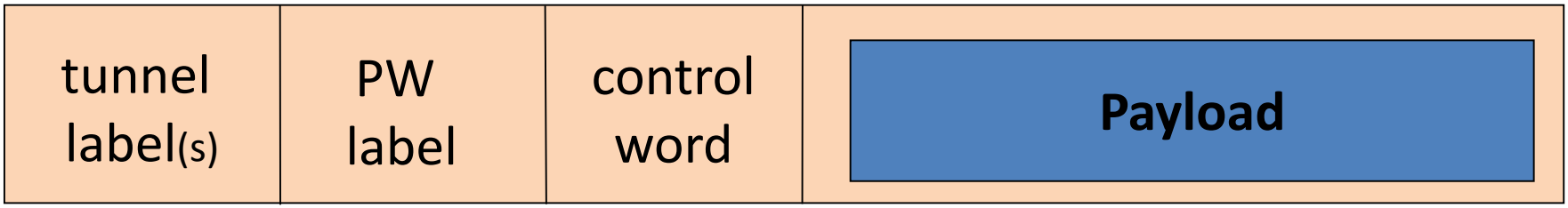
Generic PWE packet format



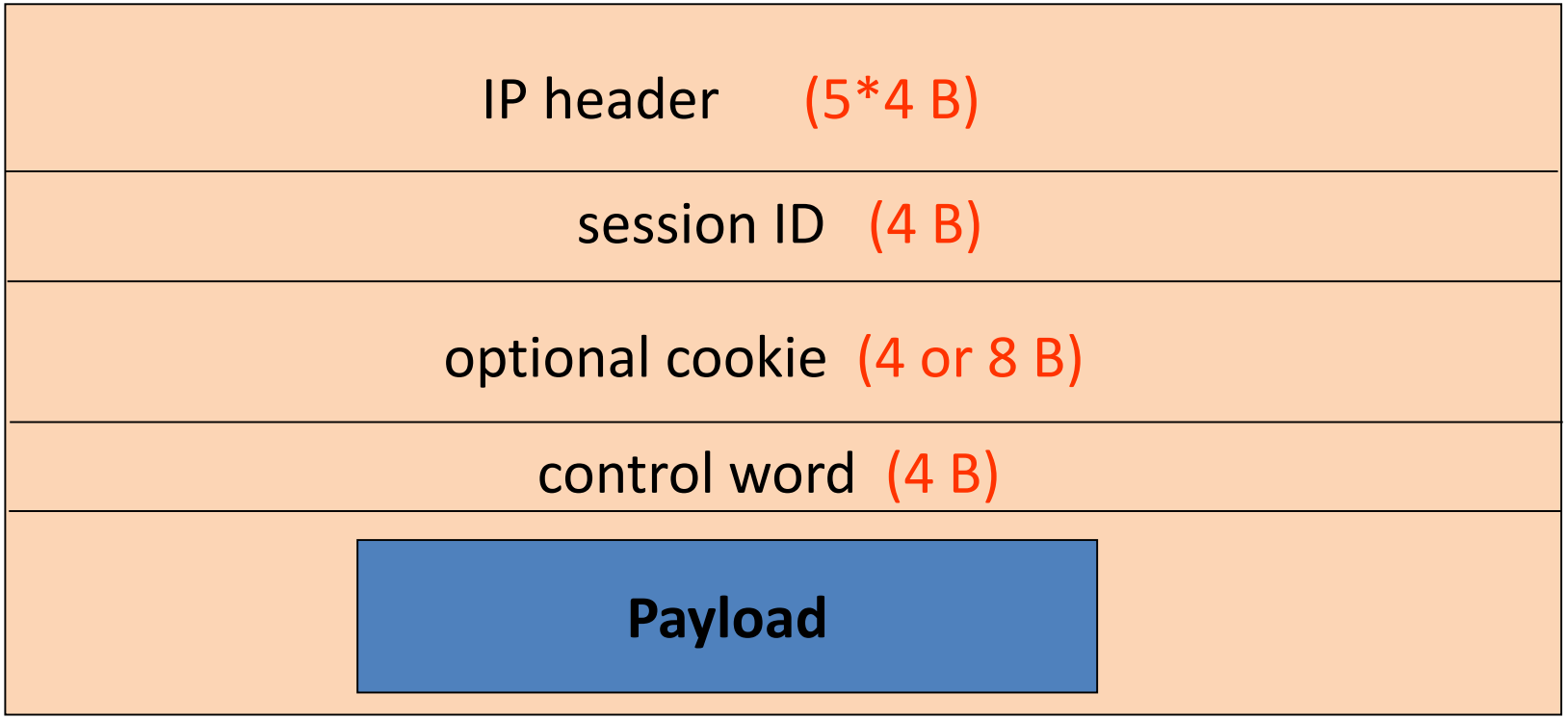
We call everything above the PW payload - the ***encap***

Example formats

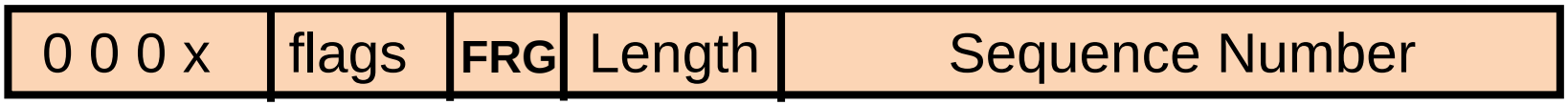
MPLS PSN



L2TPv3 PSN



Standard PWE Control Word



0 0 0 x (first nibble)

- 0000 for PW user traffic
- 0001 for PW Associated Channel (ACh), e.g., for VCCV OAM

(note that were this IP, this nibble would be 0100 or 0110)

Some ECMP mechanisms rudely recurse down the label stack and peek at the first nibble to differentiate between IP and PW

Flags (4 b)

- only a few native service encaps define flags
- used to transport native service fault indications

FRG

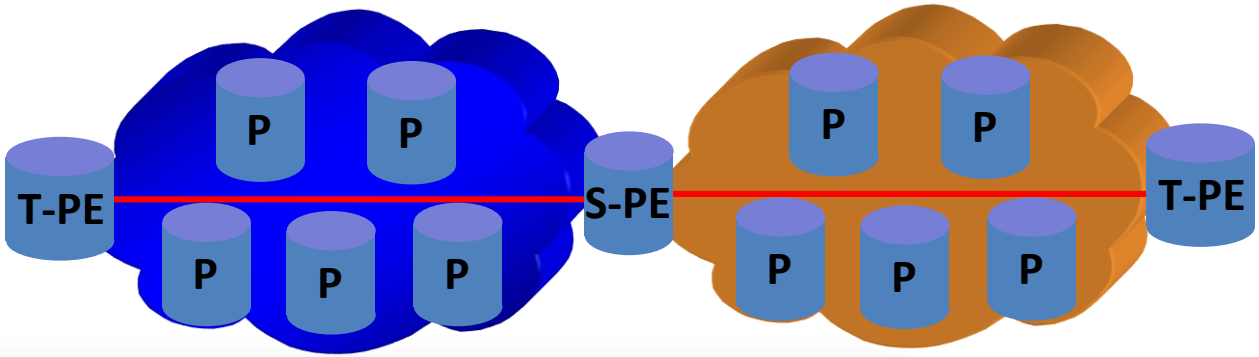
- some native service encaps use to indicate payload fragmentation
  - 00 = unfragmented 01 = 1st fragment
  - 10 = last fragment 11 = intermediate fragment

Length (6 b)

- used when packet may be padded by lower layers

Sequence Number (16 b)

- may be used to detect packet loss / misordering



Some PW-specific mechanisms

Associated Channel and VCCV

PWs have an **A**ssociated **Ch**annel for OAM, etc.

Main use is for **VC C**onnectivity **V**erification - VCCV

VC is an *old* name for PW

CV is an *incorrect* name for **C**ontinuity **C**heck

VCCV runs inside PW (same PW label) so that it *fate shares*

Different Control Word format

0 0 0 1	VER	RES=0	Channel Type (0x07-raw 0x21-IPv4 0x57-IPv6)

Inside VCCV several different OAM mechanisms may be used:

- ICMP
- LSP ping (RFC 4379)
- BFD (one-way or echo)

The ACh has been extended for MPLS-TP (to become the GACh)

VCCV Channel Types

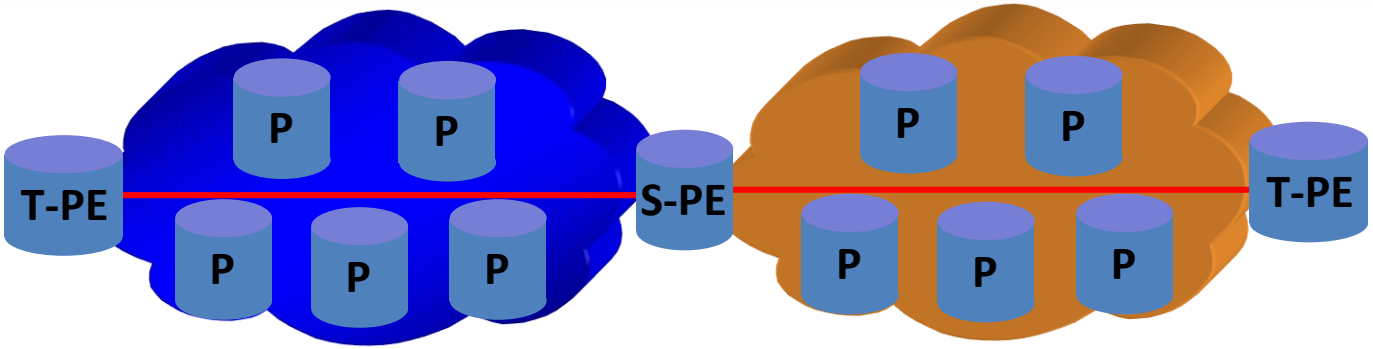
In an old proposal, there was an OAM PW (with a special label) that monitored all PWs in the same tunnel

The PW ACh fate-shares with PW traffic
so how do we recognize a VCCV packet ?

RFC 5085 defines 4 **C**ontrol **C**hannel types

Type 1	*in-band* VCCV (only when there is a CW) CW first nibble 0001
Type 2	*out-of-band* VCCV MPLS Router Alert Label (label=1) above the PW label
Type 3	TTL expiry (TTL=1 at destination PE)
Type 4	new VCCV channel type **G**eneric **A**Ch **L**abel (label=13) under the PW label

MS-PWs



Single-Segment PW (SS-PW) requires the PEs to *see* each other

With multiple PSN domains this may not be the case

MultiSegment PW (MS-PW)

Terminal-PEs interconnect via stitching-PE

PW label becomes a **true** MPLS label (switching, swapping)

When there is more than one S-PE

need to ensure that the 2 LSPs traverse the same one

FAT PWs

IP ECMP functions by hashing specific key fields in the IP header

MPLS ECMP hashes on the label stack
- but since the BoS is the closest to the individual flows
- it gives the finest granularity

There is a proposal for allowing a special *entropy label* in MPLS

The **F**low **A**ware **T**ransport PW mechanism adds a *flow label*
- below the PW label (NB the PW label is no longer BoS)
- label random (high entropy) but not a reserved label
- true flows must be consistently mapped to the same flow label
- TTL=1 so that it is never accidentally used for label switching

Unlike my proposal for PW bonding
- there is only a single PW
- but standard MPLS ECMP causes load balancing

PW redundancy

Since PWs are used by SPs for transport
Automatic Protection Switching may be needed

For SS-PWs, the MPLS network protects the tunnels (e.g., FRR)

But how do we protect against PE or AC failure ?

For MS-PWs how do we protect against S-PE failure ?

The basic idea is

- dual homing a CE to >1 PE
- and setting up primary and secondary PWs (control protocol extensions)
- monitoring PWs using VCCV
- triggering protection switch using (new bits in) PW status messages

TDM PWs

What is TDM ?

By TDM here we mean synchronous transport at

- one of the PDH (G.702) rates
- one of the SDH (G.707) rates

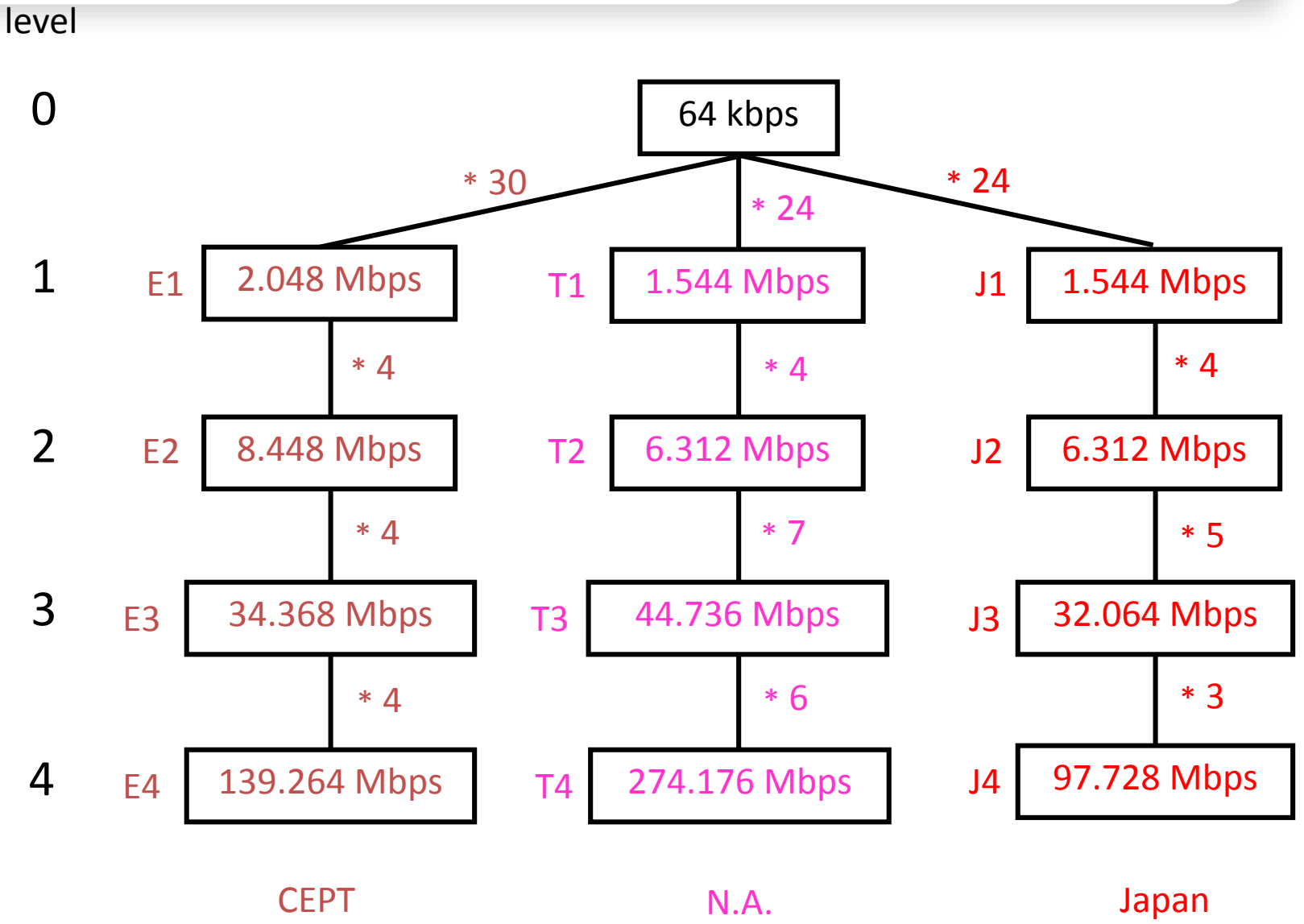
TDM networks can themselves carry

- telephony traffic
- data traffic
- video

Native TDM networks

- circuit switching ensures signal integrity
- very high reliability (“*five nines*”)
- low delay and no noticeable echo for telephony
- timing information transported over the network
- mature signaling protocols (over 3000 features)

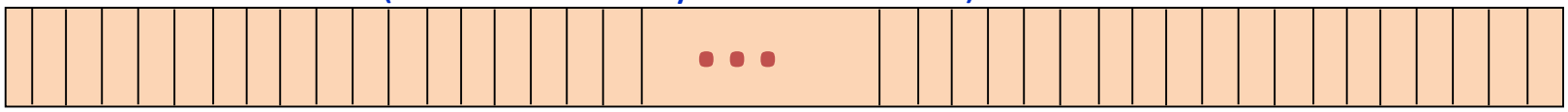
PDH rates


level
0
1
2
3
4
64 kbps
* 30
* 24
* 24
E1
2.048 Mbps
T1
1.544 Mbps
J1
1.544 Mbps
* 4
* 4
* 4
E2
8.448 Mbps
T2
6.312 Mbps
J2
6.312 Mbps
* 4
* 7
* 5
E3
34.368 Mbps
T3
44.736 Mbps
J3
32.064 Mbps
* 4
* 6
* 3
E4
139.264 Mbps
T4
274.176 Mbps
J4
97.728 Mbps
CEPT
N.A.
Japan

TDM Structure

handling of TDM depends on its structure

unstructured TDM (TDM = arbitrary stream of bits)

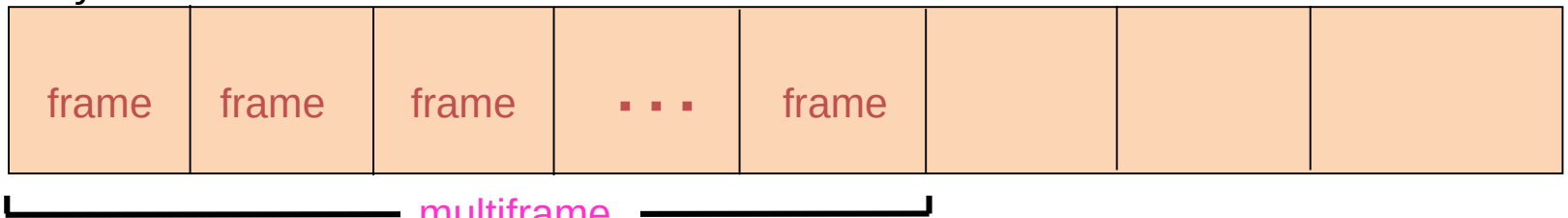


structured TDM

framed (8000 frames per second)

channelized (single byte timeslots)

multiframed



TDM transport types

Structure-agnostic transport (SAToP – RFC4553)

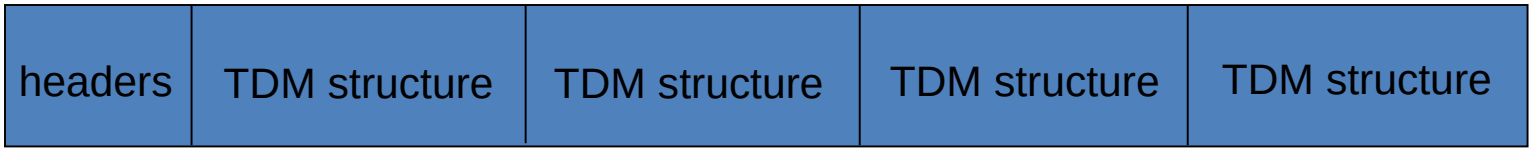
- for unstructured TDM
- even if there is structure, we ignore it
- simplest way of making payload
- OK if network is well-engineered

Structure-aware transport (TDMoIP, CESoPSN)

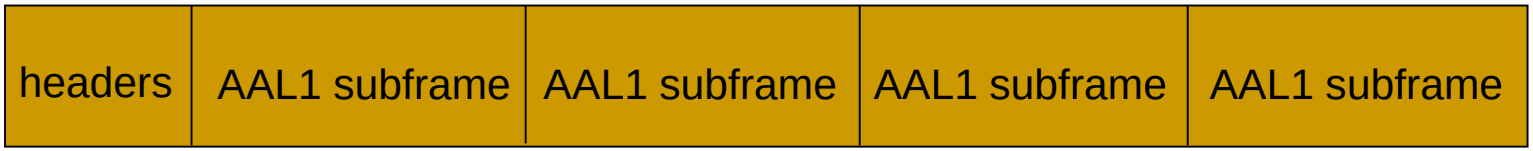
- take TDM structure into account
- must decide which level of structure (frame, multiframe, …)
- can overcome PSN impairments (PDV, packet loss, etc)

Structure aware encapsulations

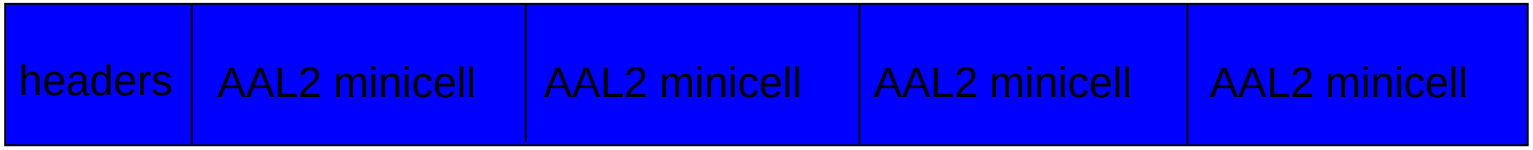
Structure-locked encapsulation (CESoPSN)



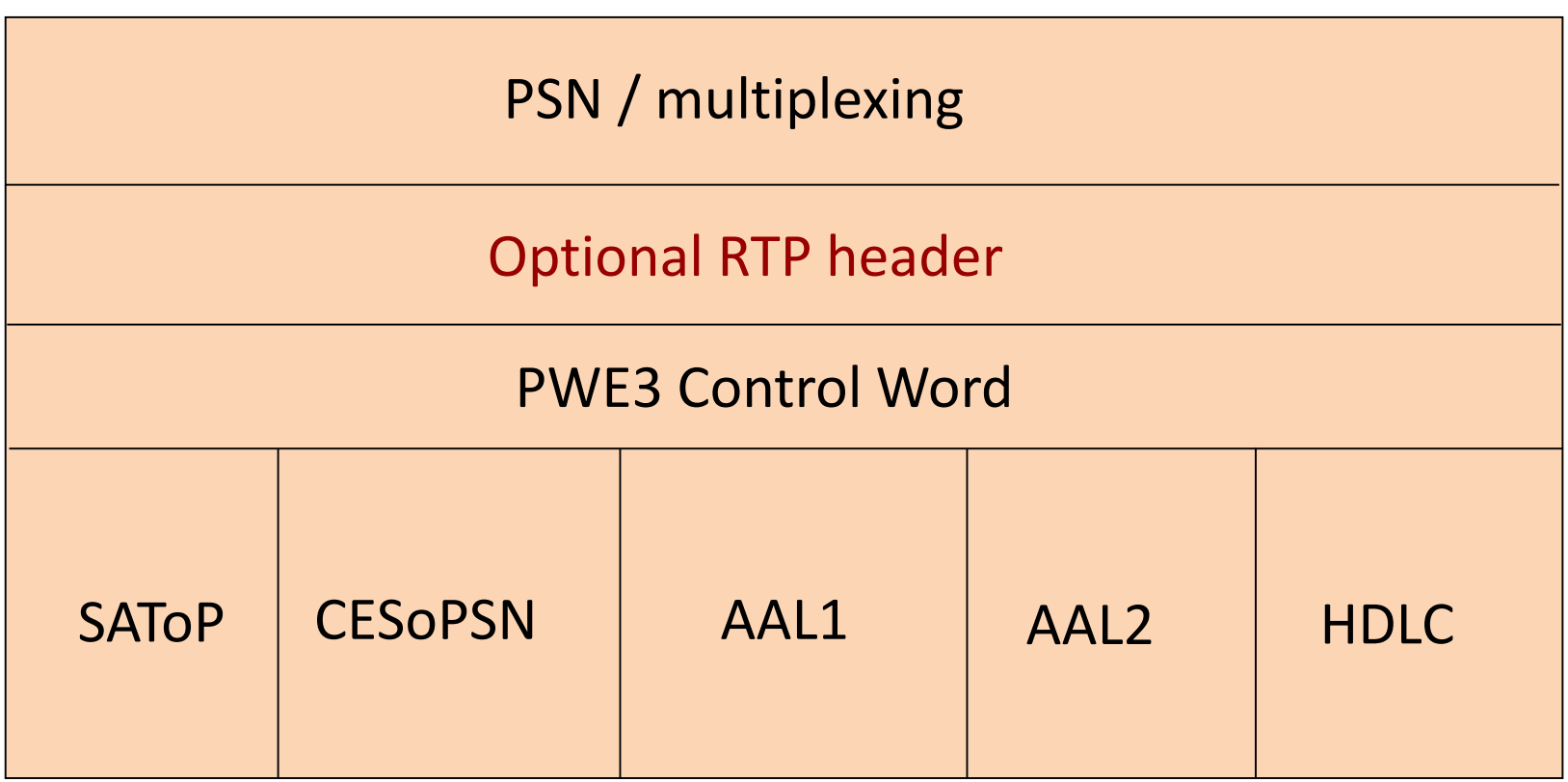
Structure-indicated encapsulation (TDMoIP – AAL1 mode)



Structure-reassembled encapsulation (TDMoIP – AAL2 mode)



TDM PW layering structure

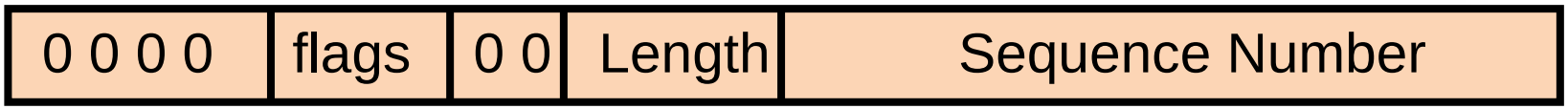


AAL1/CESoPSN used for preconfigured setup

AAL2 used for *dynamic bandwidth*

HDLC used for CCS signaling

TDMoIP Control Word



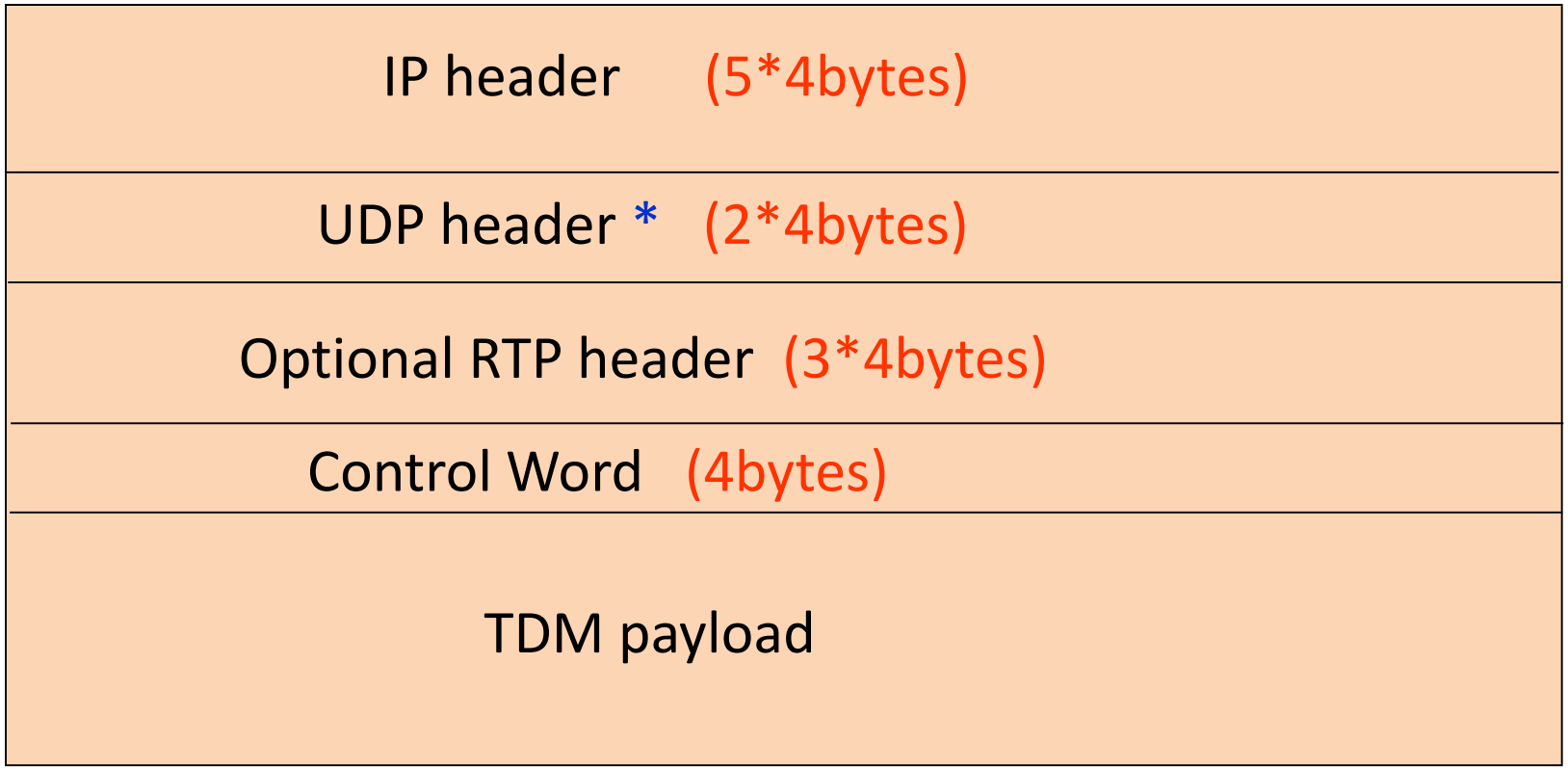
0 0 0 0
- ensures differentiation between IP and MPLS PSNs

Flags (4 b)
- L bit (Local failure)
- R bit (Remote failure)
- M field (2 b)

Length (6 b)
- used when packet may be padded by lower layer

Sequence Number (16 b)
- runs from 0 to max and wraps
- used to detect packet loss / misordering

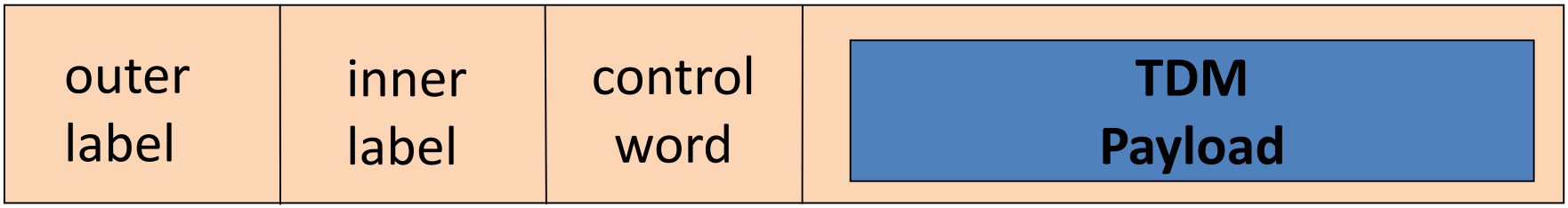
TDMo**IP** packet format



* The UDP source/destination port number is used as a PW label

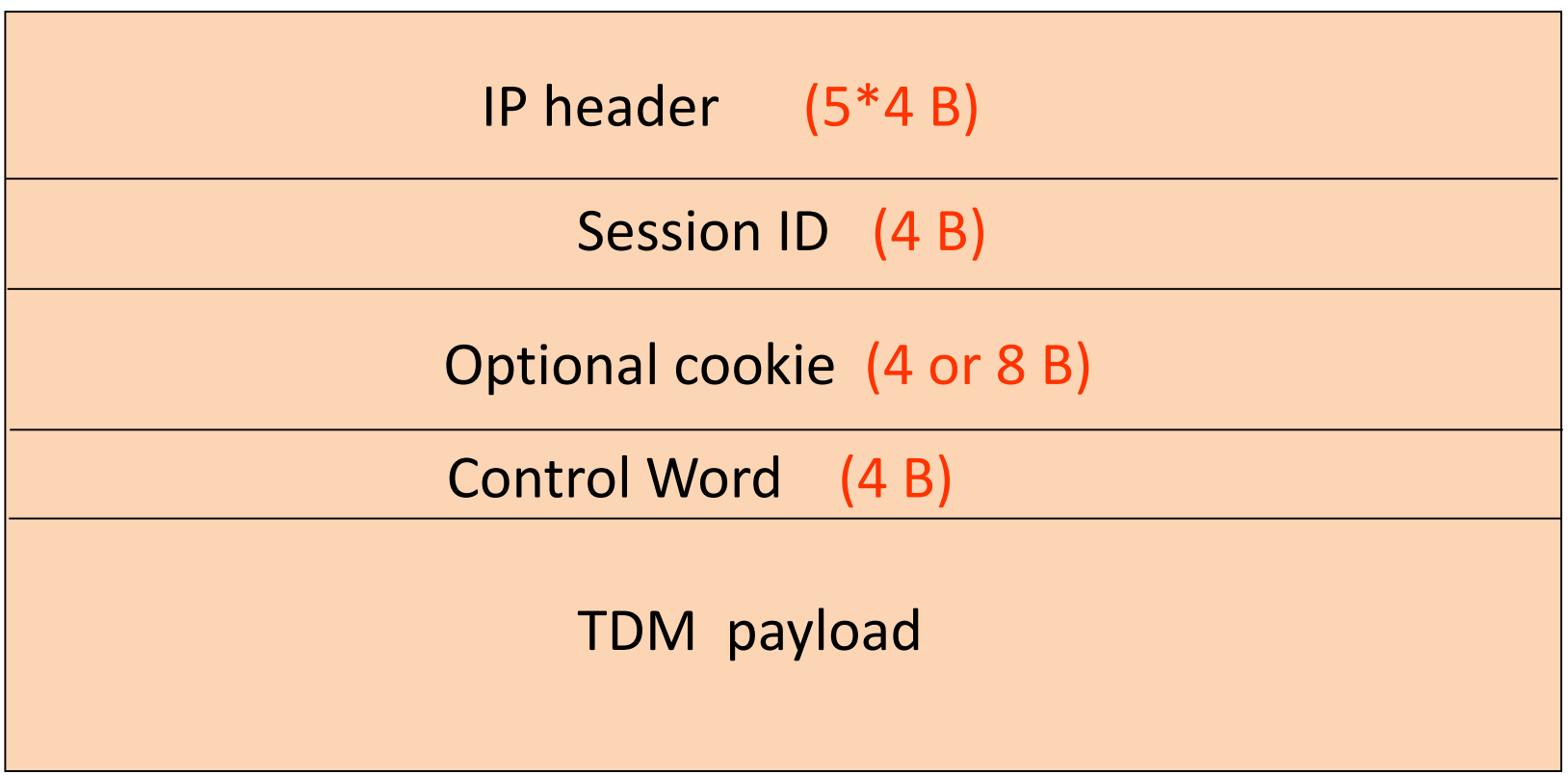
TDMoIP registered port number `0x085E (2142)`

TDMoMPLS packet format



- Inner and outer labels specify TDM routing and multiplexing
  - Inner Label contains TDMoMPLS circuit bundle number
- The control word
  - enables detection of out-of-order and lost packets
  - indicates critical alarm conditions
- The TDM payload may be *adapted*
  - to assist in timing recovery and recovery from packet loss
  - to ensure proper transfer of TDM signaling
  - to provide an efficiency vs. latency trade-off

TDM o L2TPv3



Note : No UDP header

TDM PW Protocol Processing

Steps in TDMoIP

- The synchronous bit stream is segmented
- The TDM segments may be *adapted*
- TDM PW control word is prepended
- PSN (IP/MPLS) headers are prepended (encapsulation)
- Packets are transported over PSN to destination
- PSN headers are utilized and stripped
- Control word is checked, utilized and stripped
- TDM stream is reconstituted (using adaptation) and played out

Optional explicit timing

VoIP uses RTP (Real-Time Protocol)

RTP can be used to transport timing across IP networks

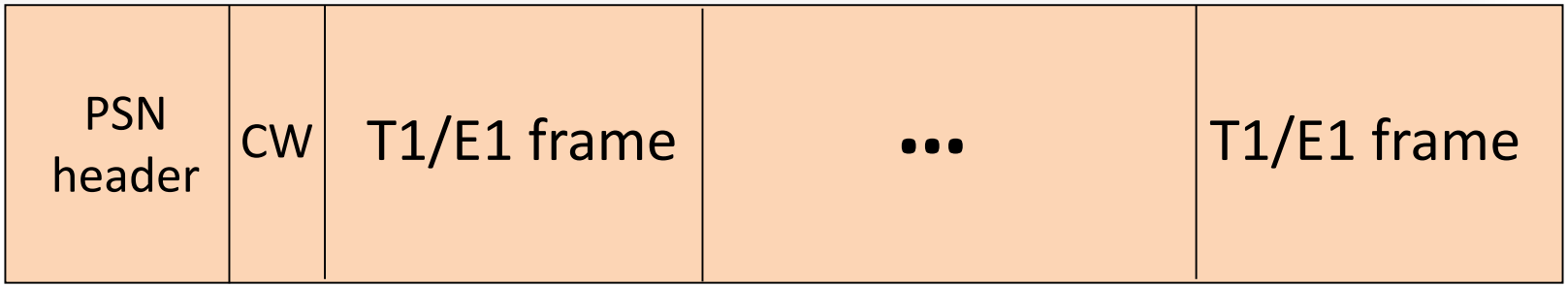
It does this by providing:
- a 16 bit sequence number
- a 32 bit timestamp

at the expense of 12 additional overhead bytes per packet

Accurate timing is important in telephony
and IP networks add packet delay variation (PDV)

For TDM PWs, only the timestamp is needed (SN is in CW)
- encodes time of sending using clock N*8kHz
- Note
  - this is NOT the normal RTP clock (number of samples)
  - rather with respect to a *common clock*

CESoPSN mode



Can efficiently handle fractional T1/E1

FRG field in CW enables support of multiframe

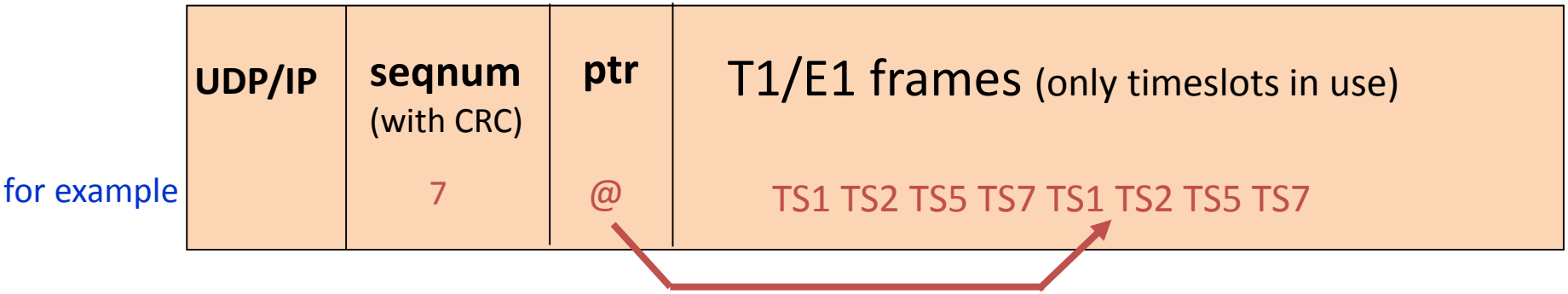
- CAS signaling uses a superframe (16/24 frames)
- Superframe/multiframe integrity must be respected

Octet aligned mode for T1 (24 bytes plus one bit per frame)

TDMoIP AAL1 mode

Packet loss, misorder, PDV problems can be solved by:

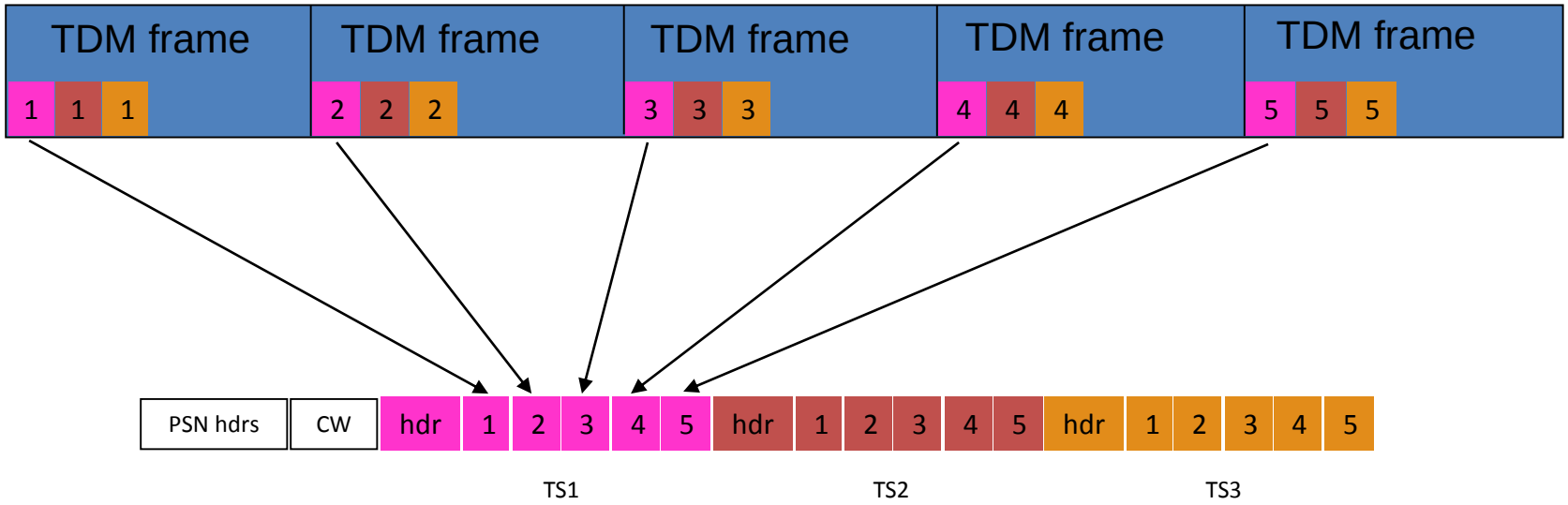
- adding a packet sequence number
- adding a pointer to the next multi-frame boundary
- only sending timeslots in use
- allowing multiple frames per packet



Good idea! This is AAL1 !

using precisely AAL1 enabled service interworking with ATM networks

TDMoIP AAL2 mode



- Each minicell consists of a header and buffered data
- Minicell header contains:
  - CID (Channel IDentifier)
  - LI (Length Indicator) = length-1
  - UUI (User-User Indication) counter + payload type ID

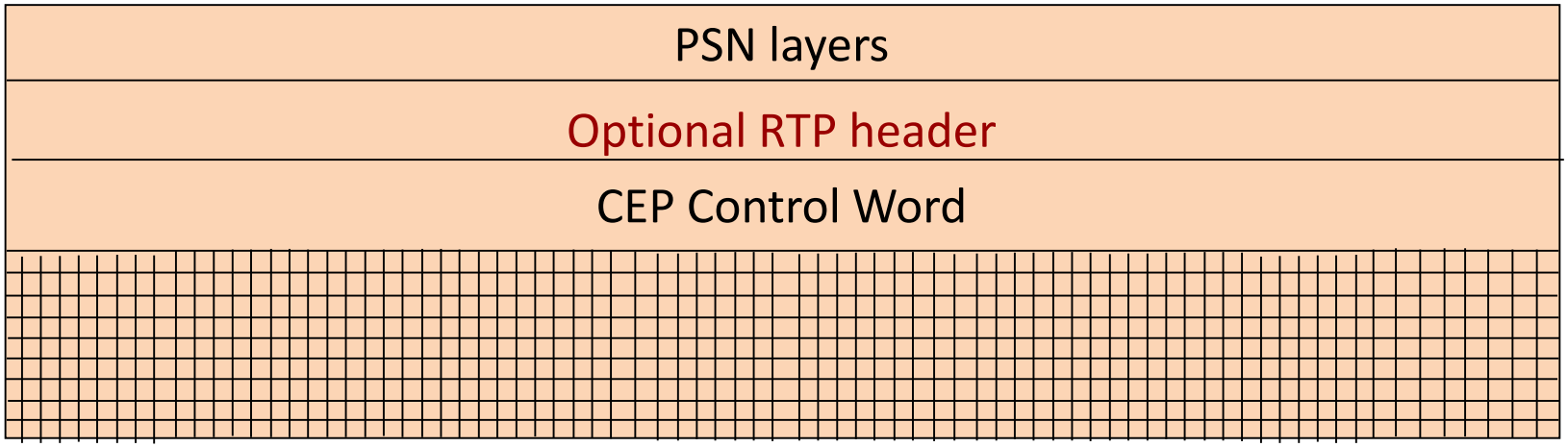
using precisely AAL2 enabled service interworking with ATM networks

CEP

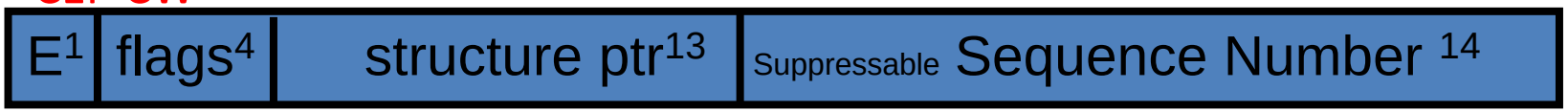
Circuit **E**mulation over **P**acket is an SONET/SDH PW

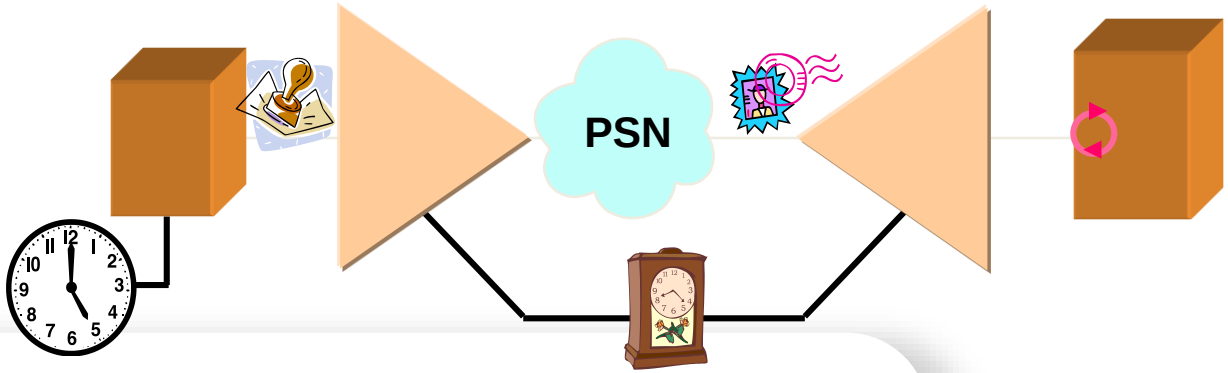
Encapsulate **S**ynchronous **P**ayload **E**nvelope fragment

Structure Pointer in CW points to J1 byte in STM frame



CEP CW





TDM PW timing recovery

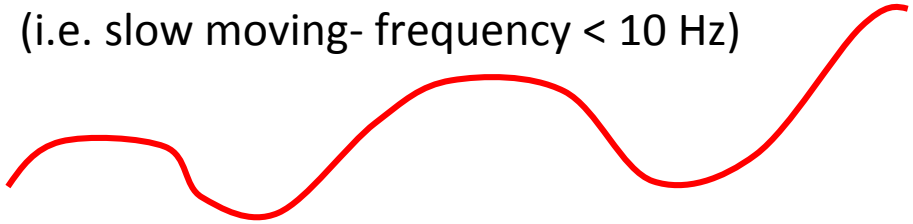
TDM Jitter and Wander

Jitter = short term timing variation *
(i.e. fast jumps - frequency > 10 Hz)

Jitter amplitude in UI_{pp}
Unit Interval pk-pk

E1 : 1 UI_{pp} = 1/2MHz = 488 ns

Wander = long term timing variation *
(i.e. slow moving- frequency < 10 Hz)



Measure in MTIE(τ) or TDEV(τ)

MTIE - max pk-pk error

TDEV expected deviation

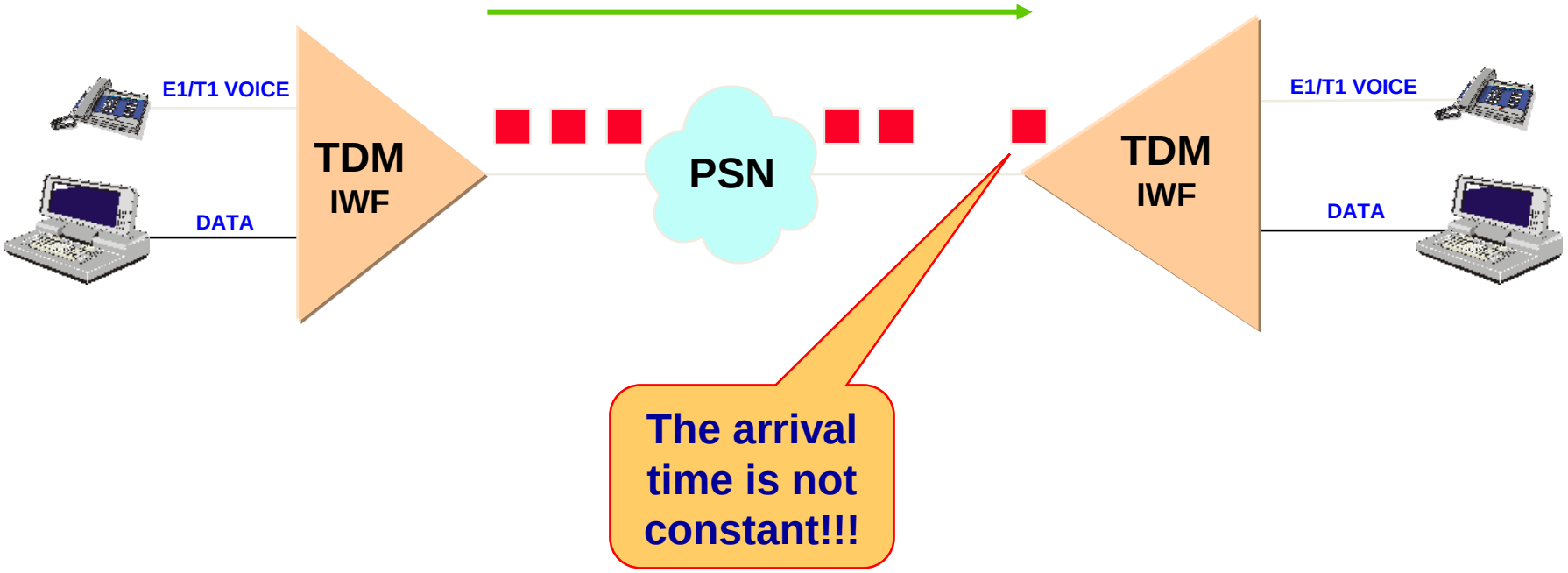
Mask as function of τ

* compared to reference clock

Note: requirements for E1 given in G.823
for T1 given in G.824

PSN - Delay and PDV

- PSNs do not inherently distribute timing
  - *clock recovery* required for TDM PWs
- PSNs introduce *delay* and *packet delay variation* (PDV)
  - Delay degrades perceived voice quality
  - PDV makes clock recovery difficult



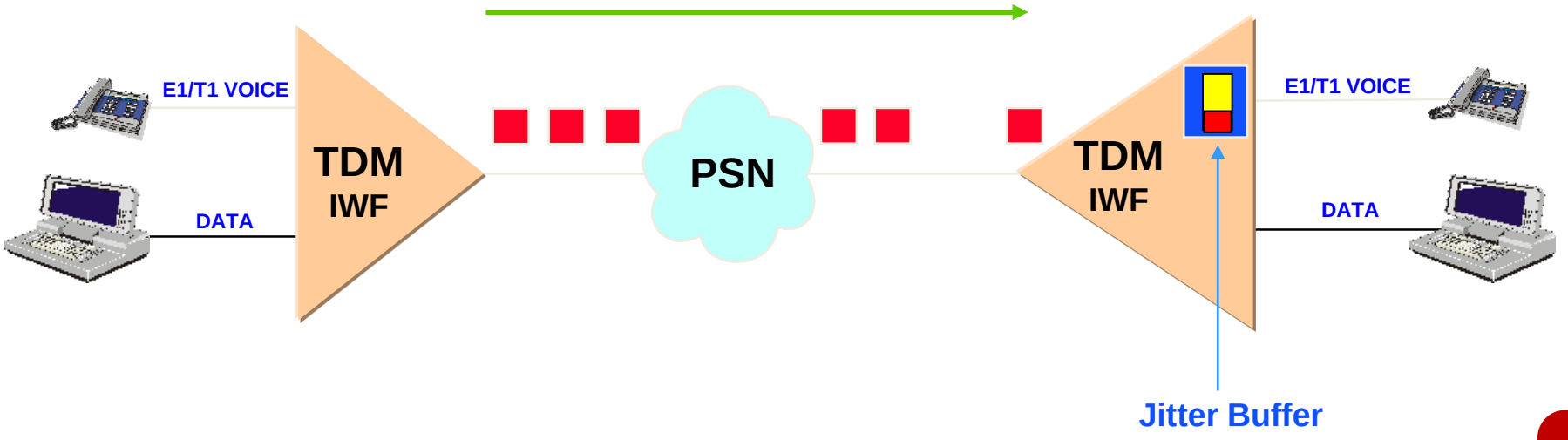
Jitter Buffer

Arriving TDM data is written into *jitter buffer*

Once buffer filled 1/2 can start reading from buffer

TDM data is read from jitter buffer at constant rate

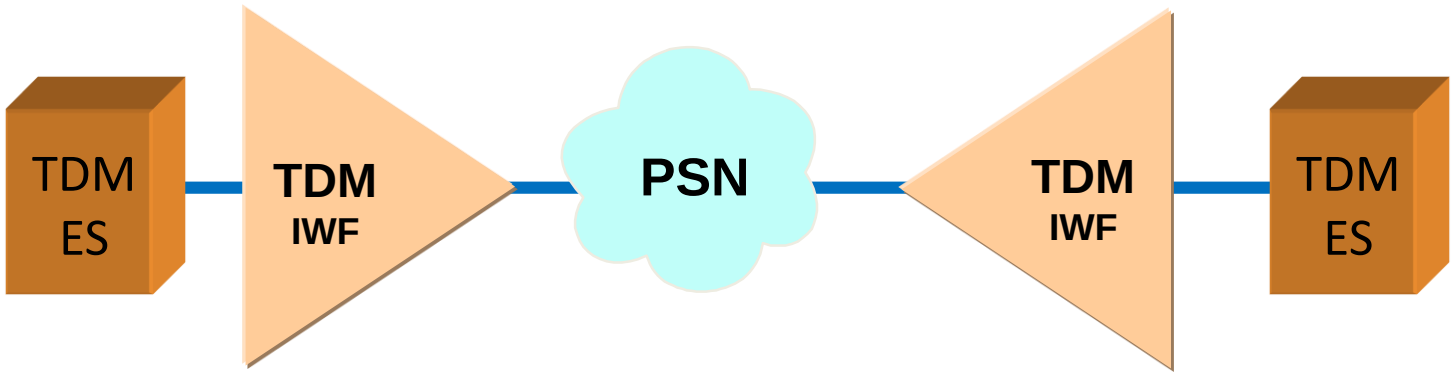
- How do we know the right rate?
- How do we guard against buffer overflow/underflow?



Timing scenarios

We can define 4 timing scenarios (see Y.1413)

- reference clock at TDM end systems
- reference clock at IWFs
- common clock at IWFs
- adaptive clock at one IWF



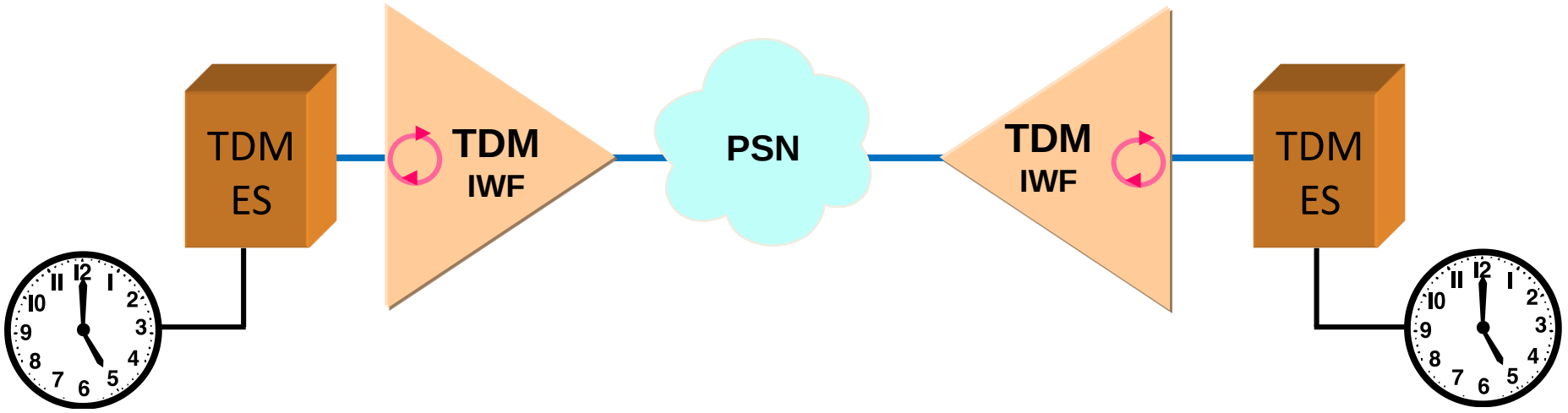
Reference at TDM ESs

prevalent when an isolated TDM link is replaced by a TDM PW

references are traceable to the same (or to *any*) G.811 PRC

the IWFs lock onto the TDM clock from the TDM end systems

performance will conform to standards



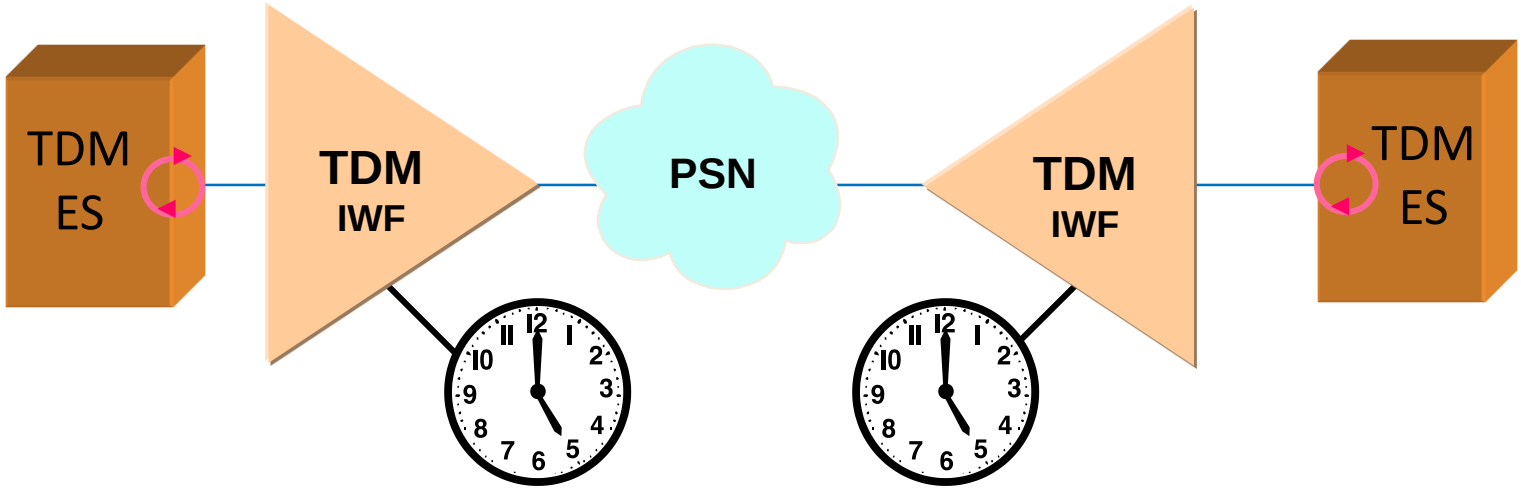
Reference at IWFs

prevalent when interconnecting isolated TDM equipment (e.g. PBXs)

references are traceable to the same (or to *any*) G.811 PRC

TDM end systems lock onto the TDM clock from IWFs (loop-back timing)

performance will conform to standards



Common clock

common clock is unrelated to TDM source clock

may be distributed over SyncE, by neighboring TDM links, or GPS

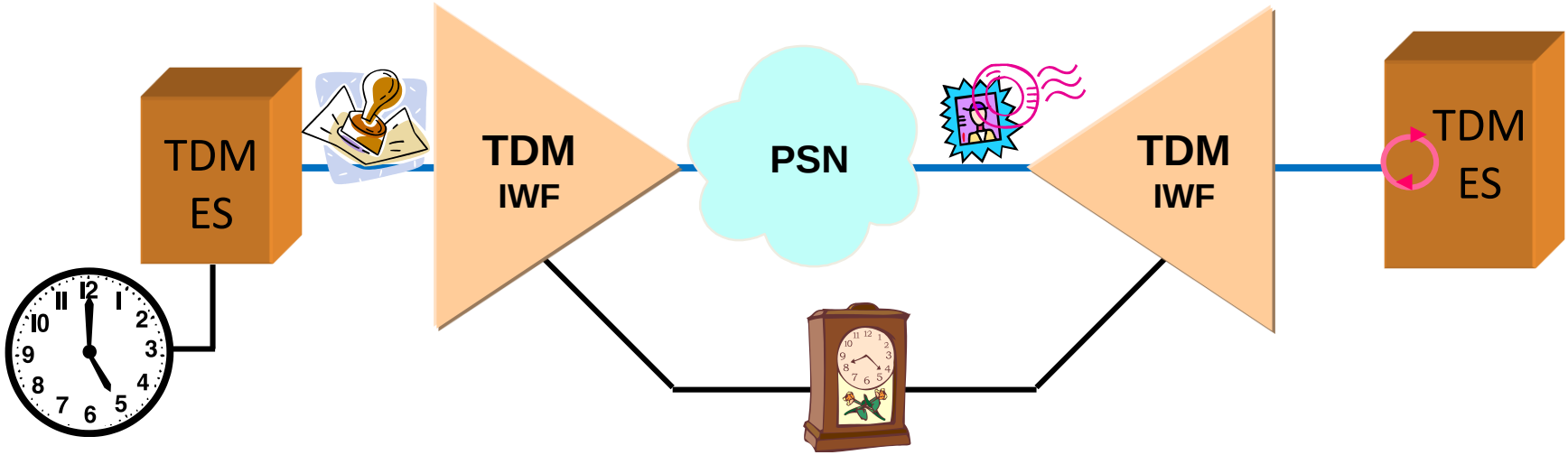
needn't be G.811 PRC, as long as both IWFs see the same clock

ingress IWF timestamps each outgoing packet using common clock

egress IWF compensates each incoming packet based on timestamp

common clock makes PSN's PDV irrelevant

performance will conform to standards



Timestamps w/o common clock

some standards *mandate* a timestamp in ***absolute mode***

i.e. a timestamp based on the TDM source clock

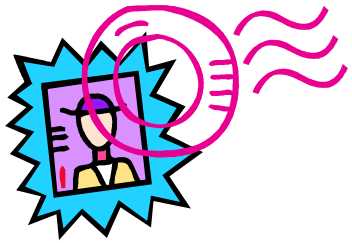
these standards call common-clock timestamps ***differential mode***

such a timestamp is ***redundant***

(linearly dependent on the packet sequence number)

and therefore completely useless for a clock recovery !

without a common clock there is *no reason* to timestamp packets



(Pure) adaptive clock

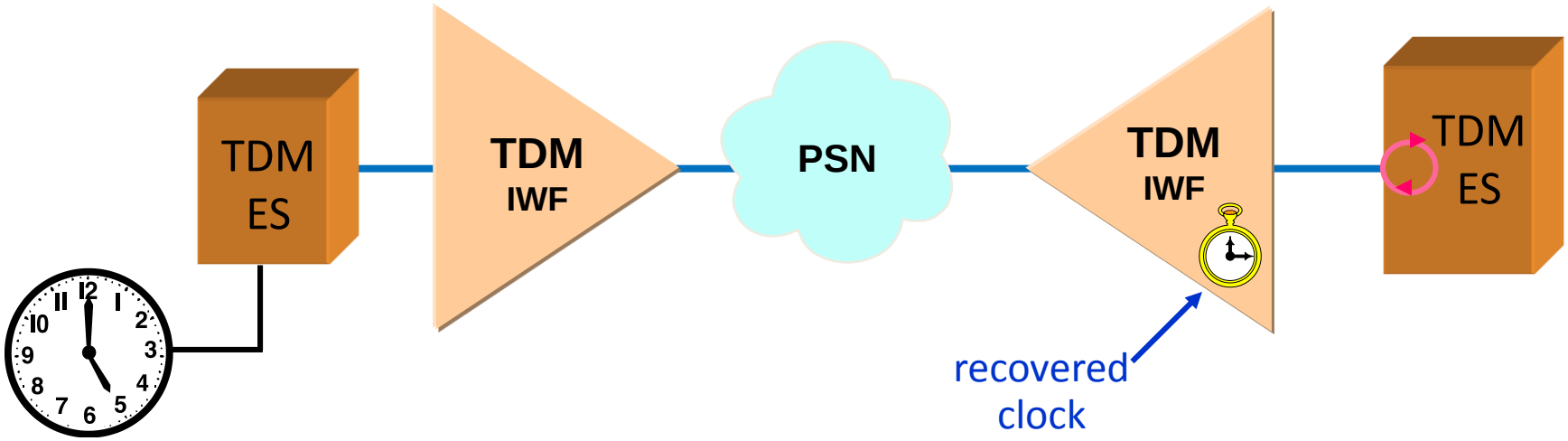
prevalent when there is a central network and remote sites

remote site has no access to TDM clock and there is no common clock

remote IWF recovers clock based solely on incoming packets

remote TDM end system uses loop-back timing

performance depends strongly on PSN characteristics



Adaptive Clock Recovery

The packets are injected into network ingress at times T_n

For TDM the source packet rate R is constant

$$T_n = n / R$$

The network delay D_n can be considered to be the sum of

typical delay d and random delay variation V_n

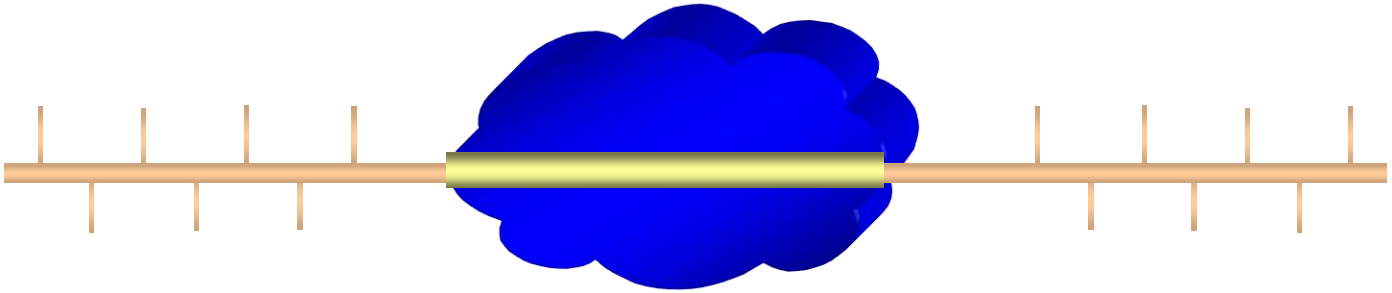
The packets are received at network egress at times t_n

$$t_n = T_n + D_n = T_n + d + V_n$$

By proper averaging/filtering (control loops)

$$\langle t_n \rangle = T_n + d = n / R + d$$

and the packet rate R has been recovered



Ethernet PWs

Traditional WAN architecture

Ethernet layer is terminated

- only higher layer (e.g., IP) is transported
- the traffic is no longer Ethernet at all

Ethernet header

- removed at ingress, and
- new header added at egress

This is *not* transparent Ethernet LAN interconnect

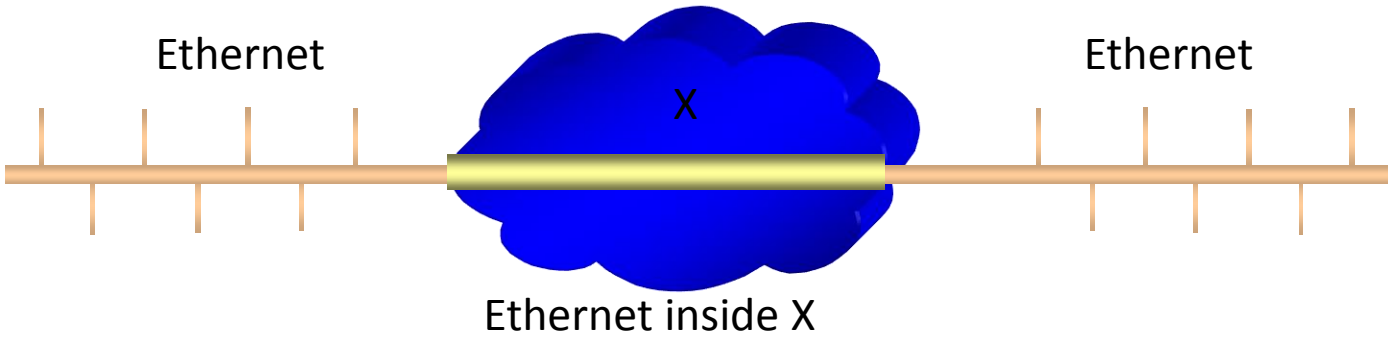
- Ethernet LANs with many higher layer packet types
  can't be interconnected
- raw L2 Ethernet frames can not be sent

Tunneling Ethernet frames

Users with multiple Ethernet sites may want to connect their LANs so that all locations appear to be on the same LAN

This requires *tunneling* of *all* Ethernet L2 frames (not only IP) between one LAN and another

The entire Ethernet frame needs to be preserved (except perhaps the FCS, which may be regenerated at egress)



Ethernet PW (RFC 4448)

While PWs were designed for legacy traffic

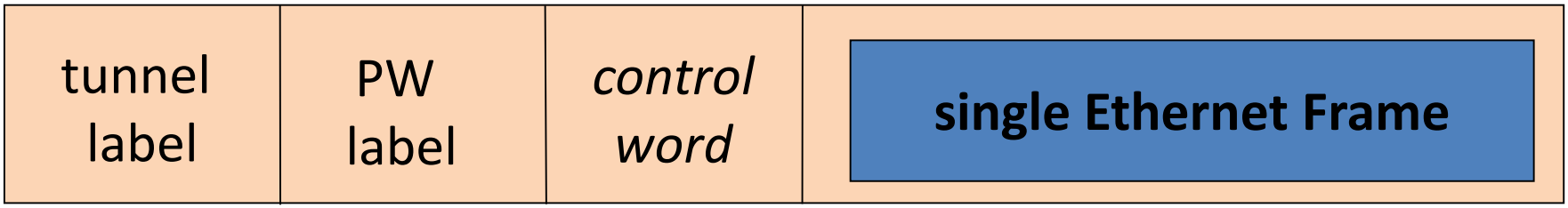
Ethernet PWs are now the most popular type

This is because native Ethernet transport is limited in range

RFC 4447 specifies:

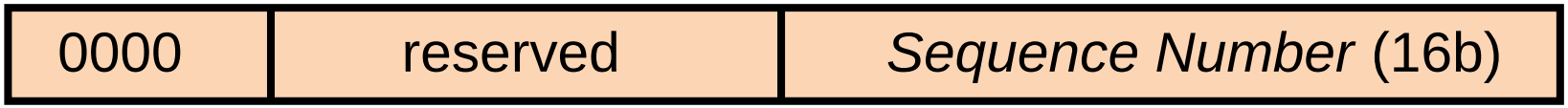
- Control word is optional
  even if control word is used, sequence number if optional
- Standard mode – FCS is stripped and regenerated
  FCS retention mode (RFC 4720) allows retaining FCS
- Can transport tagged or untagged Ethernet frames
  if tagged :
  encapsulation can be *raw* mode or *tagged* mode
  tagged mode processes (inserts/swaps/removes) service delimiting tags

Ethernet Pseudowire packet

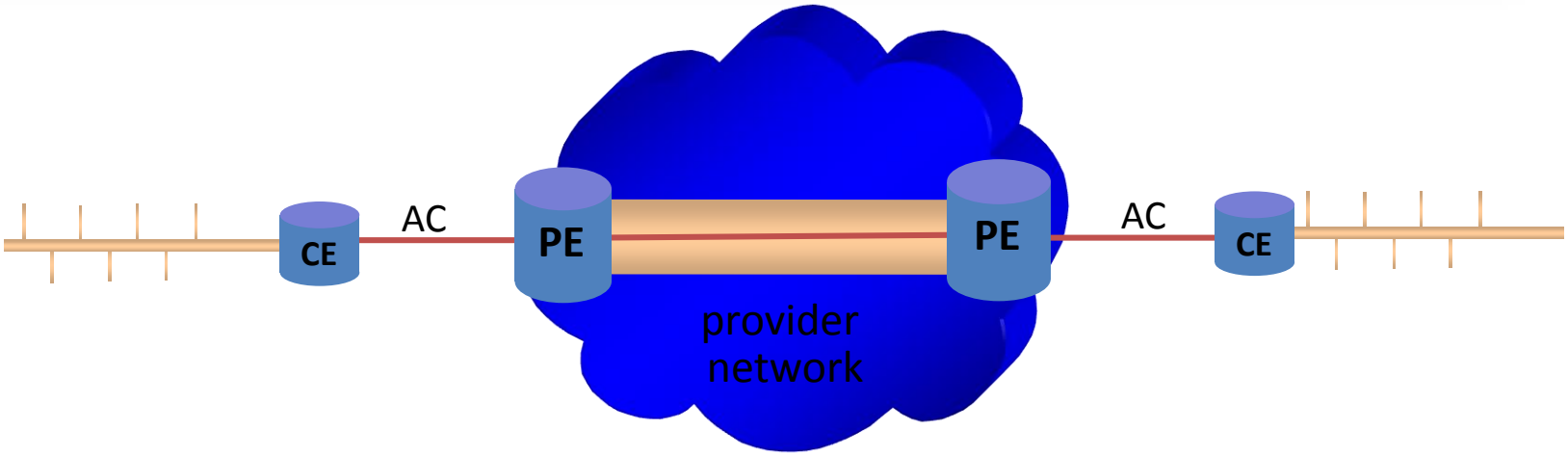


Ethernet Frame usually has FCS stripped
SP tag may also be stripped

optional control word
generation and processing of sequence number is optional



VPWS



Based on Ethernet PWs one can provide a

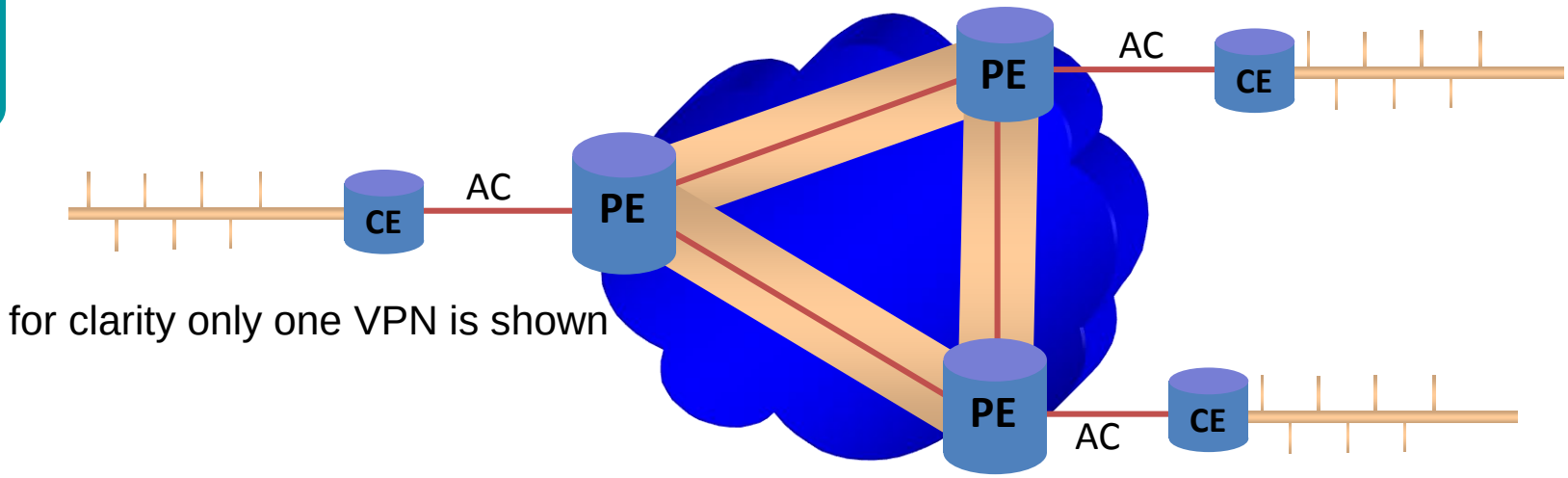
Virtual **P**rivate **W**ire **S**ervice or a

Virtual **P**rivate **L**AN **S**ervice (L2VPN)

VPWS emulates a *wire* supporting the Ethernet physical layer

- set up MPLS tunnel between PEs
- set up Ethernet PW inside tunnel
  CEs appear to be connected by a single L2 circuit
  (can also make VPWS for ATM, FR, etc.)

VPLS

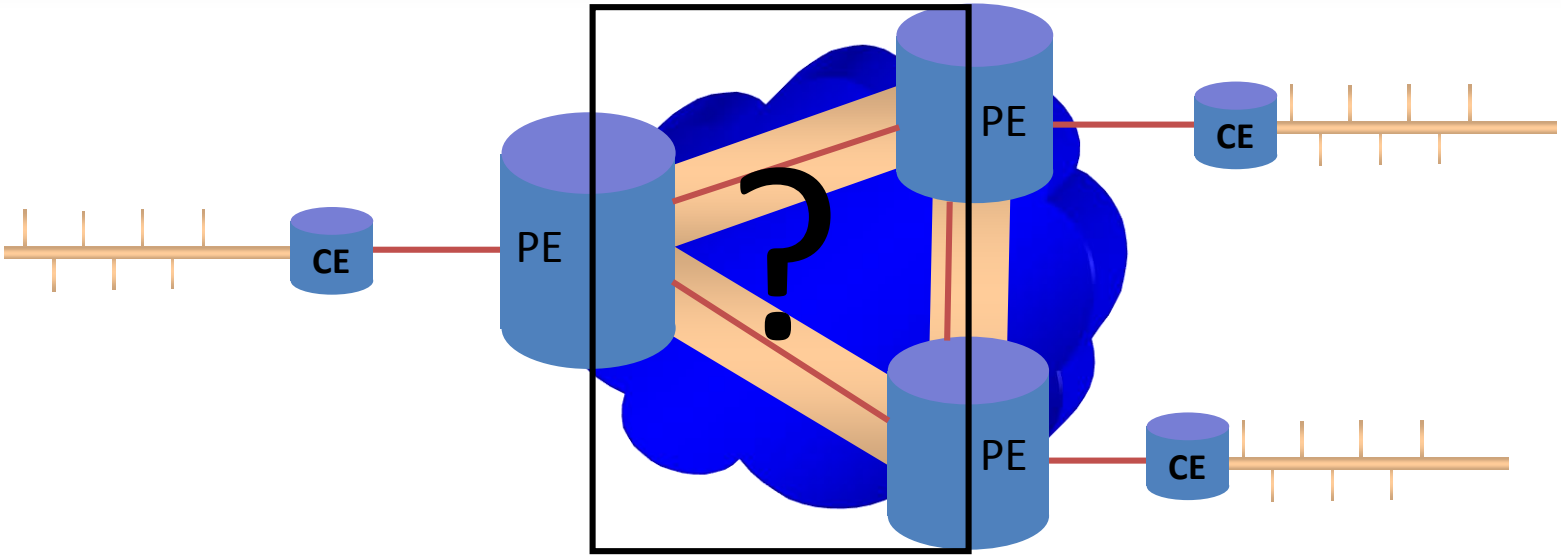


VPLS emulates a LAN over an MPLS network

- set up MPLS tunnel between every pair of PEs (full mesh)
- set up Ethernet PW inside tunnels
- for each VPN instance
  CEs appear to be connected by a single LAN

What is the difference between this and a L3 VPN service ?

L2VPN vs. L3VPN

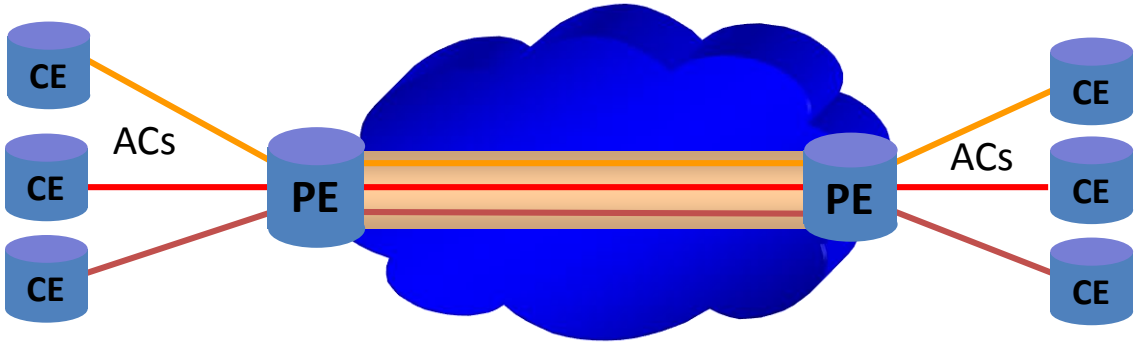


in L**2**VPN CEs appear to be connected by single L2 network

- PEs are transparent to L3 routing protocols
- CEs are routing peers

in L**3**VPN CE routers appear to be connected by a single L3 network

- CE is routing peer of PE, not remote CE
- PE maintains routing table for each VPN



Other PW types

What else is there ?

For what native service types have PW encaps been defined ?

- TDM (SONET/SDH, E1, T1, E3, T3)
- Ethernet
- ATM (port mode, cell mode, AAL5-specific modes)
- Frame Relay
- HDLC / PPP
- IP PW (in 4447, never became a document)
- Fiber channel
- Generic *packet PW*

ATM PWs (RFC 4717)

- N:1 (Martini mode)
  - control word optional
  - remove HEC to form 52-byte cells
  - pack 1 or more cells into MPLS packet (*may* mix VPI/VCI)
- 1:1 (ATM forum mode)
  - special 3-byte control word required
  - 1-byte header for each cell
  - pack 1 or more cells into MPLS packet
  - *may* mix VCI for given VPI (but then need to insert VCI)
- SDU (for AAL5 only)
  - control word with 4 flags required (SN optional)
  - payload is the complete SDU (no trailer)
- PDU (for AAL5 only)
  - special 3-byte control word required
  - 1-byte header after CW
  - payload is N*48-byte PDU
- port mode (RFC 4816) - format same as N:1

Frame Relay PWs (RFC 4619)

2 encapsulations :

- port mode (many-to-one mapping) 1 PW per all FR VCs identical to HDLC PW encapsulation RFC4618
- 1:1 mode – one PW for each FR VC

Mandatory (for 1:1) CW contains 4 flags

- **F** FECN (Forward Explicit Congestion Notification)
- **B** BECN (Backward Explicit Congestion Notification)
- **D** DE bit (Discard Eligibility)
- **C** C/R (Command/Response)

SN optional and **FRG** extensively used

Packet PW (RFC 6658)

Generic packet service to carry
any packets exchanged between adjacent LSRs, such as

- IP packets (user, system)
- Ethernet packets (e.g., IS-IS, LLDP, Ethernet OAM)

To mux different packet types we use raw Ethernet PW
although there is no real Ethernet interface

MAY use local MAC addresses
or MAY use (IANA allocated) MAC addresses PacketPWEthA / PacketPWEthB

Fiber Channel PW (RFC 6307)

FC is a high-speed communications link used for SANs

Fibre Channel Over TCP/IP (FCIP) (RFC3821) needs TCP to retransmit dropped frames and ensure order

MPLS networks may have very low loss and misorder rates

FC PW transparently transports FC traffic over MPLS with simpler processing than FCIP

However, FC PWs have more complex NSP than other PWs