



data communications

www.rad.com

Advances in Ethernet



Unique Access Solutions

Yaakov (J) Stein
Chief Scientist
RAD Data Communications

June 2010

Outline

Modern Ethernet
VLANs and their uses
Ethernet services
Additional bridging functions
QoS Aspects
Link aggregation
Ethernet protection mechanisms
EFM
RPR
Ethernet OAM
Ethernet security
Synchronous Ethernet

Modern Ethernet

Carrier grade Ethernet

IEEE 802 view

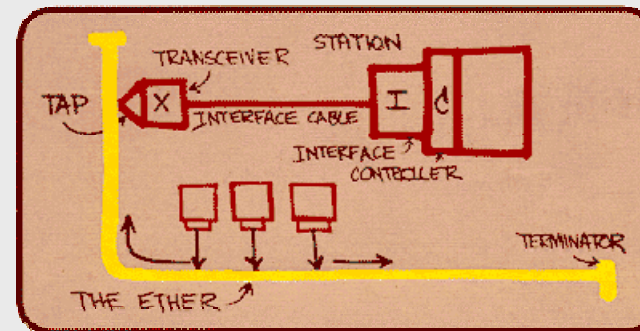
ITU-T view

MEF view

IETF view

What is Ethernet anyway?

Ethernet has evolved far from its roots of half-duplex/CSMA/CD LANs and is hard to pin down today



Metcalfe's original sketch of Ethernet

we may use the term today to describe

- full duplex 10G point-to-point optical links
- “Ethernet in the first mile” DSL access
- passive optical “GEPON” networks
- metro Ethernet networks
- “wireless Ethernet” 10M hot spots
- etc.

“Carrier grade” Ethernet

Ethernet started out as a *LAN* technology

LAN networks are relatively small and operated by consumer
hence there are usually no management problems

as Ethernet technologies advances out of the LAN environment

new mechanisms are needed, e.g.

- OAM
- deterministic (Connection-Oriented) connections
- synchronization

the situation is further complicated by different “world views”
of various SDOs working on Ethernet standardization

4 views

IEEE 802 LAN/MAN standards committee (since 1980)

Ethernet is a set of *LAN/MAN standards*



ITU-T (since 1865 / 1956)

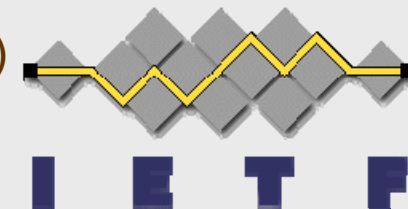
Ethernet is several *packet-based layer networks*

Metro Ethernet Forum (since 2001)



Ethernet is a *service* provided to a customer

Internet Engineering Task Force (since 1986)



Ethernet is an *IP-helper*

IEEE 802, misc WGs, documents

802 LAN/MAN Standards Committee

- 802-2001
- 802.1 LAN protocols WG
 - **802.1D-2004**
 - **802.1Q-2005**
 - 802.1ad
 - 802.1ah
- 802.2 LLC
- 802.3 **Ethernet** WG
 - **802.3-2005**
 - 802.3z GbE
 - 802.3ad link aggregation
 - 802.3ah EFM
 - 802.3as 2000 byte frames
- 802.11 Wireless LAN WG (WiFi)
 - **802.11-2005**
 - 802.11a
 - 802.11b
 - 802.11g
- 802.16 Broadband Wireless Access WG (WiMax)
- 802.17 RPR WG

Note:

working groups and study groups
(e.g 802.1, 802.3) are semi-permanent

projects and task forces

(e.g. 802.3z, EFM) are temporary

project outputs are usually

absorbed into main WG document

802.3

actually, IEEE only calls 802.3 Ethernet

802.3 is a **large** standard, defining

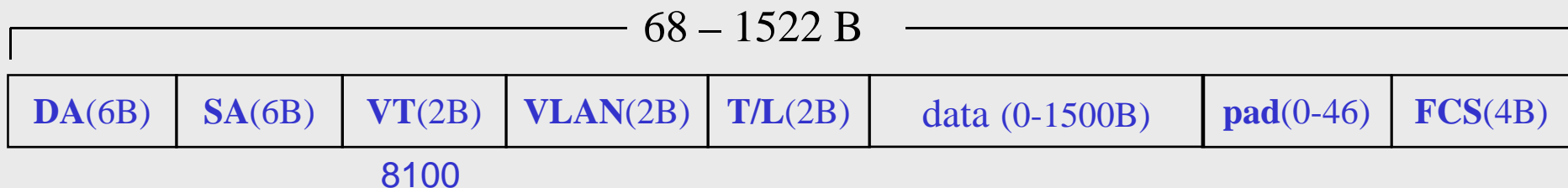
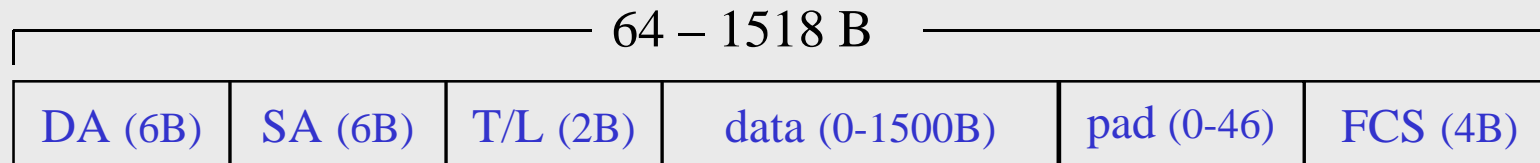
- MAC frame format, including VLAN support
- medium specifications and attachment units (UTP, coax, fiber, PON)
- repeaters
- interfaces (e.g. MII, GMII)
- rate autonegotiation
- link aggregation (we will discuss later)

new projects continue to expand scope

- 802.3aq 10GBASE-LRM
- 802.3ar congestion management
- 802.3as frame expansion

MAC frame format

a *MAC frame* uses either of the following frame formats :



802.3as expanded frame size to 2000B (approved September 2006)

Note: PHY frame may be larger – e.g. preamble, start-frame delimitator, etc.

Ethernet Addressing

the most important part of any protocol's overhead are the *address fields*

Ethernet has both source (SA) and destination (DA) fields

the addresses need to be unique to the network

the fields are 6-bytes in length in EUI-48 format

(once called MAC-48, EUI = **E**xtended **U**nique **I**dentifier)

$2^{48} = 281,474,976,710,656$ possible addresses

addresses can be “universally administered” (burned in)
or “locally administered” (SW assigned)

EUI-48 and EUI-64

EUI-48 used by

- Ethernet (802.3)
- Token ring (802.5)
- WiFi (802.11)
- Bluetooth
- FDDI
- SCSI/fiber-channel

IEEE defined a “next generation” 8-byte address called EUI-64

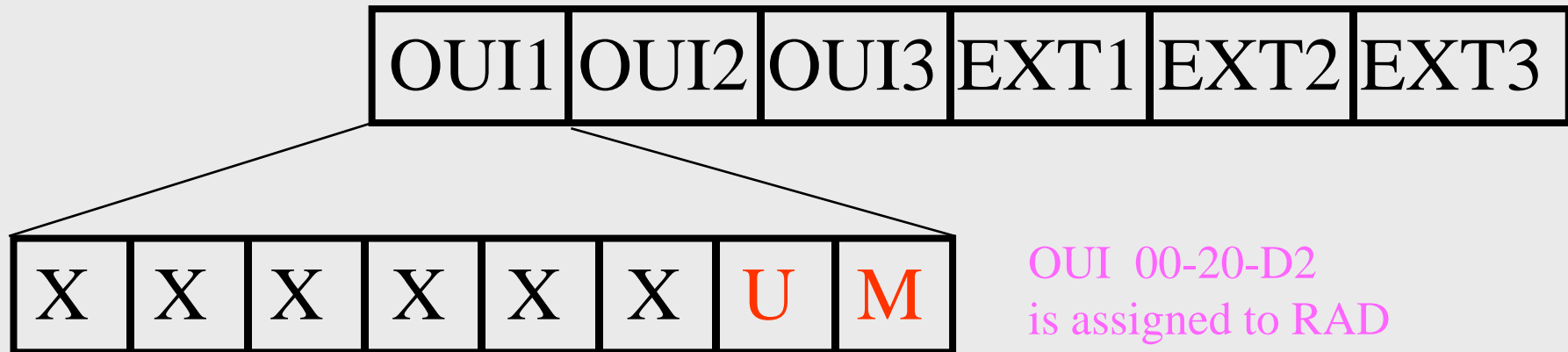
EUI-64 used for

- IEEE 1394 (firewire)
- 802.15.4 (personal area networks)
- IPv6 (LSBs of non-temporary unicast address)

EUI addresses usually expressed in hex-hex format

Broadcast address is FF-FF-FF-FF-FF-FF

EUI format



OUI (ex “company name”) is assigned by the IEEE Registration Authority
 each OUI gives 16M addresses (IEEE expects not to run out before 2100)

the LSB of the OUI is the **M**ulticast indicator (0=unicast, 1=multicast)

the next to LSB is the **U**niversal / local bit

0 means UNIVERSALLY allocated address (all assigned OUIs have zero)

1 means there is no OUI - use any unique address

WARNING – bit is reversed in IPv6!

OUIs are also used by *LLC SNAP* and in *slow protocols*



Ethernet clients

the 2-byte *Ethertype* identifies the client type
assigned by IEEE Registration Authority
all Ethertypes are greater than 0600 (1536 decimal)

some useful Ethertypes :

- 0800 IPv4
- 0806 ARP
- 22F3 TRILL
- 22F4 IS-IS
- 8100 VLAN tag
- 8138 Novell IPX
- 814C SNMP over Ethernet
- 86DD IPv6
- 8809 slow protocols
- 8847 MPLS unicast
- 8848 MPLS multicast
- 88D8 CESoETH
- 88A8 Q-in-Q SVID / MAC-in-MAC BVID
- 88F5 MVRP
- 88F6 MMRP
- 88F7 IEEE 1588v2
- 8902 CFM OAM

see them all at <http://standards.ieee.org/regauth/ethertype/eth.txt>
get your own for only \$2,500 ! (RAD has acquired 22E8 for bonding protocol)

Slow protocol frames

slow protocols are slow – no more than 5 (or 10) frames per second

no more than 100 frames per link or ONU

slow protocol frames must be untagged, and must be padded if needed

slow protocols are for single links – they do not traverse bridges

there is a specific multicast address for multi-cast slow protocols

there can not be more than 10 slow protocols

01-80-C2-00-00-02



802-3 Annex 43B

Subtype:

- 1 is Link Aggregation Control Protocol (LACP)
- 2 is link aggregation marker protocol
- 3 is EFM OAM

LLC

There are other ways to differentiate clients (other than by Ethertype)

- 802.2 (**L**ogical **L**ink **C**ontrol)

first three bytes of payload :

- **D**estination **S**ervice **A**ccess **P**oint (1B)
- **S**ource **S**ervice **A**ccess **P**oint (1B)
- Control Field (1 or 2 B)



Example SAPs

04	IBM SNA
06	IP
80	3Com
AA	SNAP
BC	Banyan
E0	Novel IPX/SPX
F4	FE CLNS

SNAP



- **S**ub-**N**etwork **A**ccess **P**rotocol

LLC parameters plus expanded capabilities

SNAP can support IPX/SPX, TCP/IP, AppleTalk Phase 2, etc.

the first eight bytes of payload :

- **D**estination **S**ervice **A**ccess **P**oint (1B) = 0xAA
 - **S**ource **S**ervice **A**ccess **P**oint (1B) = 0xAA
 - Control Field (1B) = 0x03
 - OUI (3B)
 - Type (2B) (if OUI=00:00:00 then EtherType)
- IPX (old Netware method, “raw”) - first 2B of payload FF:FF
 - Note: standard DSAP/SSAP values can not be FF !
 - RFC 1042 allows IPv4 over Ethernet with SNAP
 - DSAP=AA, SSAP=AA, Control=3, SNAP=0 followed by EtherType

Parsing

if EtherType/Length > 1500 then EtherType

else if payload starts with FF-FF then Netware

else if payload starts with AA then SNAP

else LLC





L2 control protocols

The IEEE (and others) have defined various control protocols (L2CPs)

Here are a few well-known L2CPs :

protocol	DA	reference
STP/RSTP/MSTP	01-80-C2-00-00-00 802.2 LLC	802.1D §8,9 802.1D§17 802.1Q §13
PAUSE	01-80-C2-00-00-01	802.3 §31B 802.3x
LACP/LAMP	01-80-C2-00-00-02 EtherType 88-09 Subtype 01 and 02	802.3 §43 (ex 802.3ad)
Link OAM	01-80-C2-00-00-02 EtherType 88-09 Subtype 03	802.3 §57 (ex 802.3ah)
Port Authentication	01-80-C2-00-00-03	802.1X
E-LMI	01-80-C2-00-00-07	MEF-16
Provider MSTP	01-80-C2-00-00-08	802.1D § 802.1ad
Provider MMRP	01-80-C2-00-00-0D	802.1ak
LLDP	01-80-C2-00-00-0E EtherType 88-CC	802.1AB-2009
GARP (GMRP, GVRP)	Block 01-80-C2-00-00-20 through 01-80-C2-00-00-2F	802.1D §10, 11, 12

Note: we won't discuss autonegotiation as it is a *physical layer* protocol (uses link pulses)

Ethernet over coax

IEEE notation: Rate-Modulation-CableLimits

- Rate in Mb/s
- Modulation can be BASEband, BROADband, PASSband
- CableLimits e.g. distance in units of 100m

- 10BASE2

10 Mb/s thin coax (RG58) 185m CSMA/CD

- 10BASE5

10 Mb/s thick coax (RG11) 500m CSMA/CD



- 10BROAD36

10 Mb/s PSK CATV 2.8-3.6km CSMA/CD

Ethernet over twisted pairs

- 10BASE-T (T=Twisted pair)
10 Mb/s, Manchester, >100m, 2 pairs of UTP, CSMA/CD or FD
- 100BASE-TX
“fast Ethernet”, 100Mb/s, 4B5B encoding, 2 pair CAT5, FD
- 1000BASE-T
(ex 802.3ab, now 802.3 clause 40)
GbE, 4D-TCM-PAM5/EC, 100m, 4 pairs CAT5/5e/6, FD
- 10PASS-TS
(ex EFM, now 802.3 clause 62), 10Mb/s, 750m DMT VDSL
- 2BASE-TL
(ex EFM, now 802.3 clause 63), 2Mb/s, 2.7km, SHDSL



Ethernet over optical fiber

- 10BASE-FL
10 Mb/s, P2P, CSMA/CD / FD, 2km, backward-compatible with FOIRL
- 100BASE-FX
100 Mb/s, multimode fiber, 4B5B, 400m HD / 2km FD
- 1000BASE-LX
long λ (1270-1355 nm), 8B10B, >2km (single-mode), FD only
- 1000BASE-SX
short- λ (850nm near IR), 8B10B, 220m (multi-mode), FD only
- 10GBASE-SR/LR/ER/LX4
ex 802.3ae, short-range, long-range, extended range, WDM



802.1D

802.1 discusses MAC bridges

802.1D is also a **large** standard, defining

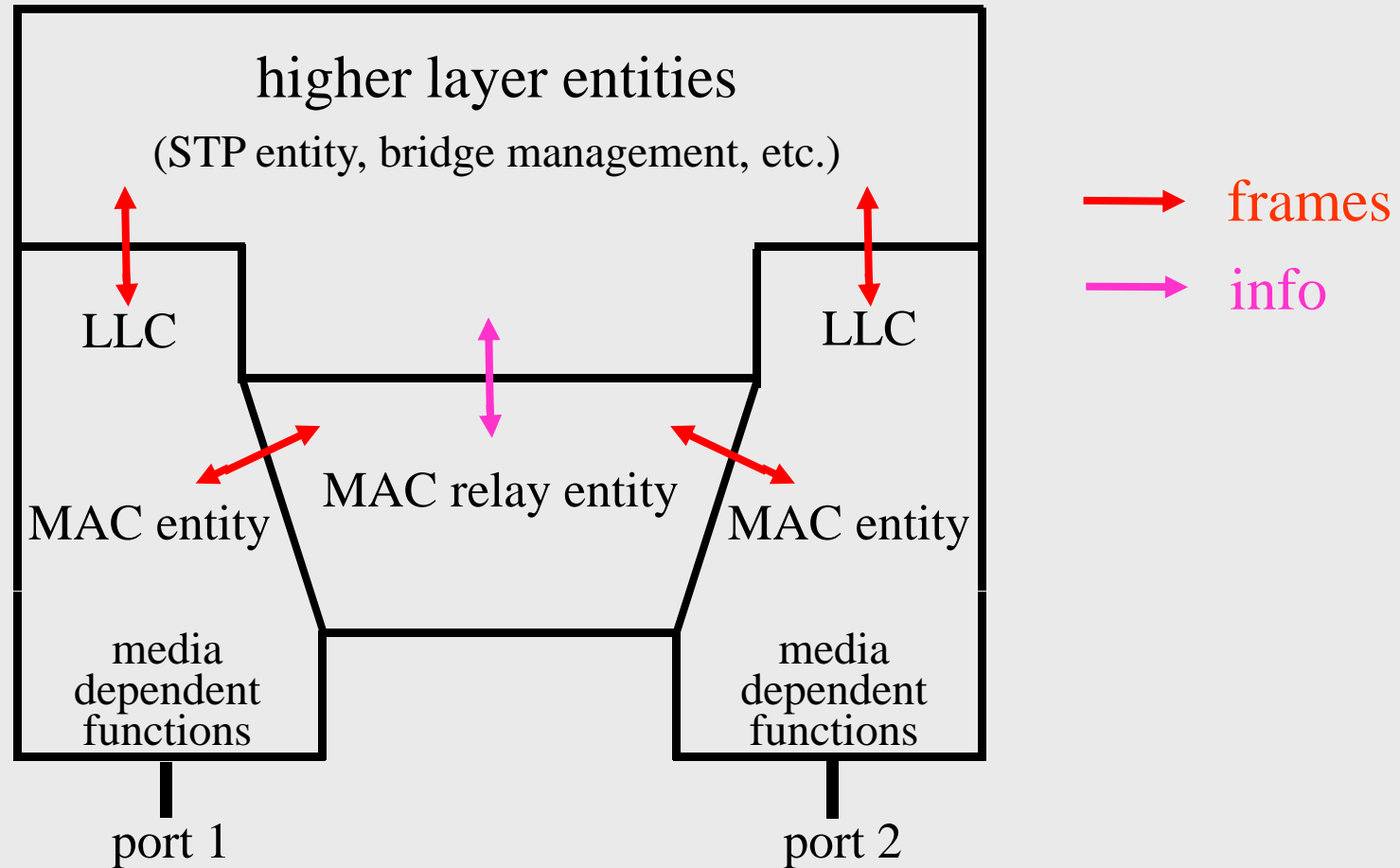
- bridge operation (learning, aging, STP, etc.)
- the architectural model of a bridge
- bridge Protocol and BPDUs
- GARP management protocols

802.1Q is a separate document on VLAN operation

new projects continue to expand scope

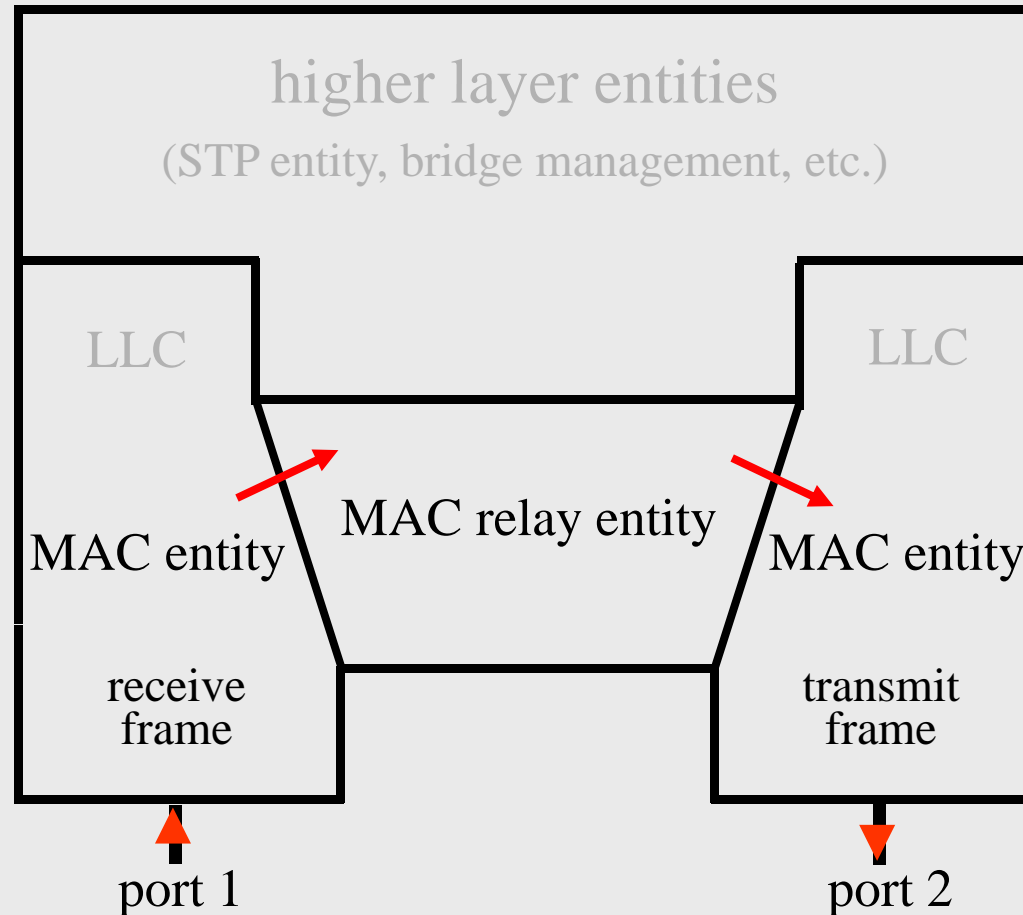
- 802.1ad – Q-in-Q
- 802.1af – MAC key security
- 802.1ag – OAM
- 802.1ah – MAC-in-MAC
- 802.1aj – 2-port MAC relay
- 802.1au – congestion notification

802.1 Baggy pants model



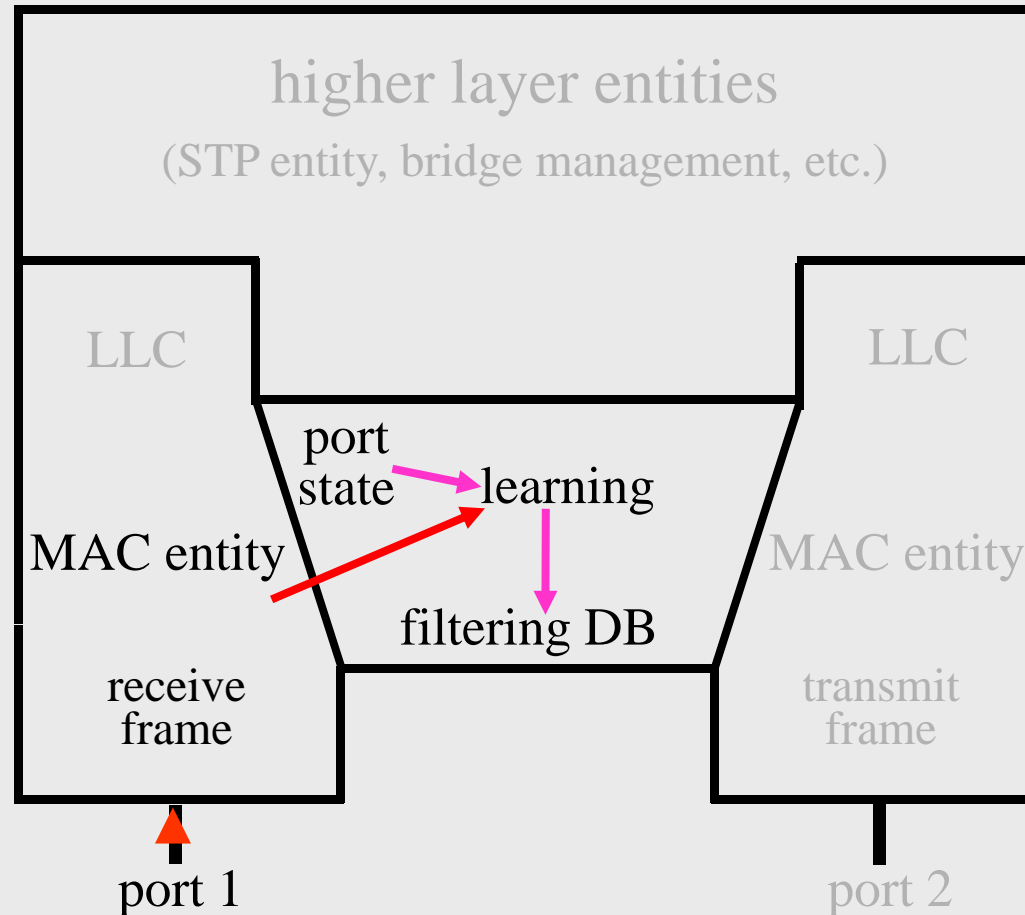
Note: a bridge must have at least 2 ports
here we depict exactly 2 ports

Baggy pants - forwarding



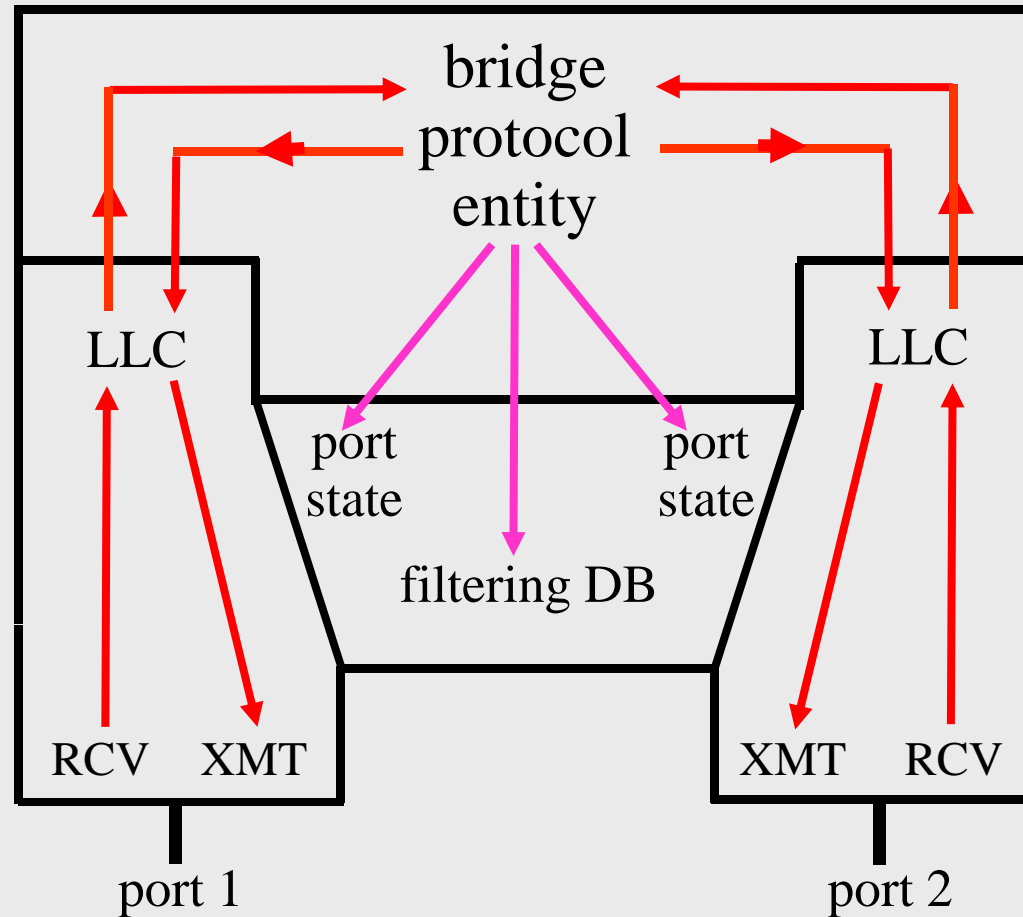
Note: relay entity passes frame to port 2
dependent on *port state* and *filtering database*

Baggy pants - learning



Note: we do not show forwarding of packet that *may* occur

Baggy pants - STP



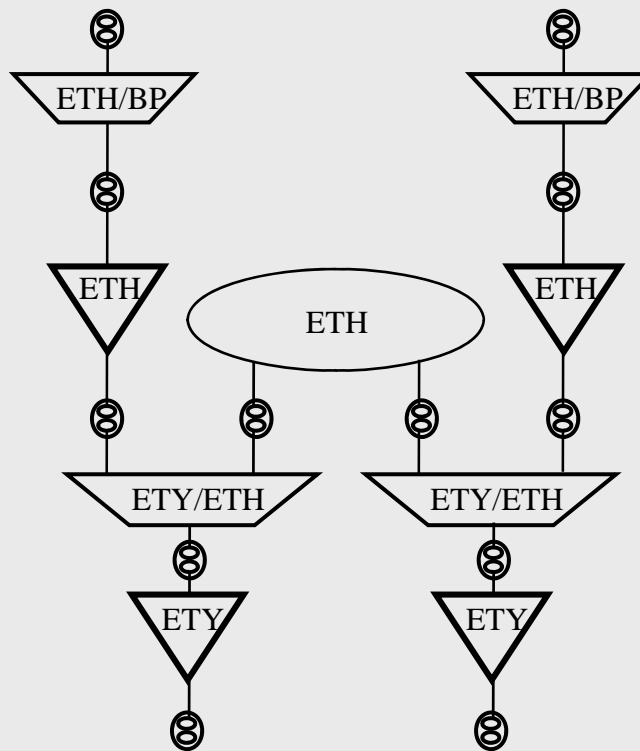
Note: PDUs are sent and received by the bridge protocol entity
bridge protocol entity updates filtering DB and port states

Translation to G.805

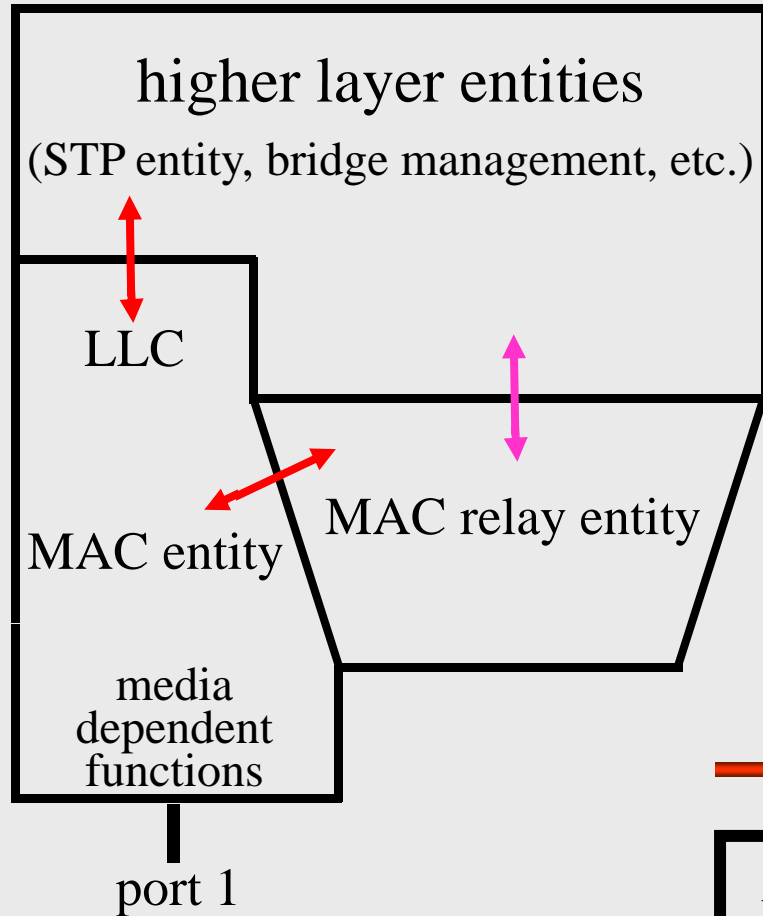
we can redraw the baggy pants model per G.805

(from G.8010 Appendix II)

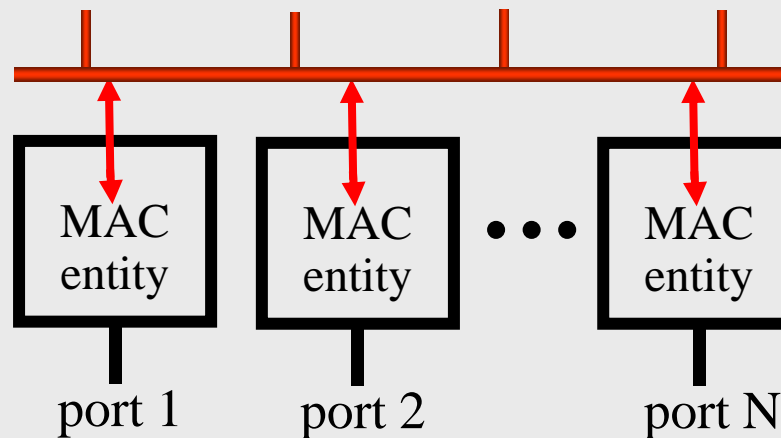
Note: drawn for CO case only



Extension to N ports



in the baggy pants diagram
 port 1 and port 2 are identical
 so it is enough to draw once
 if there are many ports
 the relay entity becomes
 an internal LAN !



ITU-T view

the name *Ethernet* disguises many different layer networks

ETH (MAC layer) is a packet/frame CO/CL network

there is also a VLAN variant called ETH-m

ETH can run over various *server layers*, including ETY

ETY (PHY layer) has a number of options

ETY_n n = 1, 2.1, 2.2, 3.1, 3.2, 3.3, 4

- ETY1 : 10BASE-T (twisted pair electrical; full-duplex only)
- ETY2.1: 100BASE-TX (twisted pair electrical; full-duplex only; for further study)
- ETY2.2: 100BASE-FX (optical; full-duplex only; for further study)
- ETY3.1: 1000BASE-T (copper; for further study)
- ETY3.2: 1000BASE-LX/SX (long- and short-haul optical; full duplex only)
- ETY3.3: 1000BASE-CX (short-haul copper; full duplex only; for further study)
- ETY4 : 10GBASE-S/L/E (optical; for further study)

ITU-T Recommendations

G.8001 – EoT definitions

G.8010 – Ethernet layer network architecture

G.8011 – Ethernet over Transport services framework

G.8011.1 – Ethernet private line service

G.8011.2 – Ethernet virtual private line service

G.8012 – Ethernet UNI and NNI

G.8021 – Ethernet transport equipment characteristics

G.8031 – Ethernet linear protection switching




G.8032 – Ethernet ring protection switching

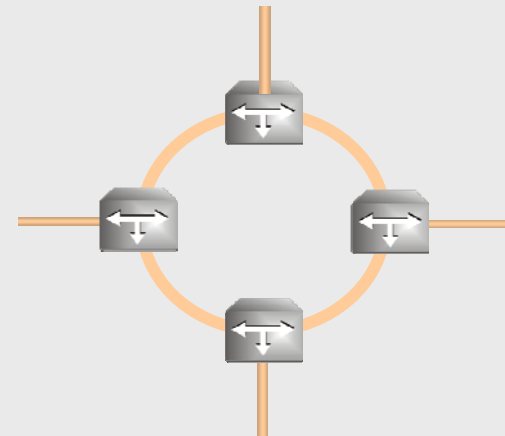
Y.1730 – Ethernet OAM - requirements

Y.1731 – Ethernet OAM

Ethernet servers

Ethernet can be carried over

- ETY_n {
- coaxial cable 
 - twisted copper pairs 
 - optical fibers 
-
- synchronous (TDM) networks
 - packet switched networks (PSN)



Ethernet over TDM

over SONET/SDH ([see EoS course](#))

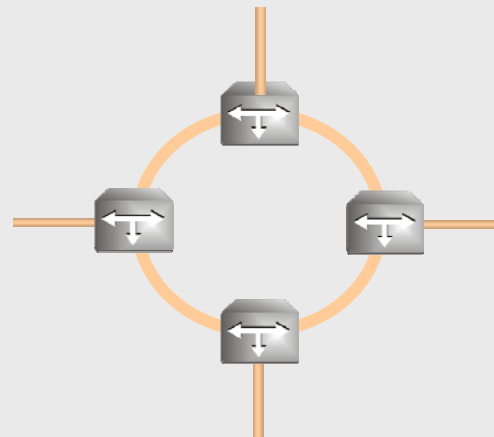
- PoS (PPP/HDLC)
- LAPS
- GFP

over low-rate TDM

- PPP/HDLC
- GFP

over OTN

- GFP



Ethernet over PSN

IP (EtherIP RFC 3378)

MPLS

Ethernet PW (RFC 4448, Y.1415)

see [PWE3 course](#)

L2VPN services (VPWS/VPLS)

see [VPLS course](#)

Ethernet (MAC-in-MAC 802.1ah)

ATM (LAN emulation)



ETH layer network

ETH is a packet/frame-based layer network

it maintains client/server relationships with other networks

networks that use Ethernet are Ethernet *clients*

networks that Ethernet uses are Ethernet *servers*

sometimes Ethernet ETY is the lowest server

i.e. there is no lower layer server network

ETH is usually *connectionless*

but *connection-oriented* variants have been proposed (PBT, PVT, etc)

ETH is a relatively simple layer network

it has no real forwarding operations

just filtering and topology pruning

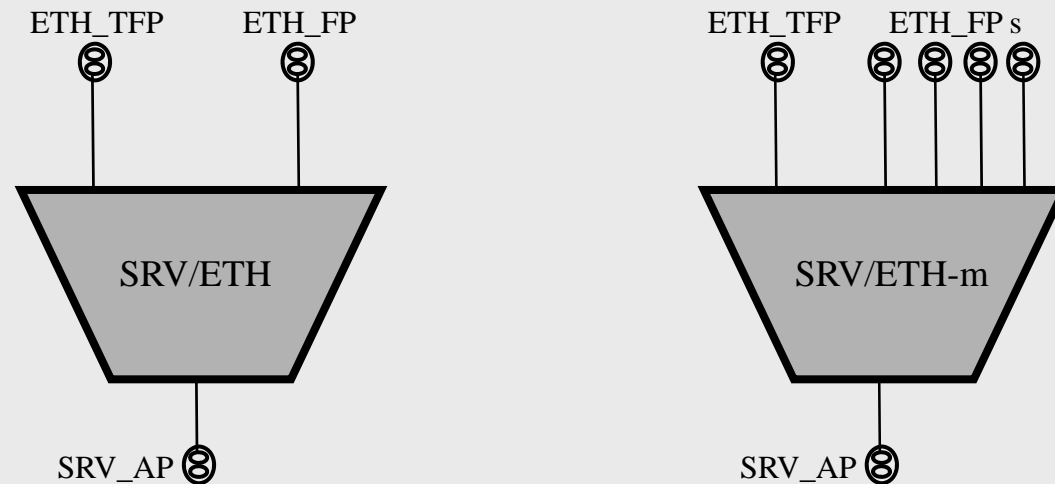
it has no real control plane

just STP, GARP, “slow protocol frames”, etc.

until recently it had no OAM

but now there are two

ETH adaptations



the adaptation from ETH to the server layer (e.g. ETY) has

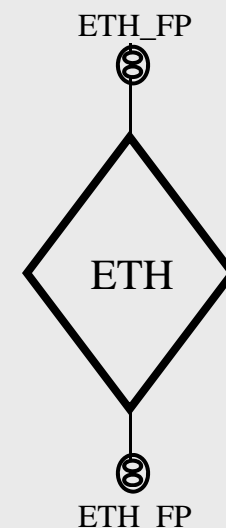
- 1 ETH Termination Flow Point responsible for DA, SA, P bits, OAM
- 1 (for ETH-m between 1 and 4094) ETH Flow Point(s) where the ETH CI enters
- 1 SRV Access Point (SRV can be ETY, but can be other server networks)

Traffic conditioning

G.8010 defines a new function (not in G.805/G.809)

traffic conditioning function:

- inputs CI
- classifies traffic units according to configured rules
- meters traffic units within class to determine eligibility
- polices non-conformant traffic units
- outputs remaining traffic units as CI



technically, the TC function is placed by expanding the ETH Flow Point

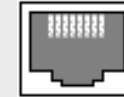
MEF view

MEF focuses on Ethernet as a service to a customer

the service is provided by a **M**etro **E**thernet **N**etwork (any technology / architecture)

the service is *seen* by the **C**ustomer **E**dge

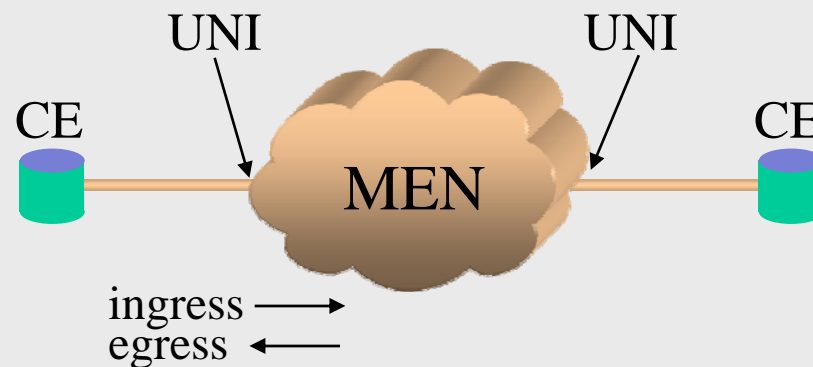
the **U**NI is the demarcation point between customer and MEN



each UNI serves a single customer, presents a standard Ethernet interface

at the UNI CE and MEN exchanged service (MAC) frames

connection between UNIs called an **E**thernet **V**irtual **C**onnection





MEF Technical Specifications

- MEF 1 Ethernet Services Model - Phase 1 (obsoleted by MEF 10)
- MEF 2 Requirements and Framework for Ethernet Service Protection
- MEF 3 Circuit Emulation Requirements
- MEF 4 MEN Architecture Framework Part 1: Generic Framework
- MEF 5 Traffic Management Specification – Phase 1 (obsoleted by MEF 10)
- MEF 6.1 Metro Ethernet Services Definitions (Phase 2)
- MEF 7.1 EMS-NMS Information Model (Phase 2)
- MEF 8 PDH over MEN Implementation Agreement (CESoETH)
- MEF 9 Abstract Test Suite for Ethernet Services at the UNI
- MEF 10.2 Ethernet Services Attributes (Phase 2)
- MEF 11 User Network Interface (UNI) Requirements and Framework
- MEF 12 MAN Architecture Framework Part 2: Ethernet Services Layer
- MEF 13 User Network Interface (UNI) Type 1 Implementation Agreement
- MEF 14 Abstract Test Suite for Ethernet Services at the UNI
- MEF 15 MEN Management Requirements - Phase 1 Network Elements
- MEF 16 Ethernet Local Management Interface
- MEF 17 Service OAM Framework and Requirements
- MEF 18 Abstract Test Suite for Circuit Emulation Services
- MEF 19 Abstract Test Suite for UNI Type 1
- MEF 20 UNI Type 2 Implementation Agreement
- MEF 21 Abstract Test Suite for UNI Type 2 Part 1 Link OAM
- MEF 22 Mobile Backhaul Implementation Agreement
- MEF 23 Class of Service Phase 1 Implementation Agreement
- MEF 24 Abstract Test Suite for UNI Type 2 Part 2 E-LMI
- MEF 25 Abstract Test Suite for UNI Type 2 Part 3 Service OAM
- MEF 26 External Network Network Interface - ENNI (Phase 1)

Other reference points

the UNI stands between the CE and MEN

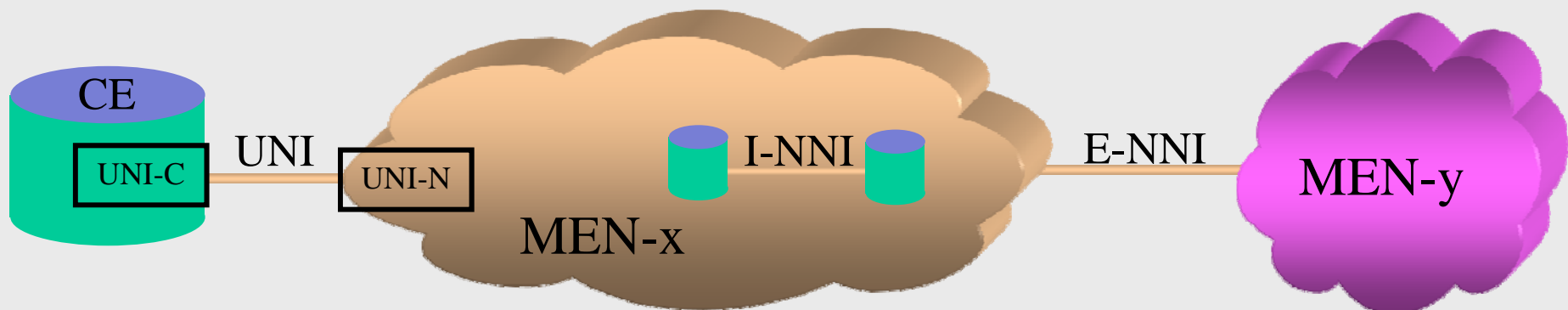
the processing functions needed at the CE to connect to the MEN are called UNI-C

the processing functions needed at the MEN to connect to the CE are called UNI-N

between networks elements of a MEN we have I-NNI interfaces

while between different MENs we have E-NNI interfaces

(MEF 4 also defines NI-NNI, SI-NNI and SNI interfaces)



EVCs

a public MEN can not behave like a shared LAN

since ingress frames must not be delivered to incorrect customers

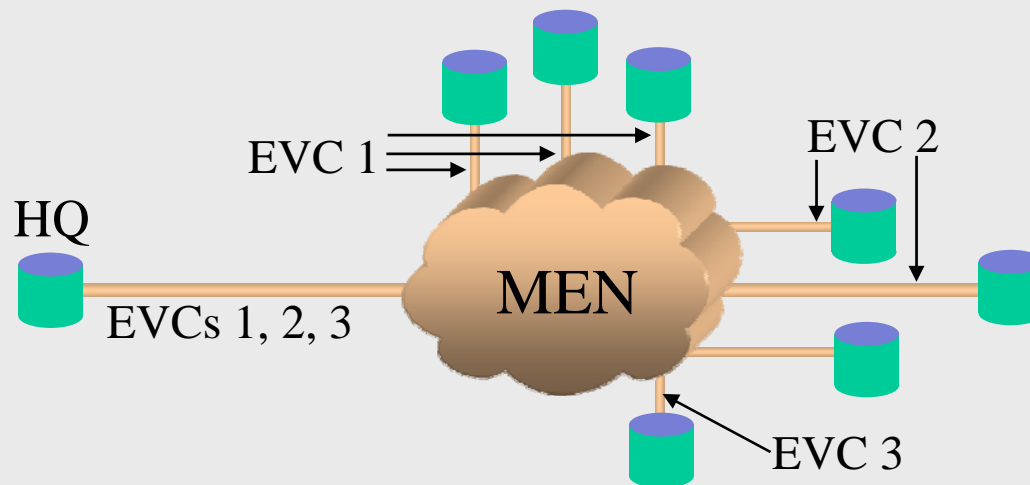
an association of 2 or more UNIs is called an **EVC**

ingress frames must be delivered only to UNI(s) in the same EVC

when several UNIs frames may be flooded to all or selectively forwarded

frames with FCS errors must be dropped in the MEN (to avoid incorrect delivery)

a single UNI may belong to several EVCs (differentiated by port and/or VLAN ID)



EVC types

a point-to-point EVC associates **exactly 2 UNIs**

- the service provided is called E-LINE [MEF-6]



a multipoint-to-multipoint EVC connects 2 or more UNIs

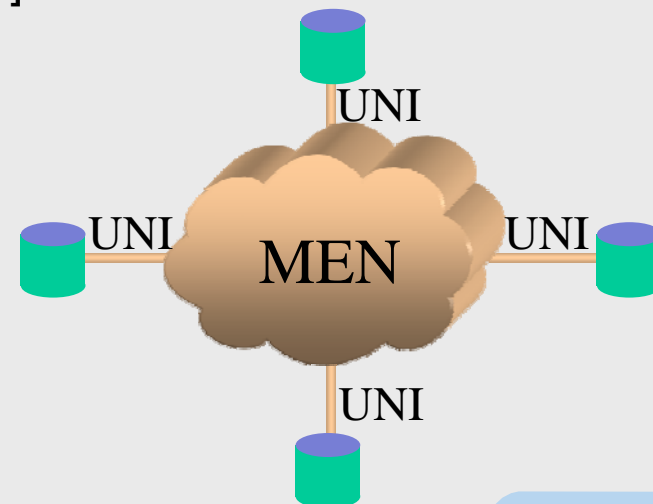
Note: MP2MP w/ 2 UNIs is different from P2P (new UNIs can be added)

unicast frames may flooded or selectively forwarded

broadcast/multicast frames are replicated and sent to all UNIs in the EVC

- the service provided is called E-LAN [MEF-6]

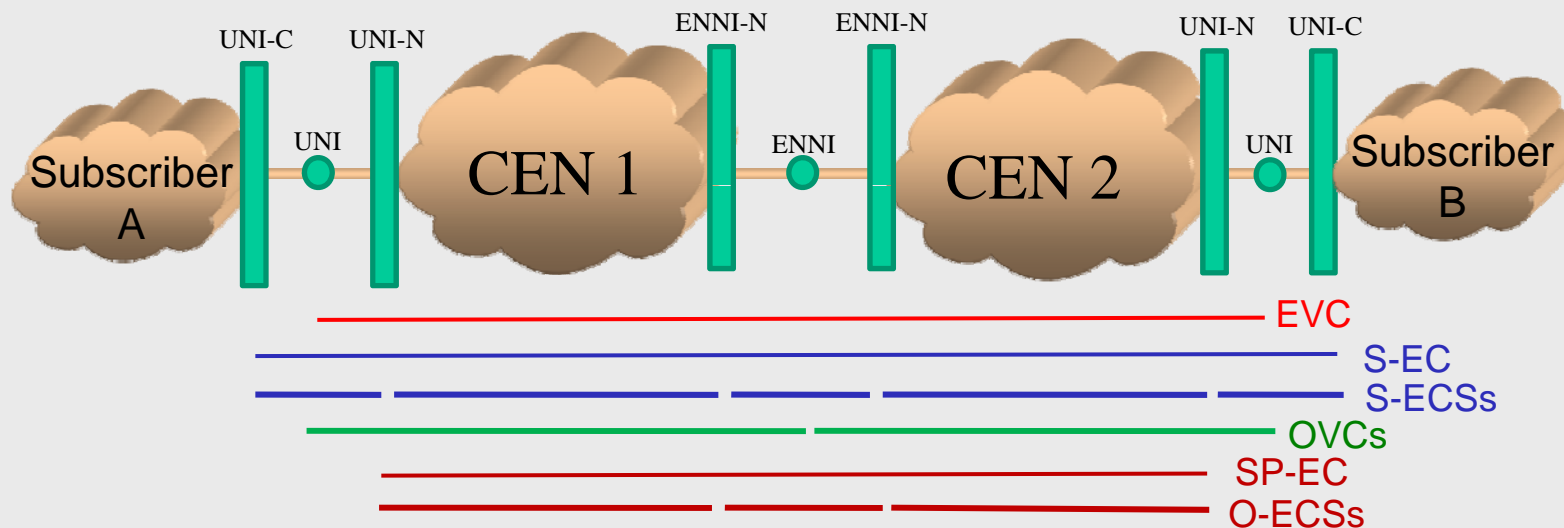
we will see more details on Ethernet services later



New MEF Model (12.1)

MEF is updating their architecture model

- Metro → Carrier so MEN → CEN
- EVC is no longer a transport entity
- Ethernet Connection (EC) is the entity that connects Ethernet flow termination points
- ECs traversing several CENs are composed of EC segments (ECS)
- Operator Virtual Connections (OVCs) connect ENNIs with other ENNIs or UNIs



What about the IETF?

Ethernet is often used to carry IP packets

since IP does not define lower layers

since IP only forwards up to the LAN, not to the endpoint

both IP and Ethernet use addresses

but these addresses are not compatible (exception – IPv6 local address)

the Address Resolution Protocol (RFC 826 / STD 37) solves this problem

if you need to know the MAC address that corresponds to an IP address

- broadcast an ARP request (Ethertype 0806, address FF...FF)
- all hosts on LAN receive
- host with given IP address unicasts back an “ARP reply”

Other ARP-like protocols

other related protocols (some use the ARP packet format)

- GARP (gratuitous ARP – **WARNING not 802.1 GARP**)
host sends its MAC-IP binding without request (e.g. backup server)
- Proxy ARP
router responds to ARP request to capture frames
- Reverse ARP, BOOTP, DHCP
host sends its MAC and wants to know its IP address
- Inverse ARP
frame-relay station unicasts DLCI to find out remote IP address
- ARP mediation
mediate over L2VPN between networks using different ARPs
(e.g. Ethernet on one side and FR on the other)

VLANs

VLANs

tagging (802.1Q)

SVL and IVL switches

VLAN stacking

PBN and PBBN

PBT

MPLS-TP

Virtual LANs

in standard practice each LAN needs its own infrastructure

- 1 broadcast domain per set of cables and hubs
- all stations on LAN see all traffic

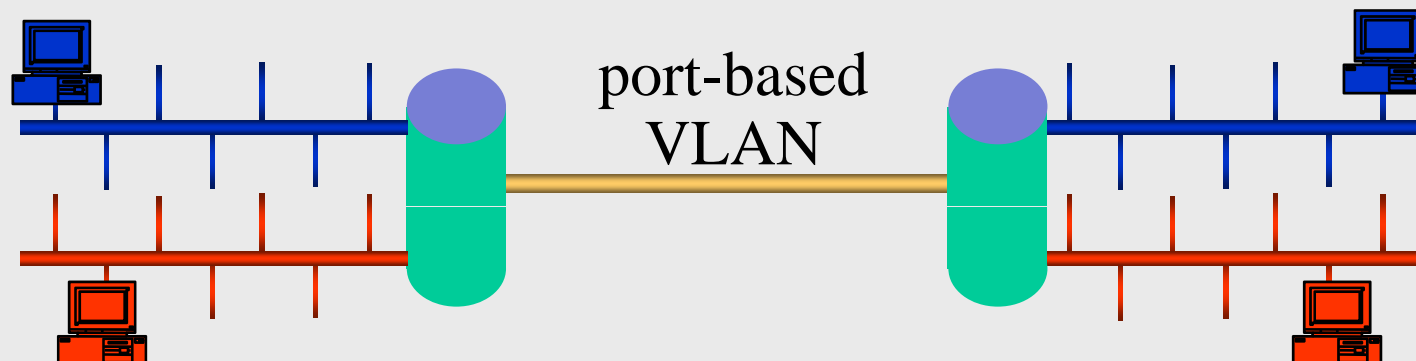
we may want a single physical infrastructure to support many LANs

- simpler and less expensive than maintaining separate infrastructures
- multiple low-speed LANs on one high-speed infrastructure
- segment broadcast domains (lower BW/processing) without routers
- security for different departments in company / groups in campus

separation may be based on switch ports or MAC address or VLAN ID (tag)

we will not delve deeply into VLANs here (see e.g. 802.1Q Appendix D)

I assume that this is treated in elementary Ethernet course



Virtual LANs (cont.)

initially there were proprietary solutions to tagging

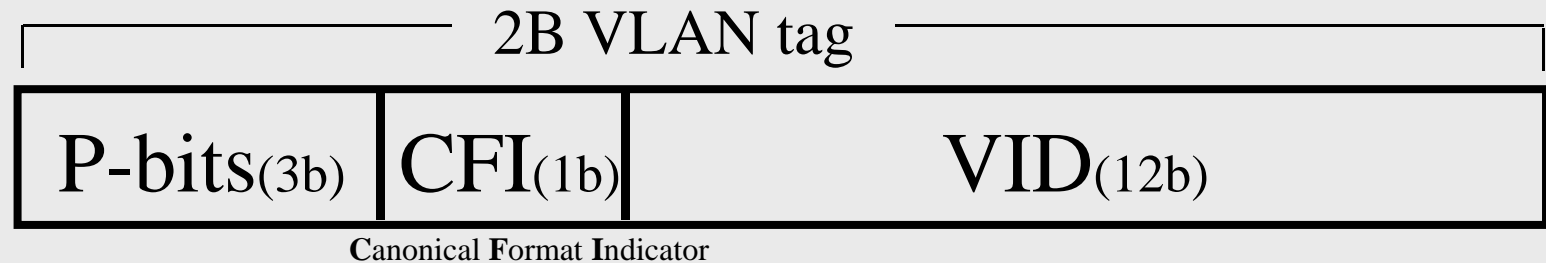
802.1Q & 802.1p projects defined format, protocols, and procedures

- 802.1p results were incorporated into 802.1D-1998
 - priority
- 802.1Q intentionally left separate and NOT incorporated considered sufficiently distinct from non-VLAN bridging
 - in particular, baggy pants model enhanced
 - new protocol – GVRP (see below)

802.1ad and 802.1ah further extend tagging formats and procedures



VLAN ID (VID)



802.1Q mandates 12 bit VID (carried after Ethertype 8100)

- 2 bytes carry P (priority) bits, CFI (not important here, always 0) and VID
- 4094 possible VID values (0 and 4095 are reserved)
- VID=0 frames are priority tagged, able to carry P bits

VLAN-aware switches

- take VID into account when forwarding
- perform VID insertion/removal
- never output a priority-tagged frame

when VLAN-aware switch receives

- VLAN tagged frame – treats according to VID
- untagged frame – may push permanent VID (PVID) of receive port
- priority-tagged frame treated like untagged frame (VLAN tag MAY be added)

Insertion / removal of VLAN tag necessitates recomputing FCS and adjusting padding

VLAN switch operation

A VLAN-aware switch performs 5-stage processing:

- ingress rule checking
 - 2 modes: *admit only tagged*, *admit all*
 - classify every incoming frame to a VID (if untagged to PVID)
 - discard frame not obeying rules, e.g.
 - port not in VID member set
 - untagged with admit only tagged
- active topology enforcement
 - check that frame *should* be forwarded, discard if, e.g.
 - spanning tree forbids forwarding or forwarding is back to port of origin
 - port not in forwarding state
 - MTU exceeded
- frame filtering (according to MAC, VID and filtering DB entry)
- egress rule checking
 - discard if VID not in member set
 - add/remove tag as needed
- queuing for transmission

SVL (SFD) switches

MAC address	PORT

Shared VLAN Learning
(Single Forwarding Database)

how do VLANs interact with 802.1D (forwarding and learning) ?

there are two different answers to this question

SVL switches still maintain a single 802.1D filtering database
but use VLAN in ingress/egress classification

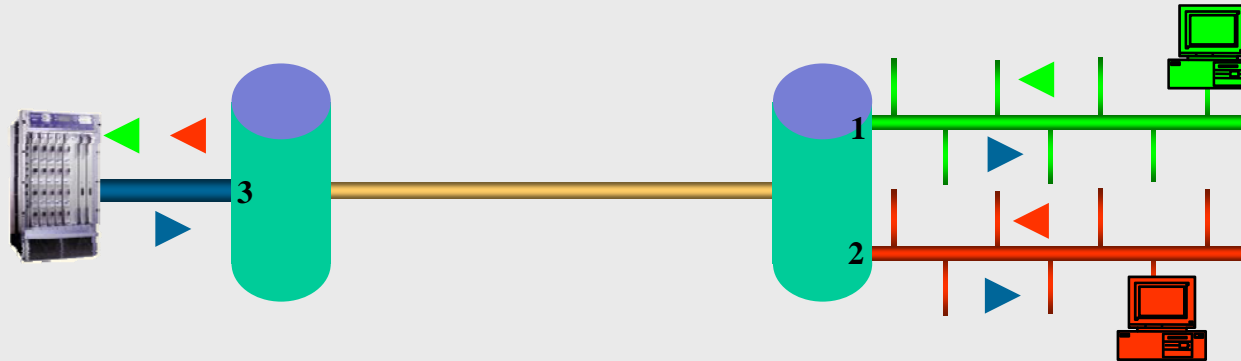
MAC addresses are learnt together for all VLANs (flood MAC once)

path to MAC does not depend on VLAN

a MAC address belongs to at most 1 VLAN

asymmetry possible

Asymmetry case



green PC can talk to server

- untagged frames on port 1 are tagged with VID=1
- tags removed when sent to server

red PC can talk to server

- untagged frames on port 2 are tagged with VID=2
- tags removed when sent to server

server can reply to both

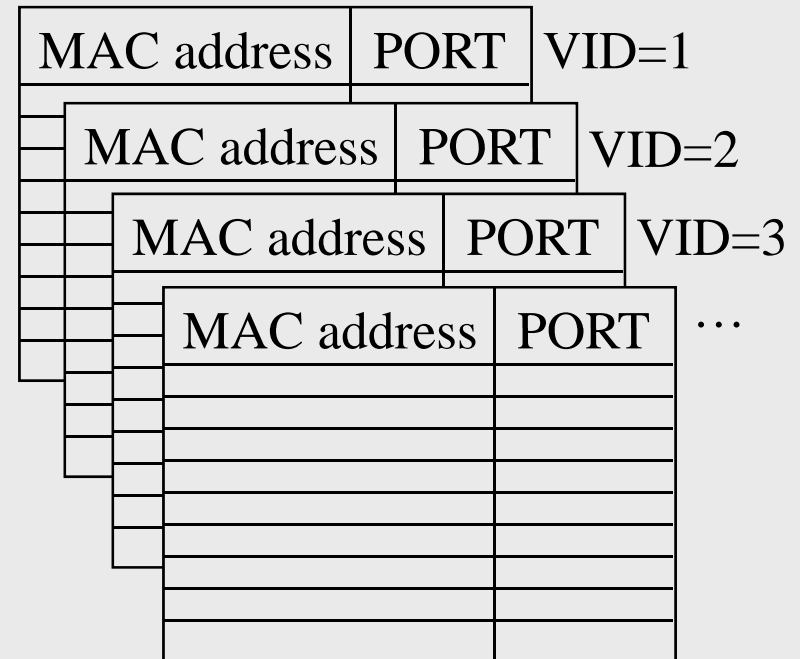
- untagged frames are tagged with VID=3
- tags removed when sent out of ports 1 or 2

but green and red PCs can NOT talk to each other

- VID=1 traffic can not be sent to red PC with MAC associated with VID=2
- VID=2 traffic can not be sent to green PC with MAC associated with VID=1

IVL (MFD) switches

Independent VLAN Learning
(Multiple Forwarding Databases)



IVL switches maintain separate 802.1D filtering databases per VID

MAC addresses are learnt independently for each VID (more flooding)

1 MAC address can belong to several VLANs

- but path to MAC depends on VID

asymmetry impossible

Note: IVL switch can be implemented as 60 (48+12) bit lookup

VLAN stacking

we tag Ethernet frames by using Ethertype 8100

DA	SA	8100	VLAN 1	type	data	pad	FCS
----	----	------	--------	------	------	-----	-----

first Ethertype is set to 8100

second Ethertype is payload protocol (as in untagged frame)

but what if we add another Ethertype to 8100 ?

DA	SA	8100	VLAN 2	8100	VLAN 1	type	data	pad	FCS
----	----	-----------------	--------	------	--------	------	------	-----	-----

this is called VLAN stacking

or (for obvious reasons) Q-in-Q

stacking is not mentioned in 802.1Q

but not ruled out either!

Warning:

although superficially Q-in-Q looks like an MPLS stack, there is no network layering here – the DA remains the same!

Provider Bridges – 802.1ad

why stack VLANs?

1 reason is for Provider Bridge Networks

customers may use VLANs for internal reasons

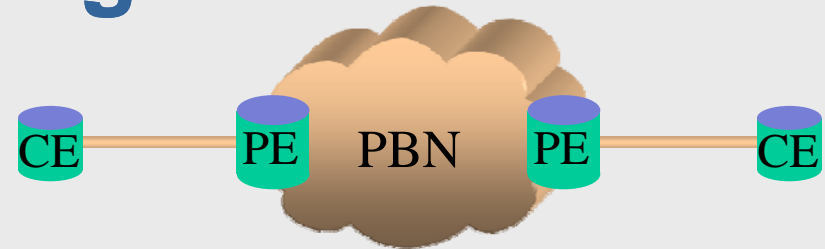
and will want them maintained across the PBN

but what if multiple customers use the same VLANs ?

the provider must either:

- manage customer VLANs
 - allocate ranges to customers
 - swap at ingress and egress
- or
- treat them as Customer VLANs, and push a Service VLAN (customer ID)

802.1ad
approved Dec 2005
published May 2006



CVID and SVID

customer frames have C-TAGs, may only have **priority** S-TAGs

PBN frames have S-TAGs and usually have C-TAGs

- C-TAG contains a C-VID
- S-TAG contains an S-VID

C-TAGs are standard format VLAN tags

802.1ad S-TAGs are similar, but use Ethertype 88A8 since they have Drop Eligible Indicator instead of CFI

service transparency

- PE inserts S-TAG, C-TAG becomes invisible to PBN
- PE removes S-TAG, C-TAG becomes visible to customer network

88A8 (S-TAG)	P	D E I	S-VID
8100 (C-TAG)	P	C F I	C-VID

Problems with PBNs

Q-in-Q PBNs simplify provider-customer interface, but

- are limited to 4K SVIDs, i.e. 4K customers
(VLANs derive from enterprise (not carrier) applications)
- do not provide true client/server (decoupled) relationship
 - customer VIDs are hidden
but not customer MAC addresses
 - no true OAM trace functionality
- provider switches still have to learn customer MAC addresses

we would really like full decoupling, i.e. a client/server relationship

this can be done via MAC-in-MAC

Progress to MAC-in-MAC

802.1D

DA	SA	T	payload	FCS
----	----	---	---------	-----

802.1Q

DA	SA	8100	VLAN	T	payload	FCS
----	----	------	------	---	---------	-----

802.1ad

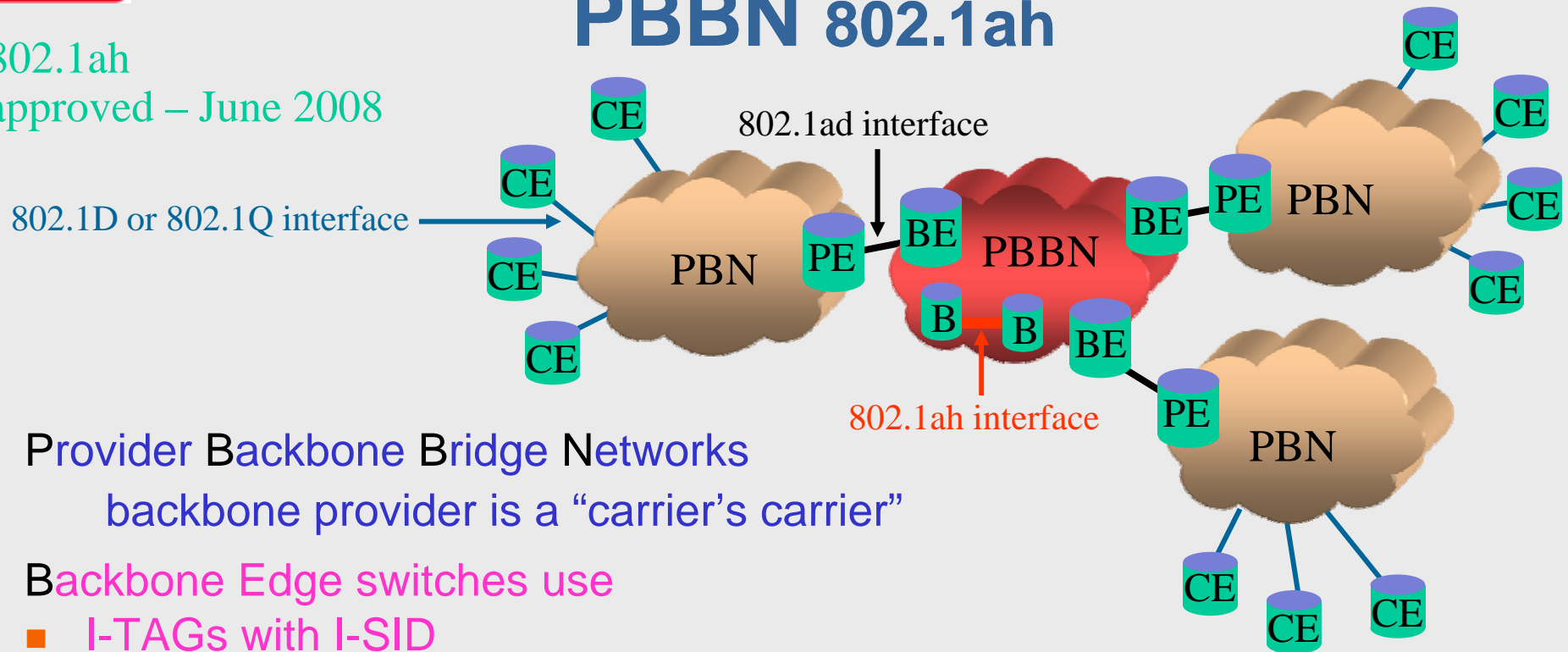
DA	SA	88A8	S-TAG	8100	C-TAG	T	payload	FCS
----	----	------	-------	------	-------	---	---------	-----

802.1ah

B-DA	B-SA	88A8	B-TAG	TBD	I-TAG	...		
DA	SA	88A8	S-TAG	8100	C-TAG	T	payload	FCS

PBBN 802.1ah

802.1ah
approved – June 2008



Provider Backbone Bridge Networks
backbone provider is a “carrier’s carrier”

Backbone Edge switches use

- I-TAGs with I-SID
- B-TAG with B-VID
- one or more I-TAGs and a B-TAG

I-TAG is a new format, I-SID is a 24-bit label (no more 4K limitation)



PBT (PBB-TE) 802.1Qay

Provider Backbone Transport builds on 802.1ah PBBNs
to achieve reliable, deterministic, carrier-class behavior

802.1ah gives backbone provider its own addressing space (MAC + VLAN)
addresses not shared with customer (no need to learn customer MACs)
and so provider is free to use address space as it sees fit

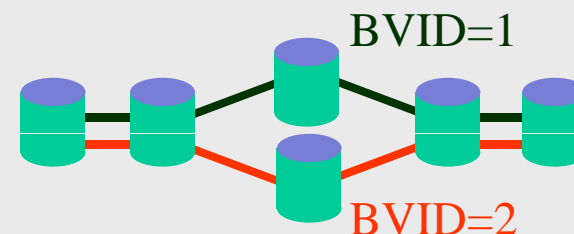
we saw before that IVL switches forward based on 60-bit addresses
but their capabilities limited due to limitations of STP, flooding, etc

but most IVL switches allow static FID (preconfigured forwarding behavior)
and support turning off learning/STP/flooding/aging

we can thus set up pure connection-oriented topology

we can use management (OSS)

or control plane protocols (e.g. RSVP-TE) to populate FID tables
(IETF GELS/CCAMP, ITU G.pbt)



What about MPLS ?

another way of carrying customer Ethernet frames over a provider network is to use MPLS (Ethernet PW, VPWS, VPLS)

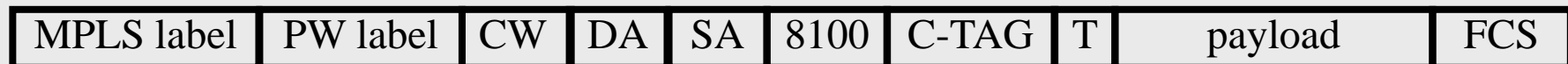
(see [MPLS](#), [PWE](#) and [VPLS](#) courses)

also complete decouples customer and provider networks

furthermore, MPLS is carrier-class (deterministic, CO, OAM, etc.)

and widely deployed (for IP and L3VPN applications)

MPLS can be carried over carrier SONET/SDH networks



PBT vs. MPLS

- both achieve complete client/server decoupling
- PBT exploits existing IVL switches
 - but many switches today can do MPLS forwarding
- Ethernet frame has SA, so OAM traceback easy
 - MPLS swaps labels and contains only destination semantics so need OAM for CV and mismerge detection
- PBT can co-exist with bridged Ethernet
- MPLS has highly developed control plane
 - Ethernet presently requires manual configuration
 - can use OSS, but not standardized
 - may be able to use GMPLS control plane (GELS)
- MPLS can transport non-Ethernet clients (IP, PWs)

More alternatives

there are other ways to

- set up CO connections
- avoid use of customer MAC addresses

MPLS-TP (Transport Profile)

- add an MPLS label to the ETH overhead
- in provider network switch solely based on label

similar to regular MPLS, but

- performed by *switch* instead of *LSR*
- various transport extensions such as linear/ring protection, OAM

TRILL (Transparent Interconnection of Lots of Links) - see later on

- invented by Radia Perlman (inventor of STP and IS-IS)
- uses IS-IS directly on MAC addresses (no need to configure IP addresses)
- adds outer MAC header + shim header (with TTL)
- is completely "no-touch" (plug-and-play).
- finds optimal paths

Ethernet services

EVCs and bundling

Ethernet Private Line (EPL)

Ethernet Virtual Private Line (EVPL)

Ethernet Private LAN (EPLAN)

Ethernet Virtual Private LAN (EVPLAN)

Ethernet services

we previously defined

- E-LINE point-to-point layer 2 service
- E-LAN multipoint-to-multipoint Ethernet service

but MEF and ITU have gone a step further

MEF 6 splits E-LINE into EPL and EVPL

ITU followed - Recommendations: G.8011.1 and G.8011.2
and E-LAN can be split into EPLAN and EVPLAN

these distinctions are made in order to live up to SLAs
i.e. provide defined service attributes

EVCs revisited

in our previous discussion of EVCs we didn't mention VLANs
we now realize customer EVCs can be distinguished by VLAN IDs

if the transport infrastructure is ETH, there may be an SVID

if the customer wants to have several EVCs, there will be a CVID
(here we simply mean the customer's 802.1Q VLAN ID)

the provider may promise "VLAN preservation"

i.e. not change CVIDs (untagged remain untagged)

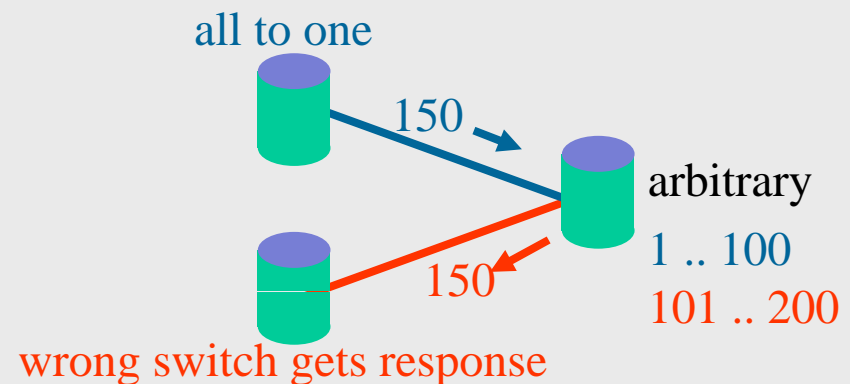
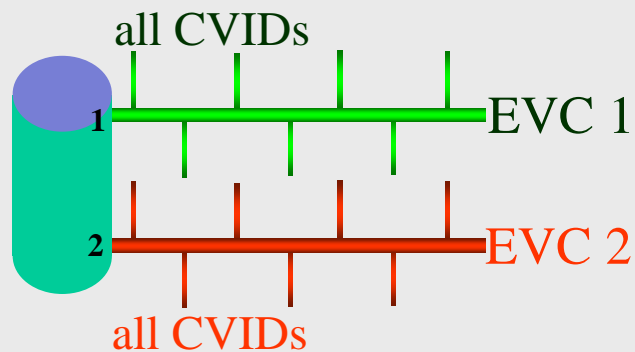
at the UNI-N there will be a **CVID to EVC map** (see MEF 10.1)

there can be three types of maps:

- all to one
- one to one (not MEF 10 term)
- arbitrary (not MEF 10 term)

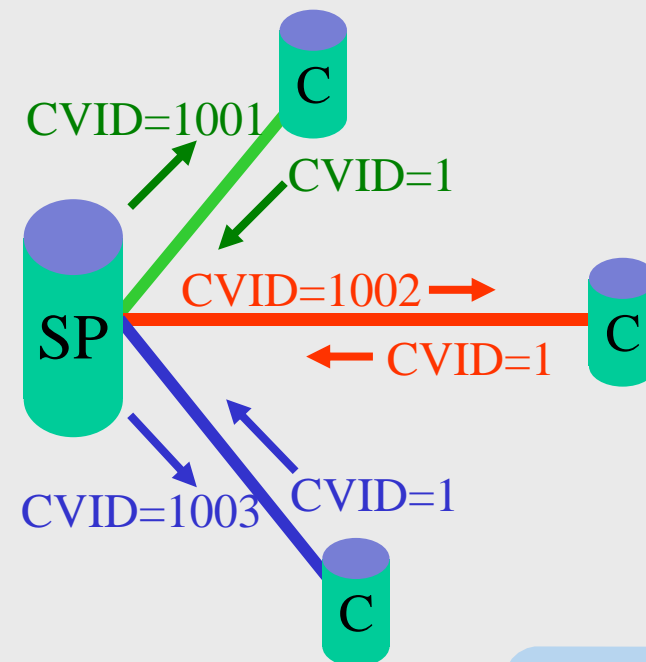
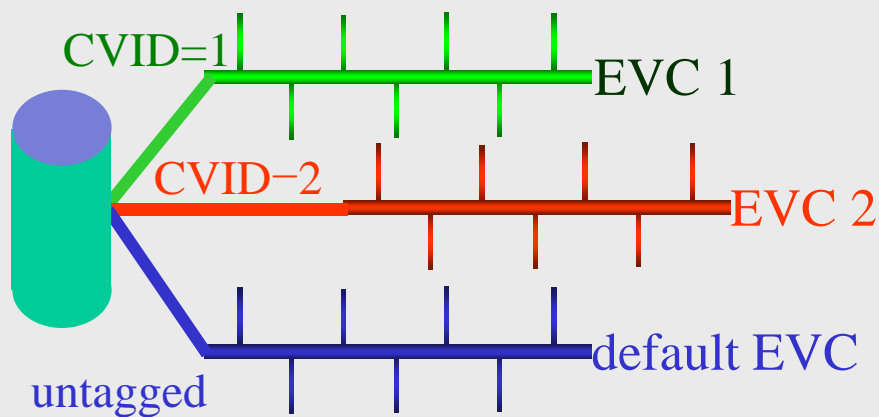
All to one bundling map

- all frames (independent of CVID) are mapped to the same EVC
- VLAN preservation
- no need for customer-provider coordination
- when p2p: similar to leased line
- when mp2mp: similar to TLS
- support multiple customer EVCs by different switch ports
- can't mix all to one switch with other map types in single EVC



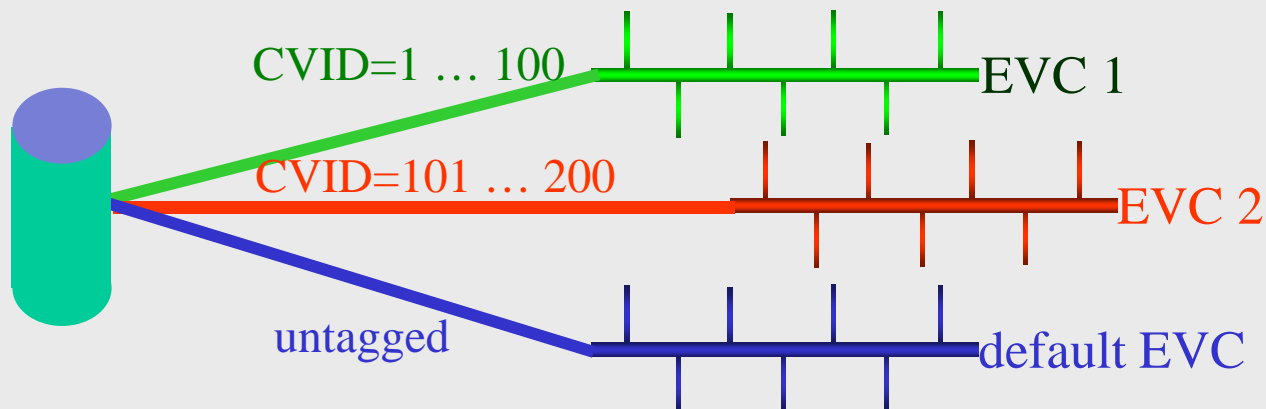
One to one bundling map

- each CVID mapped to a different EVC
- untagged (and priority tagged) mapped to default EVC
- support multiple EVCs from a single switch port
- no VLAN preservation can makes customer configuration easier
- similar to frame relay (DLCI identifies PVC)



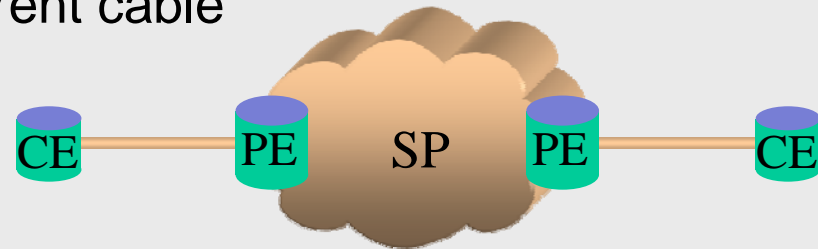
Arbitrary bundling map

- multiple CVIDs per EVC
- but multiple EVCs per port
- VLAN preservation
- can ensure customer traffic only goes to sites where VLAN is needed
- more efficient BW use
- need customer and SP coordination as to CVID – EVC map
- can coexist with one to one switches on same EVC



EPL

Ethernet Private Line is a dedicated-BW E-LINE (p2p) service
transport network seems to be a transparent cable
no frame loss unless FCS errors



transport layer may be

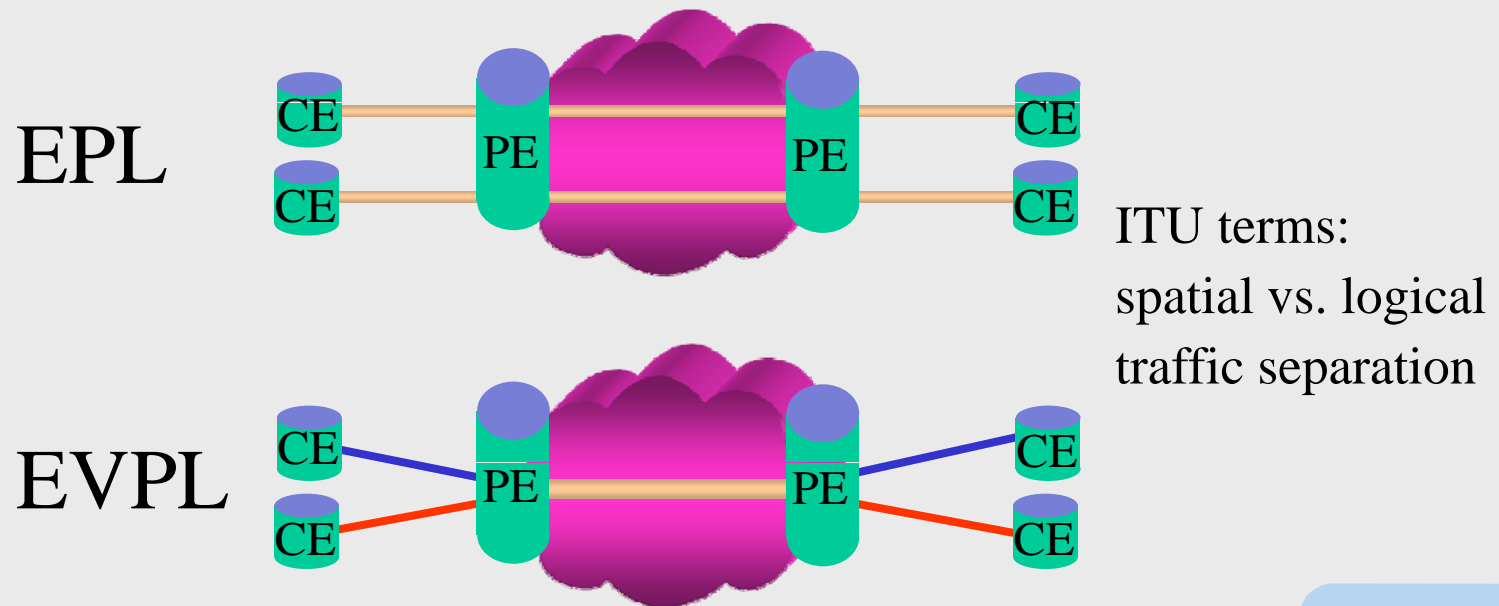
- native Ethernet – dedicated to single EVC (e.g. 100 Mb/s)
- TDM or SONET/SDH timeslot (e.g. X.86, GFP)
- VPWS service in TE tunnel

ITU further divides EPL into

- type 1 – terminate ETY, transport MAC frame over server (SDH/GFP-F, MPLS)
- type 2 – transparent transport (e.g. GFP-T)
- native (special case for 10GBASE-W)

EVPL

Ethernet Virtual Private Line is a shared-BW E-LINE (p2p) service
statistical multiplexing of user traffic, marked by VLAN IDs
(actually, all resources are shared – constraint may be switch fabric computation)
there may be frame loss due to congestion
normally a policing function is required at SP network ingress

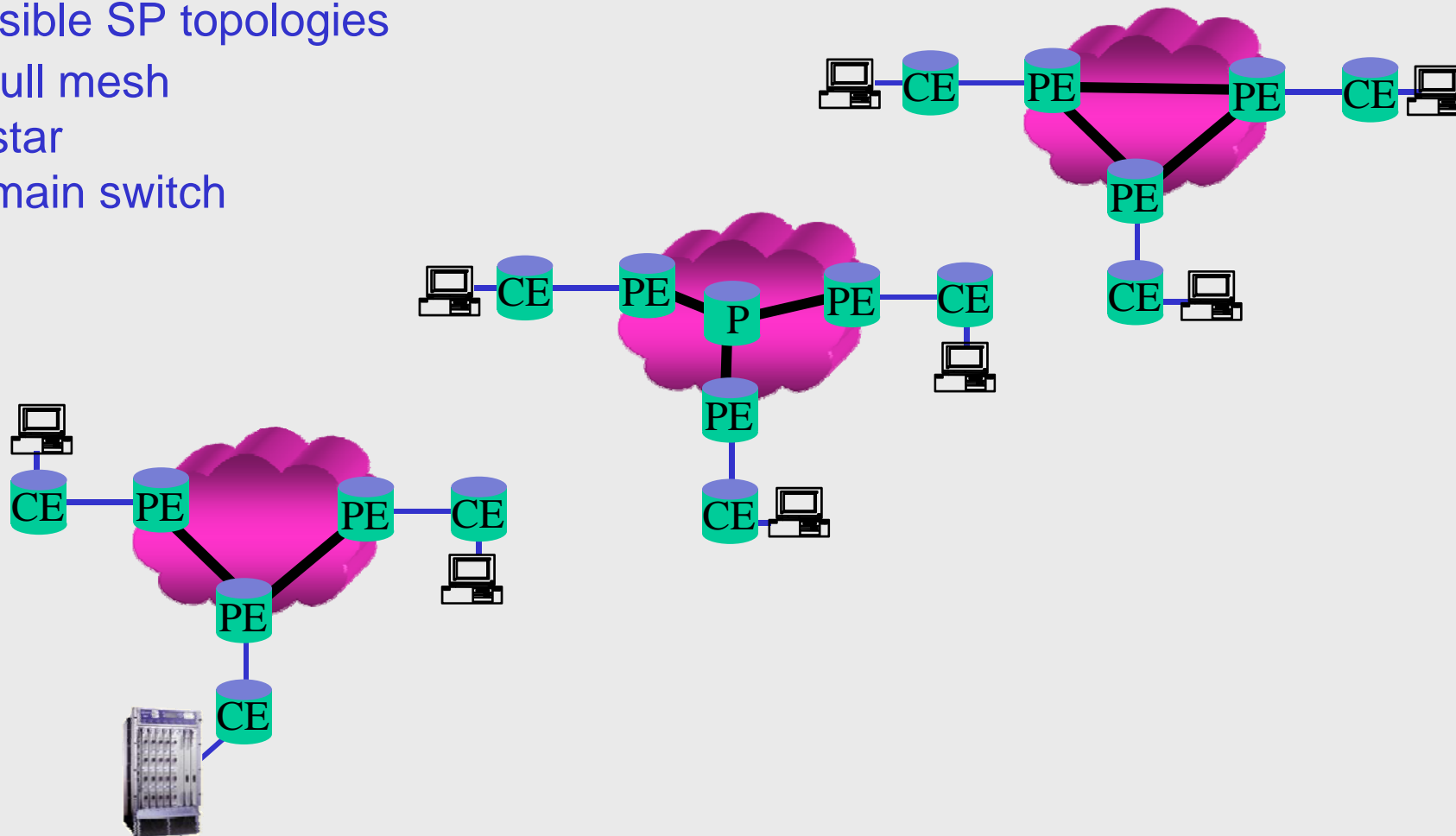


EPLAN

Ethernet Private LAN is a dedicated-BW E-LAN service

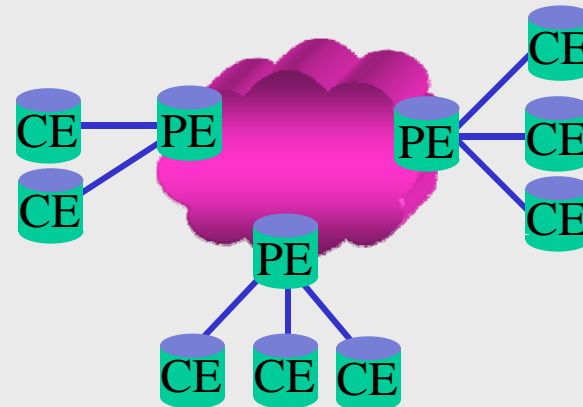
possible SP topologies

- full mesh
- star
- main switch



EVPLAN

Ethernet Virtual Private LAN is a shared-BW E-LAN service
statmuxed BW and switch fabric are shared among customers
useful service, but most difficult to manage (not yet studied)
when server is MPLS, this is VPLS
best effort version is widely deployed



Additional bridging functions

GARP, GVRP, GMRP (802.1ak)

rapid spanning tree (802.1w)

multiple spanning trees (802.1s-2002)

Rbridges

GARP (802.1D clause 12)

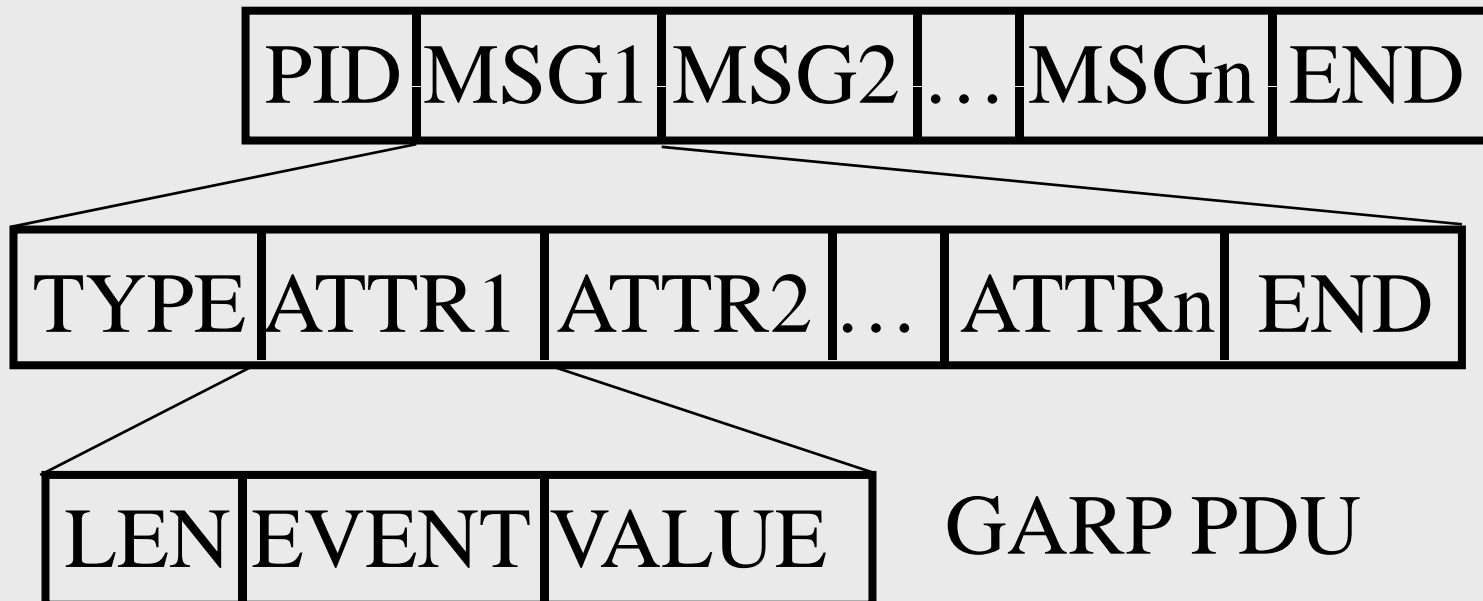
Generic Attribute Registration Protocol (WARNING not Gratuitous ARP)

generic framework to declare, withdraw, register and remove attributes

GARP defines architecture, protocol, state machine

Example uses:

- VLAN IDs (GVRP)
- multicast group membership (GMRP)



GVRP (802.1Q clause 11)

GARP VLAN Registration Protocol performs automatic VLAN configuration to properly process VLAN tagged frames, the VLAN switches need to know

- the VLANs in which it participates
- which ports to use for VLAN members

VLAN configuration of all switches needs to be consistent

802.1Q allows:

- static provisioning of VLAN membership information (via management mechanisms)
- dynamic configuration and distribution of VLAN membership info

to add a new switch to a VLAN:

- with static provisioning need to configure every switch
- with GVRP need to configure only one switch
GVRP then sends out info needed to configure all the other switches

GMRP (802.1D clause 10)

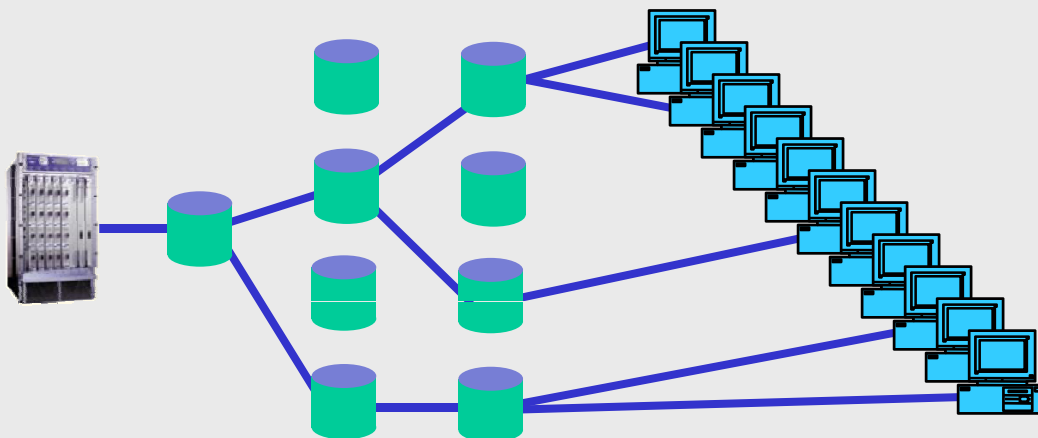
GARP Multicast Registration Protocol distributes multicast group info frames with multicast address

need to be replicated and sent to all members of the multicast group

GMRP enables automatic registering and deregistering

FIDs ensure that multicast frames are only sent to bridges that need them

GMRP must find a sub-tree of the spanning tree



RSTP (802.1w)

Rapid Spanning Tree Protocol (AKA rapid reconfiguration)

RSTP configures the state of each switch port
in order to eliminate loops

STP may takes minutes to (re)converge
goal of RSTP is 10 ms. convergence

RSTP is an evolutionary update of STP
new algorithm
same terminology
mostly same parameters
backwards compatible with STP

but

additions to BPDU format (all 8 bits of flag byte used)
simplified port *states*
new variable holding the port *role*

802.1w incorporated into
802.1D-2004 clause 17

it supersedes the previous
STP and *STA*

RSTP states and roles

The 802.1D concept of *port state* includes both

- forwarding state (blocks or forwards traffic) and
- topology role (root port, designated port).

802.1w decouples the concepts

802.1D has 5 port states

disabled }
 blocking }
 listening }
 learning }
 forwarding

802.1w has only 3

discarding

learning

forwarding

802.1D defines a concept of port role, but has no matching variable

Spanning Tree Algorithm determines role based on BPDUs

802.1w defines 4 port roles

- root
- designated
- backup
- alternate

multiple spanning trees 802.1s

conventionally, all VLANs use the same spanning tree
(even if IVL switches use different FIDs)

so links blocked by STP will **never** carry **any** traffic

we **can** utilize these links

if different VLANs could use different spanning trees

Multiple Spanning Tree Protocol - 1998 amendment to 802.1Q

the protocol and algorithm are now in 802.1Q-2003 clauses 13 and 14

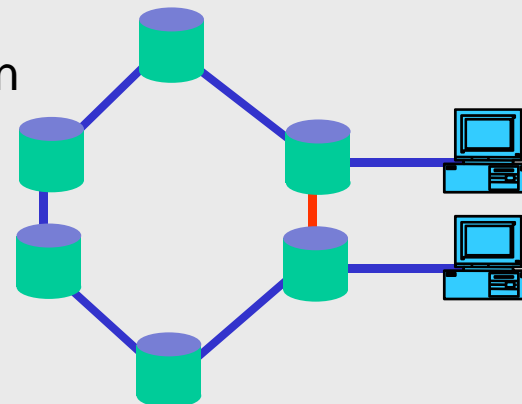
MSTP configures a separate spanning tree for each VLAN

blocks redundant links separately for each spanning tree

Cisco has its own **Per VLAN Spanning Tree (PVST and PVST+)** protocols

Rbridges (IETF TRILL WG)

Spanning tree is a *clean* protocol - needs no configuration
but STP converges slowly
and may make inefficient trees
 hosts that are actually close become far apart
we *could* use IP routing protocols
 but that requires allocating IP addresses, etc.



A new solution **TR**ansparent **I**nterconnection of **L**ots of **L**inks

defines a combination of router and bridge called an Rbridge
that run a link state protocol (e.g. OSPF, IS-IS)

Rbridges have the advantages of both with the disadvantages of neither

- optimized paths
- but no configuration
- no IP layer

Algorhyme

*I think that I shall never see
a graph more lovely than a tree.*

*A tree whose crucial property
is loop-free connectivity.*

*A tree that must be sure to span
so packet can reach every LAN.*

*First, the root must be selected.
by ID, it is elected.*

*Least-cost paths from root are traced.
in the tree, these paths are placed.*

*A mesh is made by folks like me,
then bridges find a spanning tree.*

Radia Perlman

Algorhyme v2

*I hope that we shall one day see
a graph more lovely than a tree.*

*A graph to boost efficiency
while still configuration-free.*

*A network where RBridges can
route packets to their target LAN.*

*The paths they find, to our elation,
are least cost paths to destination.*

*With packet hop counts we now see,
the network need not be loop-free.*

*RBridges work transparently.
without a common spanning tree.*

Ray Perlner

QoS Aspects

Flow control (PAUSE frames)

handling QoS

prioritization (802.1p)

MEF service attributes

Flow control

When an Ethernet switch receives traffic faster than it can process it it needs to tell its immediate neighbor(s) to slow down

On half-duplex links the *back pressure* can be employed

- overloaded device jams the shared media by sending preambles or idle frames
- detected by other devices as collisions causing senders to wait (CSMA/CD)

On full-duplex point-to-point links, *PAUSE frames* are sent

Since they are sent on a point-to-point link, the DA is unimportant, and the standard multicast address 01-80-C2-00-00-01 is used making the PAUSE frame easy to recognize

The PAUSE frame encodes the requested pause period as a 2-byte unsigned integer representing units of 512 bit times

Handling QoS

Ethernet switches have **FIFO** buffers on each port's input and output

But prioritization may be needed

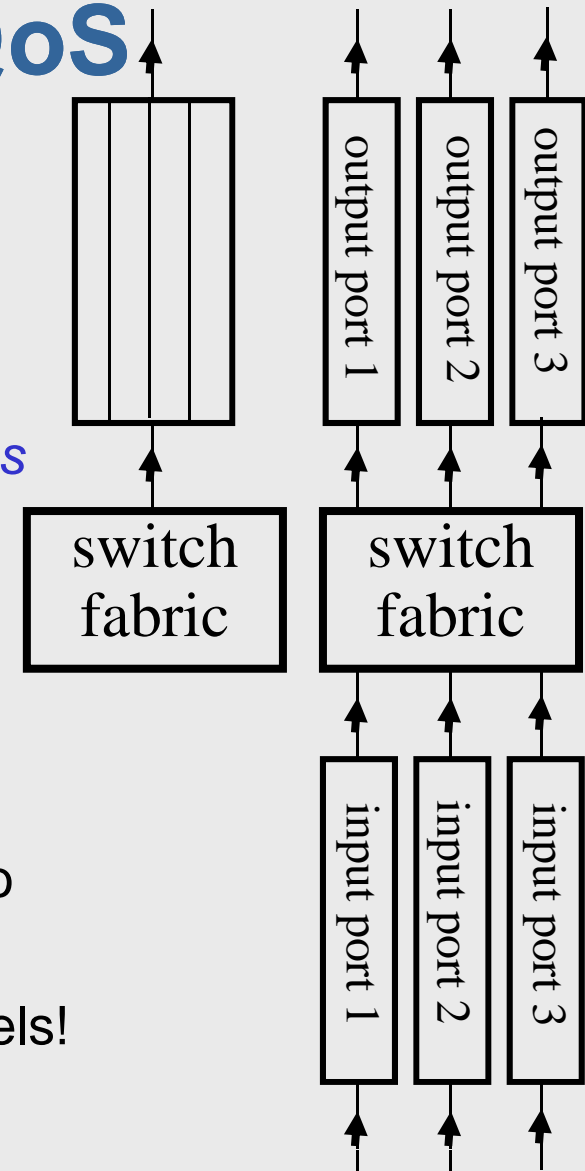
so output buffers may be divided into multiple *queues* outgoing frames put into queues of specified priority

we *could* base priority on input port

but then for the next switch to know the priority too we would need to send to its appropriate port too

so the number of both input and output ports would be multiplied by the number of priority levels!

a better way is to mark the frames



802.1p

the VLAN tag reserves a 3 bit *user priority* field AKA P-bits

P-bits allow marking individual frames with a value 0 ... 7

non-VLAN frames can use priority tagging (VLAN=0)

just to have a user priority field

user priority levels map to traffic classes (CoS)

traffic class indicates drop probability, latency across the switch, etc.
but there are no BW/latency/jitter guarantees

P=0 means non-expedited traffic

802.1Q recommends mappings from P-bits to traffic class

see later for RPR traffic classes and priority

(MEF) Service attributes

all per EVC, per CoS

■ frame loss

fraction of frames that should be delivered that actually are delivered
specified by T (time interval) and L (loss objective)

■ frame delay

measured UNI-N to UNI-N on delivered frames
specified by T, P (percentage) and D (delay objective)

■ frame delay variation

specified by T, P, L (difference in arrival times), V (FDV objective)

■ BW profiles

per EVC, per CoS, per UNI
specified by CIR, CBS, EIR, EBS, ...

Burst size token buckets

the profile is enforced in the following way

there are two byte buckets, C of size CBS and E of size EBS

tokens are added to the buckets at rate $CIR/8$ and $EIR/8$

when bucket overflows tokens are lost (*use it or lose it*)

if ingress frame length < number of tokens in C bucket

frame is **green** and its length in tokens is debited from C bucket

else if ingress frame length < number of tokens in E bucket

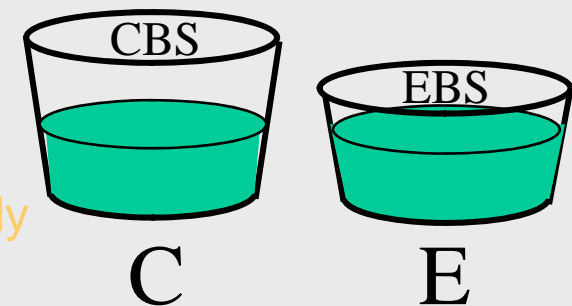
frame is **yellow** and its length of tokens is debited from E bucket

else frame is **red**

green frames are delivered and service objectives apply

yellow frames are delivered by service objectives don't apply

red frames are discarded



Hierarchical BW profiles

MEF 10.1 allows bandwidth profile

- per UNI (can be different at different UNIs of same multipoint EVC)
- per EVC *and* CoS

but doesn't allow a single frame to be subject to more than 1 profile

New work in the MEF is aimed at allowing

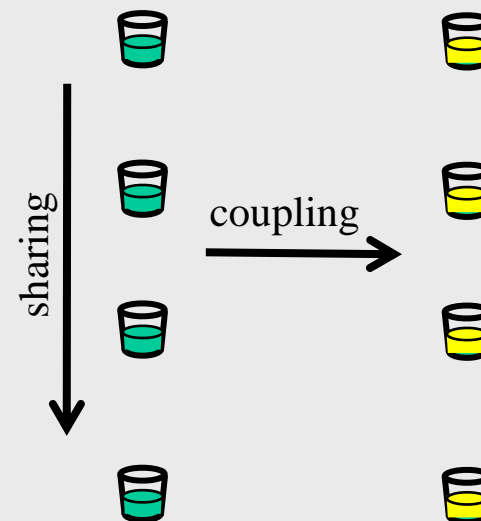
- per CoS bandwidth profile, followed by
- per EVC color-aware profile

The idea is to allow the user to use excess “paid for” bandwidth for lower priority traffic (*BW sharing*)

Thus

- frames will never be downgraded (green → yellow, or yellow → red)
- frames may be upgraded (red → yellow, yellow → green)

There are complex inter-relationships between sharing and coupling



Link aggregation

Link aggregation (ex 802.3ad, 802.3 clause 43)
Link Aggregation Control Protocol (LACP)
conversations

Link Aggregation task force

Ethernet needed “link aggregation”

(AKA bonding, Ethernet trunk, inverse mux, NIC teaming, etc)

enables bonding several ports together as single uplink

- increased uplink BW *can exceed 1 Gb/s (or now 10Gb/s)*
- common practice to install multiple fibers/cables
- incrementally grow uplink capacity
 - needn't purchase new expensive Gb switch when exceed 100 Mb
- increased availability *continue at reduced rate when 1 link fails*
- needn't constrain peak capacity of individual ports

Link Aggregation 802.3ad task force

PAR approved June 1998

WG ballot Nov 1999

LMSC ballot Nov 1999

standard Feb 2000

folded into 802.3-2000 as clause 43

Alternative inverse MUX defined by EFM task force (EFM bonding)

Link aggregation in action

Link Aggregation Group controlled by aggregator

aggregation port looks standard Ethernet MAC (no SN, etc.)

MAC address may be one that of one of the links

only for p2p full duplex operation

the aggregator = frame distributor + frame collector + parser + multiplexer

- distributes frames to links making up LAG
- collects frames from LAG and passes to clients

link aggregation should not cause misordering, replication, etc.

binding of ports to LAGs distributed via Link Aggregation Control Protocol

LACP uses slow protocol frames

links may be dynamically added/removed from LAG

optional marker protocol provides sequence markers

LA *conversations*

frame distributor assigns all frames from a *conversation* to one link

a conversation is defined as frames with same:

- SA
- DA
- reception port
- protocol (Ethertype)
- higher layer protocol (LC info)

hash on above maps to port

before moving conversation to a different link,
ensure that all transmitted frames have been received (marker protocol)

LACP continuously monitors to detect if changes needed

Ethernet protection

Linear protection

Ring protection

APS

Automatic Protection Switching (APS)

is a functionality of carrier-grade transport networks

is often called resilience

since it enables service to quickly recover from failures

is required to ensure high reliability and availability

APS includes :

- detection of *failures* (signal fail or signal degrade) on a *working channel*
- switching traffic *transmission* to a *protection channel*
- selecting traffic *reception* from the protection channel
- (optionally) reverting back to the working channel once failure is repaired

Using STP and LAG

STP and **RSTP** automatically converge to a loop-free topology

RSTP converges in about the same time as STP

but can reconverge after a topology change in less than 1 second

Thus RSTP can be used as a protection mechanism

However, the switching time will be many tens of ms to 100s of ms

LAG also detects failures (using physical layer or LACP)

and automatically removes failed links

Thus LAG too can be used as a primitive protection mechanism

When used this way it is called *worker/standby* or *N+N mode*

The restoration time will be on the order of 1 second

G.8031

Q9 of SG15 in the ITU-T is responsible for protection switching

In 2006 it produced G.8031 Linear Ethernet Protection Switching

G.8031 uses standard Ethernet formats, but is incompatible with STP

The standard addresses

- point-to-point VLAN connections
- SNC (local) protection class
- 1+1 and 1:1 protection types
- unidirectional and bidirectional switching for 1+1
- bidirectional switching for 1:1
- revertive and nonrevertive modes
- 1-phase signaling protocol

G.8031 uses Y.1731 OAM CCM messages in order to detect failures

G.8031 defines a new OAM opcode (39) for APS signaling messages

Switching times should be under 50 ms (only holdoff timers when groups)

G.8031 signaling

The APS signaling message looks like this :

MEL (3b)	VER=0 (5b)	OPCODE=39 (1B)	FLAGS=0 (1B)	OFFSET=4 (1B)
req/state (4b)	prot. type (4b)	requested sig (1B)	bridged sig (1B)	reserved (1B)
END=0 (1B)				

- regular APS messages are sent 1 per 5 seconds
- after change 3 messages are sent at max rate (300 per sec)

where

- req/state identifies the message (NR, SF, WTR, SD, forced switch, etc)
- prot. type identifies the protection type (1+1, 1:1, uni/bidirectional, etc.)
- requested and bridged signal identify incoming / outgoing traffic since only 1+1 and 1:1 they are either null or traffic (all other values reserved)

G.8031 1:1 revertive operation

In the normal (NR) state :

- head-end and tail-end exchange CCM (at 300 per second rate) on both working and protection channels
- head-end and tail-end exchange NR APS messages on the protection channel (every 5 seconds)

When a failure appears in the working channel

- tail-end stops receiving 3 CCM messages on working channel
- tail-end enters SF state
- tail-end sends 3 SF messages at 300 per second on the APS channel
- tail-end switches selector (bi-d and bridge) to the protection channel
- head-end (receiving SF) switches bridge (bi-d and selector) to protection channel
- tail-end continues sending SF messages every 5 seconds
- head-end sends NR messages but with bridged=normal

When the failure is cleared

- tail-end leaves SF state and enters WTR state (typically 5 minutes, 5..12 min)
- tail-end sends WTR message to head-end (in nonrevertive - DNR message)
- tail-end sends WTR every 5 seconds
- when WTR expires both sides enter NR state

Ethernet rings ?

Ethernet has become carrier grade :

- deterministic connection-oriented forwarding
- OAM
- synchronization

The only thing missing to completely replace SDH is ring protection

However, Ethernet and ring architectures don't go together

- Ethernet has no TTL, so looped traffic will loop forever
- STP builds trees out of any architecture – no loops allowed

There are two ways to make an Ethernet ring

- open loop
 - cut the ring by blocking some link
 - when protection is required - *block the failed link*
- closed loop
 - disable STP (but avoid infinite loops in some way !)
 - when protection is required - *steer* and/or *wrap* traffic

Ethernet ring protocols

Open loop methods

- G.8032 (ERPS)
- rSTP (ex 802.1w)
- RFER (RAD)
- ERP (NSN)
- RRST (based on RSTP)
- REP (Cisco)
- RRSTP (Alcatel)
- RRPP (Huawei)
- EAPS (Extreme, RFC 3619)
- EPSR (Allied Telesis)
- PSR (Overture)

Closed loop methods

- RPR (IEEE 802.17)
- CLEER and NERT (RAD)

G.8032

Q9 of SG15 produced G.8032 between 2006 and 2008

G.8032 is similar to G.8031

- strives for 50 ms protection (< 1200 km, < 16 nodes)
 - but here this number is deceiving as MAC table is flushed
- standard Ethernet format but incompatible with STP
- uses Y.1731 CCM for failure detection
- employs Y.1731 extension for R-APS signaling (opcode=40)
- R-APS message format similar to APS of G.8031
(but between every 2 nodes and to MAC address 01-19-A7-00-00-01)
- revertive and nonrevertive operation defined

However, G.8032 is more complex due to

- requirement to avoid loop creation under any circumstances
- need to localize failures
- need to maintain consistency between all nodes on ring
- existence of a special node (RPL owner)

RPL

G.8032 defines the **R**ing **P**rotection **L**ink (RPL)
as the link to be blocked (to avoid closing the loop) in NR state

One of the 2 nodes connected to the RPL
is designated the *RPL owner*

Unlike RAD's RFER

- there is only one RPL owner
- the RPL and owner are designated before setup
- operation is usually revertive

All ring nodes are simultaneously in 1 of 2 modes – idle or protecting

- in idle mode the RPL is blocked
- in protecting mode the failed link is blocked and RPL is unblocked
- in revertive operation
once the failure is cleared the block link is unblocked
and the RPL is blocked again

G.8032 revertive operation

In the idle state :

- adjacent nodes exchange CCM at 300 per second rate (including over RPL)
- exchange NR RB (RPL Blocked) messages in dedicated VLAN every 5 seconds (but *not* over RPL)
- R-APS messages are never forwarded

When a failure appears between 2 nodes

- node(s) missing CCM messages *peek twice* with holdoff time
- node(s) block failed link and flush MAC table
- node(s) send SF message (3 times @ max rate, then every 5 sec)
- node receiving SF message will check priority and unblock any blocked link
- node receiving SF message will send SF message to its other neighbor
- in stable protecting state SF messages over every unblocked link

When the failure is cleared

- node(s) detect CCM and start guard timer (blocks acting on R-APS messages)
- node(s) send NR messages to neighbors (3 times @ max rate, then every 5 sec)
- RPL owner receiving NR starts WTR timer
- when WTR expires RPL owner blocks RPL, flushes table, and sends NR RB
- node receiving NR RB flushes table, unblocks any blocked ports, sends NR RB

G.8032-2010

After coming out with G.8032 in 2008 (*G.8032v1*)

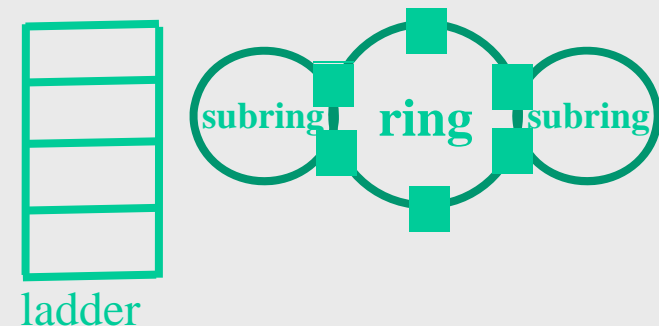
the ITU came out with G.8032-2010 (*G.8032v2*) in 2010

This new version is not *backwards-compatible* with v1

but a v2 node must support v1 as well (but then operation is according to v1)

Major differences :

- 2 designated nodes – *RPL owner* and *RPL node*
- significant changes to
 - state machine
 - priority logic
 - commands (forced/manual/clear) and protocol
- new **Wait To Block** timer
- supports more general topologies (sub-rings)
 - ladders (*For Further Study* in v1)
 - multi-ring
- ring topology discovery
- virtual channel based on VLAN or MAC address



EFM

Ethernet in the First Mile (ex-802.3ah)

EFM bonding

xDSL (2M and 10M)

P2P optics

P2MP optics (EPON)

EFM OAM

EFM task force

in IEEE new works starts with a **Call For Interest**

after which a **Study Group** is formed to consider a new project

SG was formed in Nov. 2000 to think about **Ethernet in the First Mile**

the next step is defining a **Project Authorization Request**

PAR is approved if it passes the **5 criteria**

1. **Broad market potential**
2. **Compatibility**
3. **Distinct identity**
4. **Technical feasibility**
5. **Economic feasibility**

EFM PAR was approved in Sept 2001

a task force or task group in a **WG** works on a specific project

it receives a name of the form **WG.unique_chars**

the EFM task force was called **802.3ah**

EFM task force (cont.)

a task force reviews proposals

and then produces and refines drafts

when complete the draft goes to WG ballot (need 75%)

EFM went to WG ballot in July 2003

after WG ballot, sponsor ballot (75%)

and then LMSC Standards Body approval

EFM became full standard in June 2004

after approval the standard is published

1 year later available free of charge

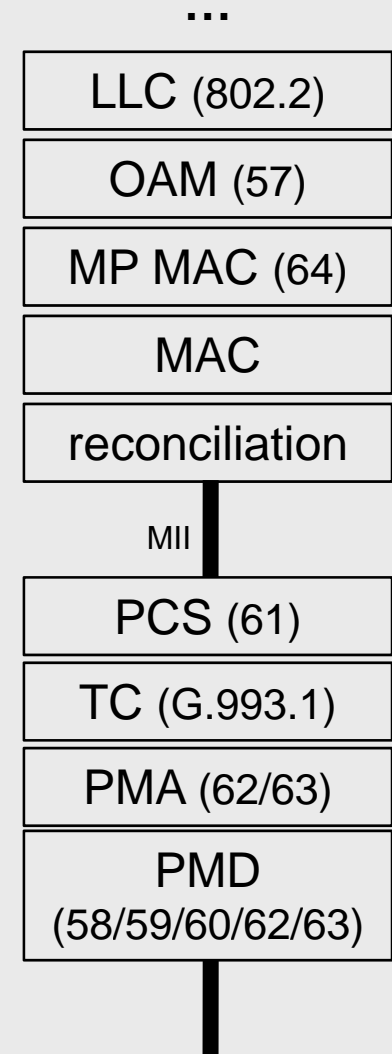
it is usually combined with the WG standard at its next revision

EFM was folded into 802.3-2005

EFM

the 802.3ah task force had 4 tracks:

- Copper (now in clause 61)
 - 10PASS-TS (10M, 750m) (now in clause 62)
 - 2BASE-TL (2M, 2.7km) (now in clause 63)
 - PME aggregation (now in subclause 61.2.2)
- Optics
 - 100M (now in clause 58)
 - 1G (now in clauses 59, 60)
 - EPON (now in clause 65 [see GPON/GEPON course](#))
- P2MP clause 64
 - logic to enable EPON ([see GPON/GEPON course](#))
- OAM (see OAM section below) clause 57



PCS = Physical Coding Sublayer
 TC = Transmission Convergence
 PMA = Physical Medium Attachment
 PMD = Physical Medium Dependent

EFM bonding (PME Aggregation)

Physical Medium Entity inverse MUX (*EFM bonding*)

Defined in 802.3-2005 61.2.2 and ITU-T G.998.2 (ITU-T allows for ADSL as well)

Optional feature applicable **only** to copper EFM (DSL) links

PME Aggregation Function (PAF) is part of PCS (between MII and TC)

- fragments Ethernet frame
- forms non-Ethernet fragments (16b header + fragment)

Divides fragments over physical links (mechanism is implementation dependent)

Uses 14-bit sequence number to recover order of fragments

Fragments can be from 64 to 512 bytes in length (all multiple of 4 except EOP)

Fragments delineated by TC layer framing (64/65 octet coding – **not** HDLC)

Can compensate for

- rate ratios up to 4
- differential delay up to 15,000 bit times
(delay is due to modem interleaving and rate differences)

PME Aggregation (cont.)



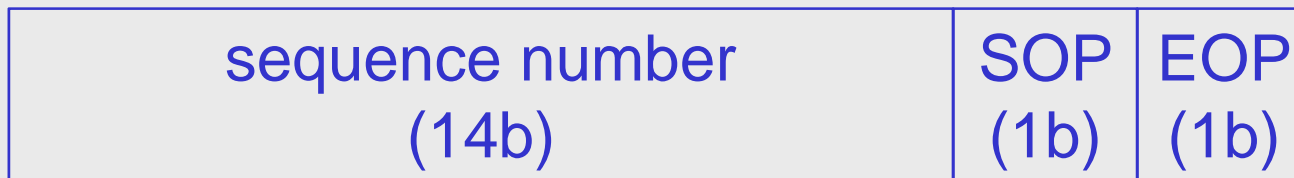
the Ethernet frame from DA SA to FCS is fragmented



mechanism for choosing fragment size and order is *implementation dependent*

CRC is 16 bits for VDSL 32 bits for SHDSL with polynomials defined in 802.3 61.3.3.3

Fragmentation header :



RPR – 802.17

Resilient Packet Rings

resilience and fairness requirements

MAC operation

topology discovery

RPR – 802.17

Resilient Packet Rings

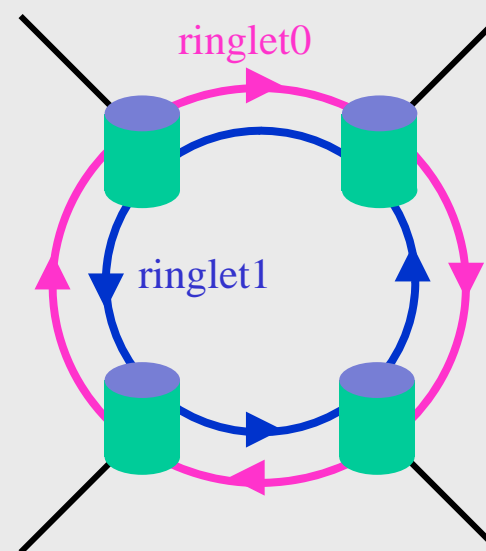
- are compatible with standard Ethernet
- are robust (lossless, <50ms protection, OAM)
- are fair (based on client throttling)
- support QoS (3 classes – A, B, C)
- are efficient (full spatial reuse)
- are plug and play (automatic station autodiscovery)
- extend use of existing fiber rings

counter-rotating add/drop ringlets, running

- SONET/SDH (any rate, PoS, GFP or LAPS) or
- “packetPHY” (1 or 10 Gb/s ETH PHY)

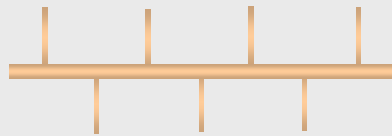
developed by 802.17 WG

based on Cisco’s Spatial Reuse Protocol (RFC 2892)

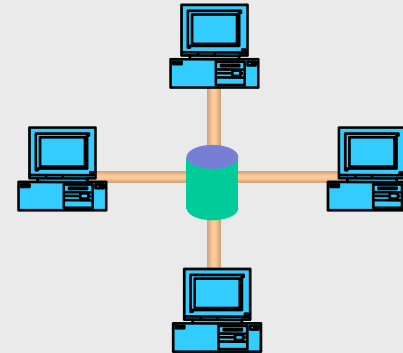


Why rings?

conventional Ethernet topologies are
point-to-point bus



star



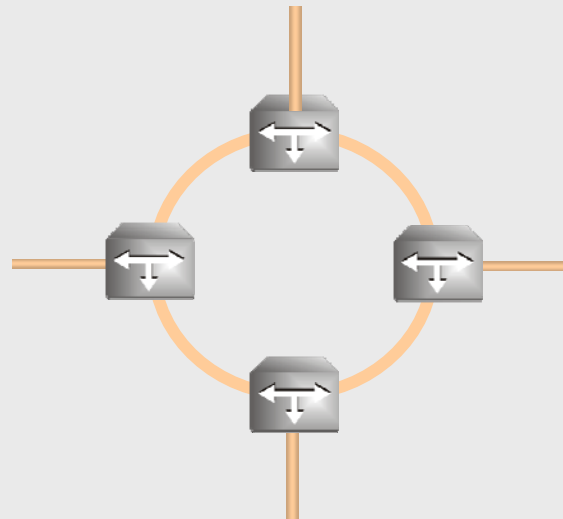
while conventional SONET/SDH topologies are rings

advantages of ring topologies

- protection
- fairness
- simple multicast support

RPR mechanisms

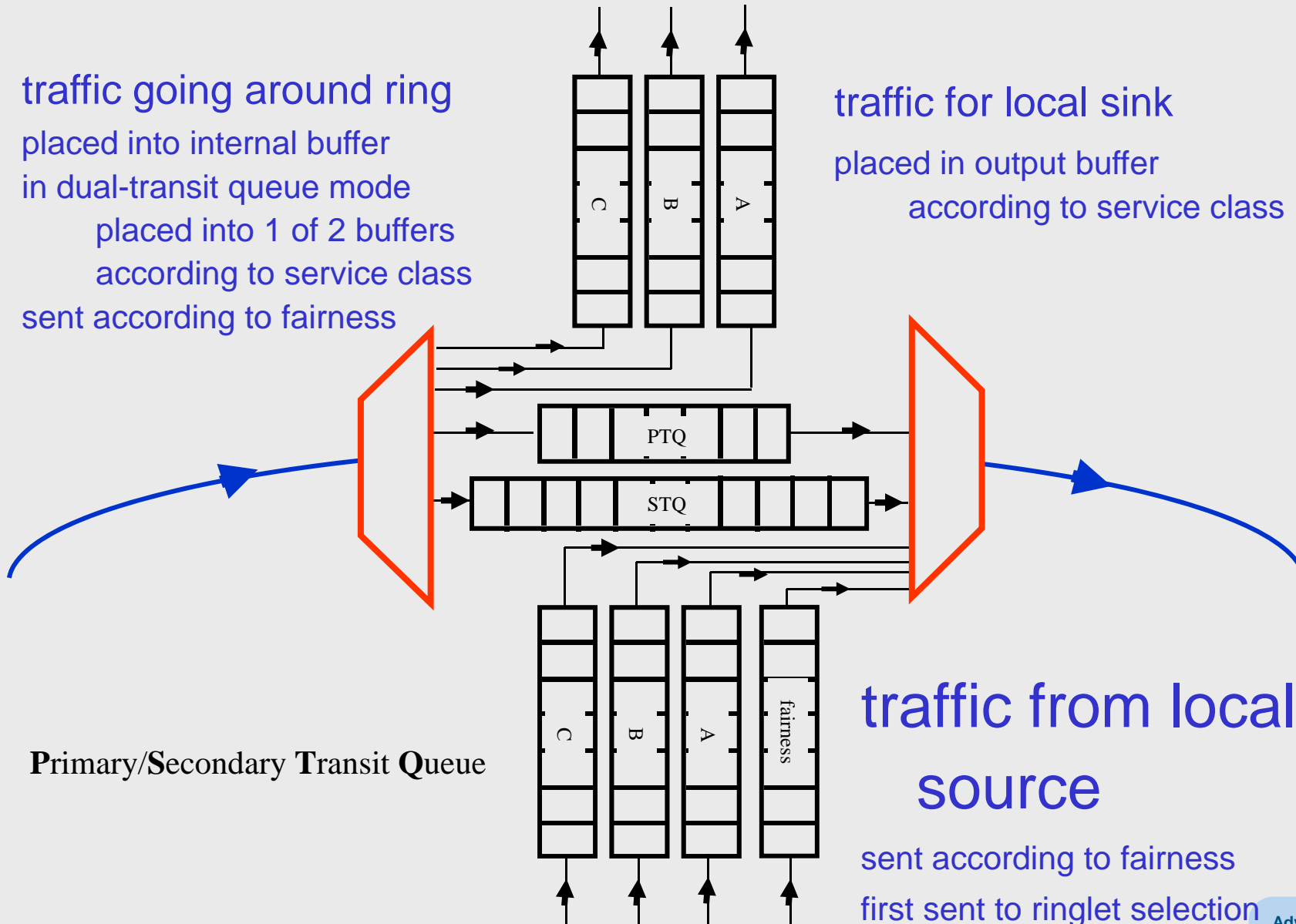
- input shaping
- ringlet selection
- buffer insertion
- transit buffer(s)



Basic queuing

traffic going around ring
 placed into internal buffer
 in dual-transit queue mode
 placed into 1 of 2 buffers
 according to service class
 sent according to fairness

traffic for local sink
 placed in output buffer
 according to service class



Primary/Secondary Transit Queue

traffic from local
 source

sent according to fairness
 first sent to ringlet selection

RPR service classes

RPR defines 3 main classes

- class A : real time (low latency/FDV)
- class B : near real time (bounded predictable latency/FDV)
- class C : best effort

class	use	info rate	D/FDV	FE
A0	RT	reserved	low	No
A1	RT	allocated, reclaimable	low	No
B-CIR	near RT	allocated, reclaimable	bounded	No
B-EIR	near RT	opportunistic	unbounded	Yes
C	BE	opportunistic	unbounded	Yes

Class use

A0 ring BW is reserved – not reclaimed even if no traffic

in dual-transit queue mode:

- class A frames from the ring are queued in PTQ
- class B, C in STQ

priority for egress

- frames in PTQ
- local class A frames
- local class B (when no frames in PTQ)
- frames in STQ
- local class C (when no PTQ, STQ, local A or B)

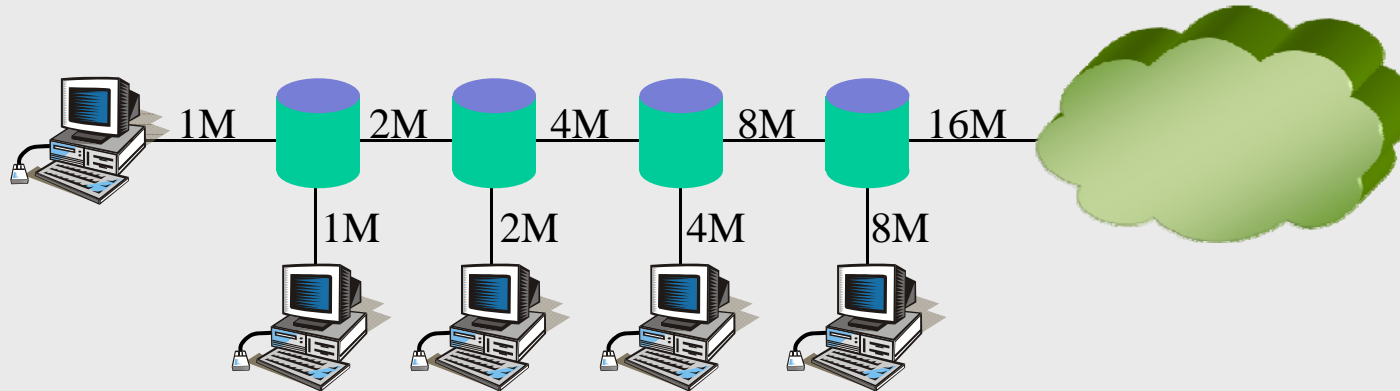
Notes:

class A have minimal delay

class B have higher priority than STQ transit frames, so bounded delay/FDV

classes B and C share STQ, so once in ring have similar delay

RPR - fairness



regular Ethernet is inherently unfair!

RPR never discards packets to resolve congestion
instead uses ingress policing

an upstream node can potentially starve a downstream node

when a downstream node experiences BW starvation
it sends a *fairness request* upstream

upstream node **MUST** reduce its traffic injection rate

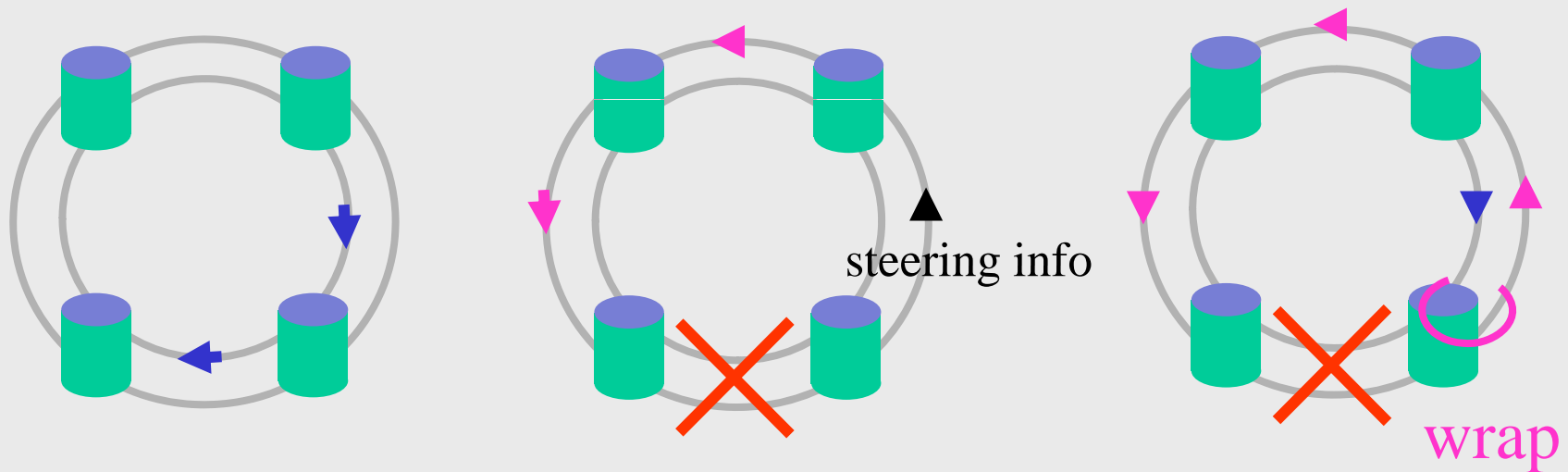
RPR - protection

rings give inherent protection against single point of failure

RPR specifies 2 mechanisms

- steering
- wrapping (optional)

(implementations may also do wrapping then steering)



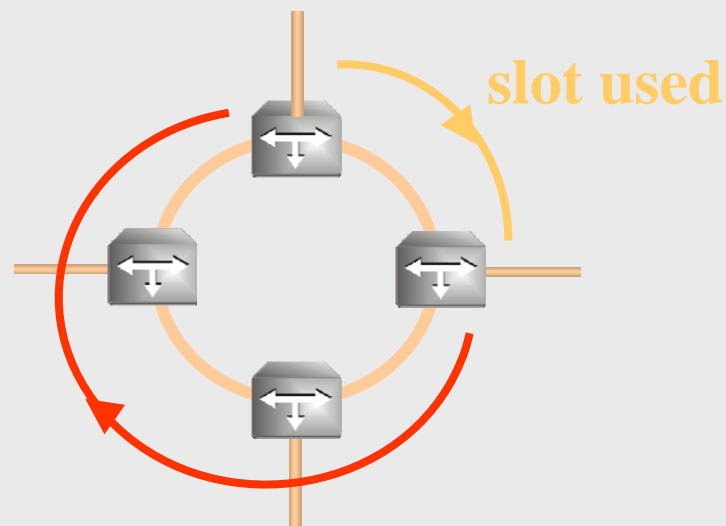
Twice as efficient as EoS ?

it is frequently said that RPR is *twice as efficient as EoS*

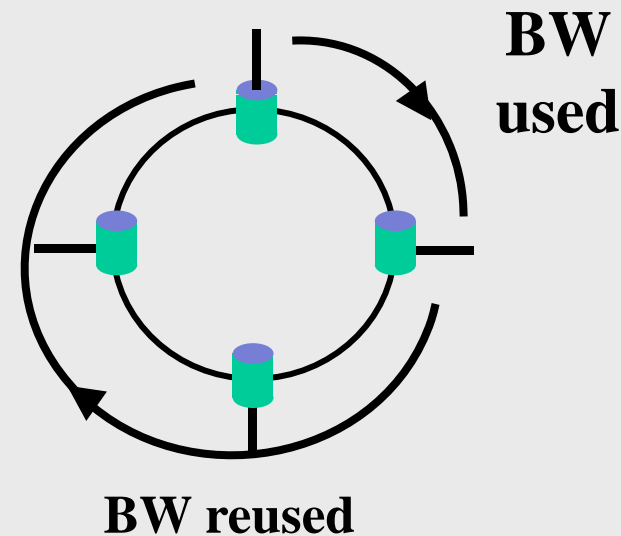
how could that be true?

RPR has inherent spatial re-use

slots in SONET/SDH rings are typically not re-used
due to management complexity



**data remains in slot –
removed by originator**



RPR - multicast

for regular Ethernet multicast requires replicating frames

for RPR, broadcast/flooding/multicast

 simply requires not removing frame from ring

multicast can be unidirectional or bidirectional

when TTL=0 the frame is finally removed

RPR frame formats

802.17 defines 4 frame types:

- data frame
ETH MAC frame + TTL, frame type and flag fields
- control frame
attribute discovery, topology, protection, round-trip measure, OAM, etc.
- fairness frame
sent upstream to indicate required fair rate
- idle frame
sent to neighboring node to avoid PTQ overflow due to lack of sync

Ethernet OAM

OAM functions

link OAM (802.3ah)

service OAM (Y.1731, 802.1ag)

OAM

analog channels and 64 kbps digital channels
did not have mechanisms to check signal validity and quality

thus

- major faults could go undetected for long periods of time
- hard to characterize and localize faults when reported
- minor defects might be unnoticed indefinitely

as PDH networks evolved, more and more overhead was dedicated to
Operations, **A**dministration and **M**aintenance (OAM) functions

including:

- monitoring for valid signal
- defect reporting
- alarm indication/inhibition

when SONET/SDH was designed
overhead was reserved for OAM functions

today service providers require complete OAM solutions

OAM (cont.)

OAM is a *user-plane* function

but may influence control and management plane operations

for example

- OAM may trigger protection switching, but doesn't switch
- OAM may detect provisioned links, but doesn't provision them

OAM is more complex and more critical for PSNs

since in addition to previous problems

- loss of signal
- bit errors

we have new defect types

- packets may be lost
- packets may be delayed
- packets may incorrectly delivered

OAM requirements are different for CO and CL modes

ITU-T concept - Trail

since OAM is critical to proper network functioning

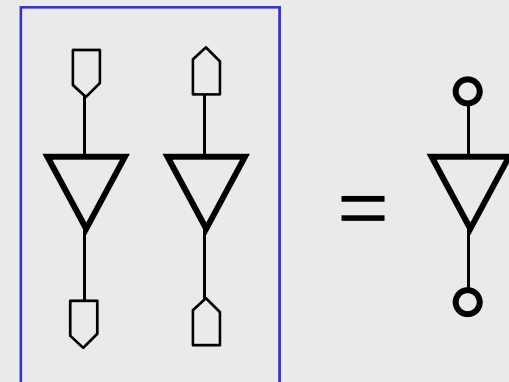
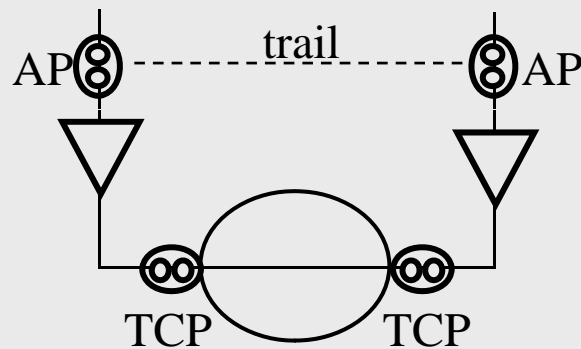
OAM must be added to the concept of a connection

a **trail** is defined as a connection along with integrity supervision

clients gain access to the trail at **access points (AP)**

the trail termination function is responsible for

generating / processing OAM



Trail Termination Functions

what functionality does the trail termination function add ?

- Continuity Check (e.g. LOS, periodic CC packets)
- Connectivity Verification (detect misrouting)
- signal quality monitoring (e.g. error detection coding)
- defect notification/alarm inhibition (AIS(FDI), RDI(BDI))

source termination functions:

- generates error check code (FEC, CRC, etc)
- returns remote indications (REI, RDI)
- inserts trail trace identification information

sink termination functions:

- detects misconnections
- detects loss of signal, loss of framing, AIS instead of signal, etc.
- detects code violations and/or bit errors
- monitors performance

Defects, Faults, etc.

G.806 defines:

- anomaly (n):** smallest observable discrepancy between desired and actual characteristics
- defect (d):** density of anomalies that interrupts some required function
- fault cause (c):** root cause behind multiple defects
- failure (f):** persistent fault cause - ability to perform function is terminated
- action (a):** action requested due to fault cause
- performance parameter (p):** calculatable value representing ability to function

for example:

- dLOS = loss of signal defect
- cPLM = payload mismatch cause
- aAIS = insertion of AIS action

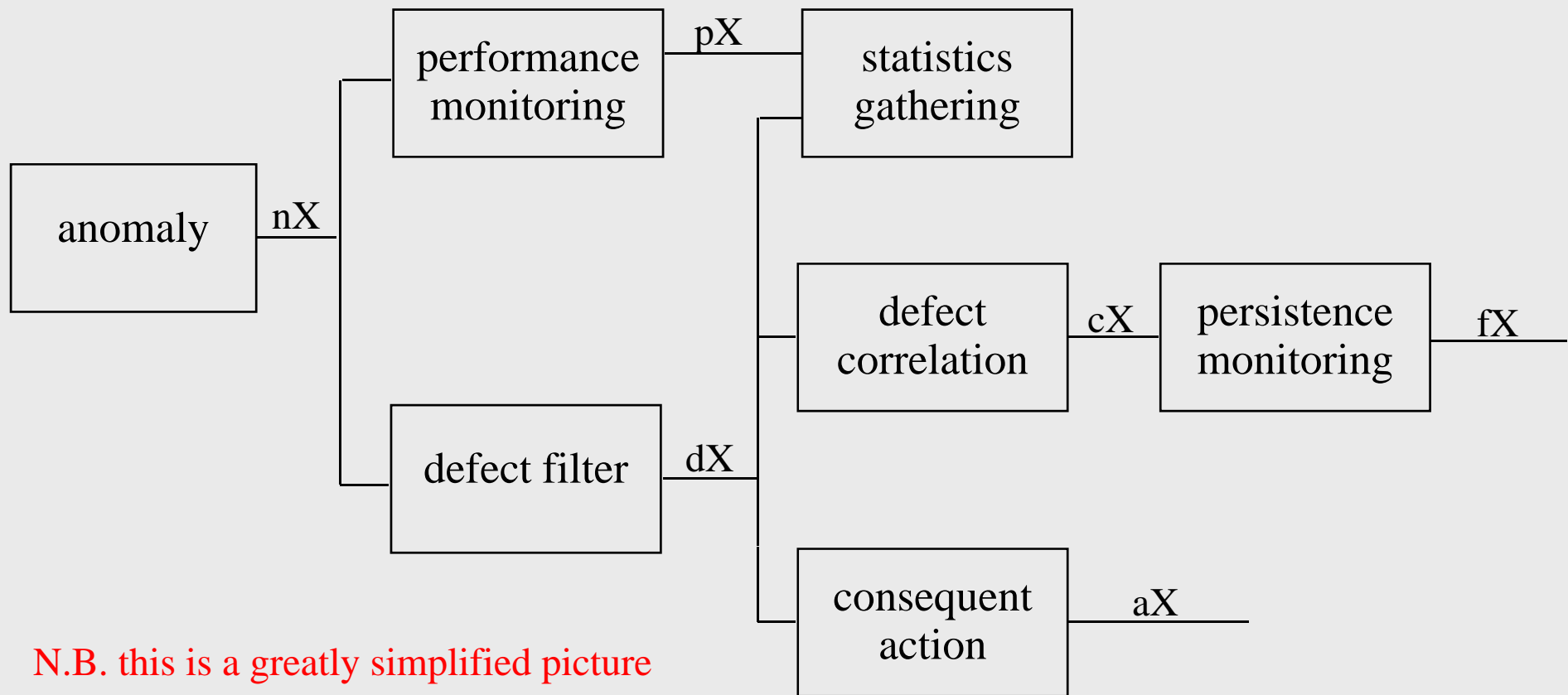
equipment specifications define relationships

e.g.

$aAIS \leq dAIS \text{ or } dLOS \text{ or } dLOF$

alarms are human observable failure indications

Supervision Flowchart



N.B. this is a greatly simplified picture
 more generally there are external
 signals, time constants, etc.

Ethernet OAM functionality

- Continuity Check / Connectivity Verification
- LoopBacks
 - in-service (nonintrusive)
 - out-of service (intrusive)
 - linktrace
- defect notification / alarm inhibition
 - AIS (FDI)
 - RDI (BDI)
- performance monitoring
 - frame loss
 - one-way delay
 - round-trip delay
 - delay variation
 - throughput

Two flavors

for many years there was no OAM for Ethernet
now there are two incompatible ones!

Link layer OAM – EFM 802.3ah 802.3 clause 57
single link only
limited functionality

Service OAM – Y.1731, 802.1ag (CFM)
any network configuration
full OAM functionality

in some cases may need to run both (e.g. ETH over ETY)
while in others only service OAM makes sense



EFM OAM

EFM networks are mostly p2p links or p2mp PONs
thus a link layer OAM is sufficient for EFM applications

Since EFM link is between customer and Service Provider
EFM OAM entities are classified as active (SP) or passive (customer)
active entity can place passive one into LB mode, but not the reverse

but link OAM may be used for any Ethernet link, not just EFM ones

EFM OAMPDUs are a slow protocol frames – not forwarded by bridges

Ethertype = 88-09 and subtype 03

messages multicast to slow protocol specific group address

OAMPDUs must be sent once per second (heartbeat)

messages are TLV-based

DA 01-80-C2- 00-00-02	SA	TYPE 8809	SUB TYPE 03	FLAGS (2B)	CODE (1B)	DATA	CRC
-----------------------------	----	--------------	-------------------	---------------	--------------	------	-----

EFM OAM capabilities

6 codes are defined

- Information (autodiscovery, heartbeat, fault notification)
- Event notification (statistics reporting)
- Variable request (active entity query passive's configuration) (not really OAM)
- Variable response (passive entity responds to query) (not really OAM)
- Loopback control (active entity enable/disable of passive's PHY LB mode)
- Organization specific (proprietary extensions)

flags are in every OAMPDU

expedite notification of critical events

- link fault (RDI)
- dying gasp
- unspecified

monitor slowly degradations in performance

Y.1731 OAM

SPs want to monitor full networks, not just single links

Service layer OAM provides end-to-end integrity
of the Ethernet service over arbitrary server layers

Ethernet is the hardest case for OAM

- connectionless – can't use ATM-like connection continuity check
- MP2MP – so need full connectivity verification
- layering – need separate OAM for operator, SPs, customer
- specific ETH behaviors – flooding, multicast, etc.

Y.1731 messages

Y.1731 supports many OAM message types:

- **Continuity Check** proactive heartbeat with 7 possible rates
- **LoopBack** unicast/multicast pings with optional patterns
- **Link Trace** identify path taken to detect failures and loops
- **AIS** periodically sent when CC fails, useful when no STP
- **RDI**
- **LoCK signal** inform peer entity about intentional diagnostic actions
- **Test signal** in-service/out-of-service tests for loss rate, etc.
- **Automatic Protection Switching**
- **Maintenance Communications Channel** remote maintenance
- **EXPerimental**
- **Vendor SPecific**

Y.1731 frame format

after DA, SA and Ethertype (8902)

Y.1731/802.1ag PDUs have the following header (may be VLAN tagged)

LEVEL (3b)	VER (5b)	OPCODE (1B)	FLAGS (1B)	TLV-OFF (1B)
---------------	-------------	----------------	---------------	-----------------

if there are sequence numbers/timestamp(s), they immediately follow
then come TLVs, the “end TLV”, followed by the CRC

TLVs have 1B type and 2B length fields

there may or not be a value field

the “end-TLV” has type = zero and no length or value fields

MEPs and MIPs

Maintenance Entity (ME) – entity that requires maintenance

ME is a relationship between ME end points

because Ethernet is MP2MP, we need to define a ME Group

MEGs can be nested, but not overlapped

MEG LEVEL takes a value 0 ... 7

by default - 0,1,2 operator, 3,4 SP, 5,6,7 customer

MEP = MEG end point (MEG = ME group, ME = Maintenance Entity)
(in IEEE MEG → MA = Maintenance Association)

unique MEG IDs specify to which MEG we send the OAM message

MEPs responsible for OAM messages not leaking out
but transparently transfer OAM messages of higher level

MIPs = MEG Intermediate Points

- never originate OAM messages,
- process some OAM messages
- transparently transfer others

Ethernet security

Security functions

802.1X

MACsec (802.1AE)

MACkey (802.1af)

Security functions

Some threats that may need to be countered in Ethernet networks

- denial of service (DoS) to all or some stations
- theft of service
- access to confidential information
- modification of information
- control of restricted resources

Some security functions that solve some of these problems

- source authentication
- confidentiality
- data integrity
- replay protection
- non-repudiation
- blocking DoS attacks
- protection against traffic analysis

802.1X

802.1X is a port-based access control mechanism

- it enables or blocks traffic from a port

It provides authentication for devices wishing to communicate

It is based on the **E**xtensible **A**uthentication **P**rotocol (RFC3748)

It is used in 802.11i for WiFi (WPA2)

In 802.1X there are three entities :

- the authenticator
- the supplicant
- the authentication server (usually a RADIUS server)

802.1X PDUs use EtherType 88-8E

and multicast address **01-80-C2-00-00-03**

802.1X operation

Upon detection of a new *supplicant*

- the *authenticator's* switch port is set to *unauthorized*
- only 802.1X traffic is allowed

The authenticator sends *EAP-Request identity* to the supplicant

The supplicant responds with *EAP-response*

The authenticator forwards response to the *authenticating* (RADIUS) server

If the server accepts the request

- the authenticator sets the port to the "authorized" mode
- traffic from supplicant is allowed in

MACsec

802.1AE was approved in June 2006

based on well known AES-128 encryption

but with a new mode - **G**alois **C**ounter **M**ode

Main features

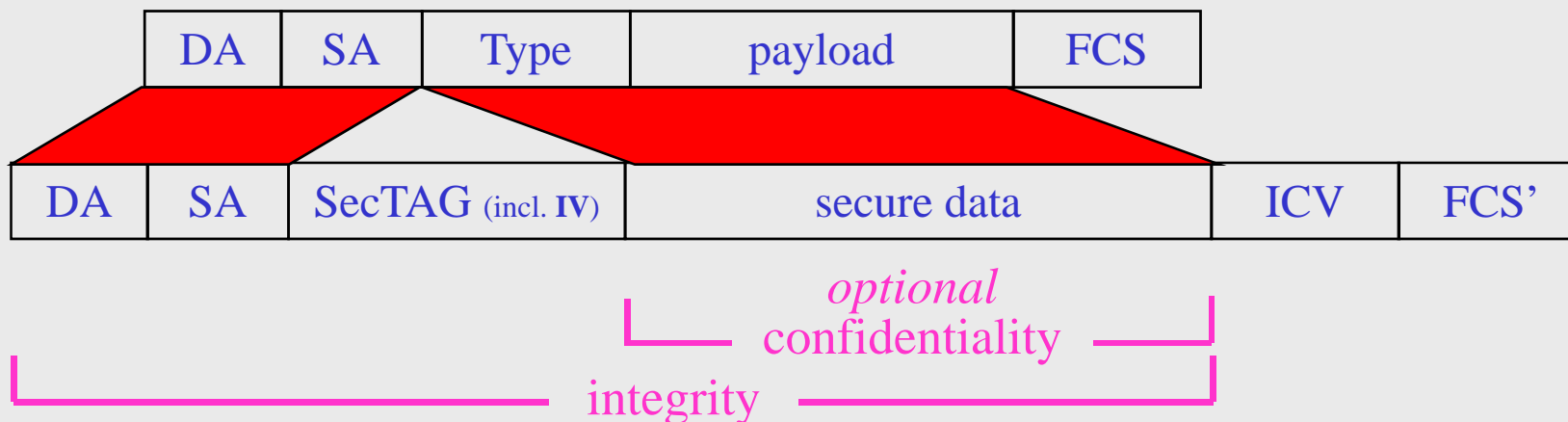
- works over Connectionless network by forming secure associations
- integrated into Ethernet frame format
- key management and association establishment outside scope

802.1AE MACsec provides

- origin authentication
- confidentiality
- connectionless data integrity
- replay protection
- limited blocking of DoS attacks

but may lower some QoS attributes (e.g. introduces bounded delay)

MACsec format



SecTAG contains

- MACsec Ethertype (88E5)
 - 4B Packet Number (sequence number)
 - 8B Secure Channel Identifier
 - ...
- } 12 B Initialization Vector

AES/GCM advantages

- encryption is provided by “state-of-the-art” AES (128/256 bit keys)
- mode of operation uses a *counter* to thwart replay attacks
- **I**ntegrity **C**heck **V**alue verifies the payload integrity
- encryption, integrity, and source authentication by a *single* algorithm
- authentication can be performed without encrypting
- data not in packet payload (e.g. source identifiers) can be authenticated too
- **I**nitialization **V**ector nonce can be any length (but should not repeat for given key)
- algorithm can be efficiently implemented in software
- computation can be parallelized for high speed hardware implementations
- unencumbered by IPR claims

adopted by IEEE 802.1ae for MACsec and RFCs 4106 and 4543 for IPsec

AES/GCM Input / Output

Encryption Input

- plaintext to be encrypted (up to $2^{36}-32$ bytes)
- encryption key (128 or 256 bits)
- per-packet randomly generated IV (12 B recommended)
- additional data to be authenticated but not encrypted (0 .. 2^{61} B)

Encryption Output

- ciphertext (length = length of plaintext)
- ICV (16 B)

Decryption Input

- ciphertext
- encryption key
- IV used for encryption
- ICV generated by encryption

Decryption Output

- Authentication pass/fail
- if pass - plaintext

802.1af

MACsec peers need to share encryption keys
keys need to be regularly updated

MACkey (802.1af) is a key distribution protocol
provides authenticated distribution of keys needed by MACsec

MACkey defines MAC Key Distribution Protocol Data Units MKDPDUs

Authentication based on **E**xtensible **A**uthentication **P**rotocol (RFC 3748)
uses a centrally administered Authentication Server
MACkey defines EAP encapsulation over LANs (EAPOL)

Other 802 security documents

802.1AR - Secure Device Identity

Extends 802.1X specifying unique per-device identifiers (DevID)
Management and binding of a device to its identifiers

802.11i Enhanced WiFi security

WiFi's **W**ired **E**quivalent **P**rivacy encryption broken by Biham (Technion)
802.11i (WPA2) amendment approved June 2004
Uses 802.1X/EAP and AES block cipher

802.10 Enhanced WiFi security

Former 802 security protocol (approved 1998)
Had security associations, key management, confidentiality, integrity
Used by Cisco's Inter Switch Link (ISL) protocol
Withdrawn (Jan 2004) due to lack of use

Synchronous Ethernet

synchronizing networks

packet time protocols

synchronous Ethernet physical layer

Synchronizing networks

SONET/SDH/PDH/TDM networks require highly accurate timing
in every such network there is a primary reference clock
all other clocks derive timing from the PRC
the clock signal is carried in the physical layer

we can say that such networks distribute data + timing

applications needing accurate timing get it “free”

- cellular base stations
- electric power facilities
- data/modem terminals
- TDMoIP GWs
- high quality audio/video systems



PSNs such as Ethernet have no physical layer clock
and thus do not natively distribute timing
how can we support applications needing timing?

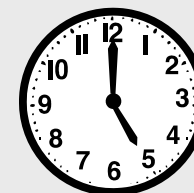
Packet time protocols

TDM networks distribute accurate frequency but not wall-clock
RAD has developed methods for adaptive frequency recovery
for TDMoIP applications

There are packet-based protocols for timing distribution, e.g.,

- IETF NTP (over UDP/IP)
- IEEE 1588 (over UDP/IP or pure Ethernet)

both are based on time servers that send timestamps
usually frequency is acquired first, then ranging, then calibration



NTP and 1588 can be used for frequency distribution
present versions are not sufficiently accurate for all applications
1588 can exploit Transparent Clocks and Boundary Clocks

Synchronous Ethernet (SyncE)

Ethernet started as a CSMA/CD bursty LAN technology
receiver acquired the transmitter clock by locking onto preamble

But for all continuously transmitting full duplex ETY layers
we can lock clocks based on PHY layer

then synchronous Ethernet distributes clock just like TDM networks

This requires modifications to the Ethernet PHY
but no other changes to network

SyncE can co-exist with nonsynchronous Ethernet
and makes IEEE 1588 wall-clock distribution work even better!

ESMC

Synchronous network devices need to identify their clock quality

This is traditionally done using **S**ynchronization **S**tatus **M**essages

G.8264 defines an **E**thernet **S**ynchronization **M**essaging **C**hannel

ESMC frames are slow protocol frames

- with the ITU's OUI (0x0019A7)
- and a new slow protocol subtype 0x0A

The frames carry a 4-bit SSM code

This code can contain any of the usual SSM values (PRC, SSU, SEC, ...)
or one of 2 new values defined for SyncE interfaces