Ethernet

Lecture 5

- Coax, hubs, switches
- Broadcast & multicast
- New additions CGE, DCE

Ethernet reminder

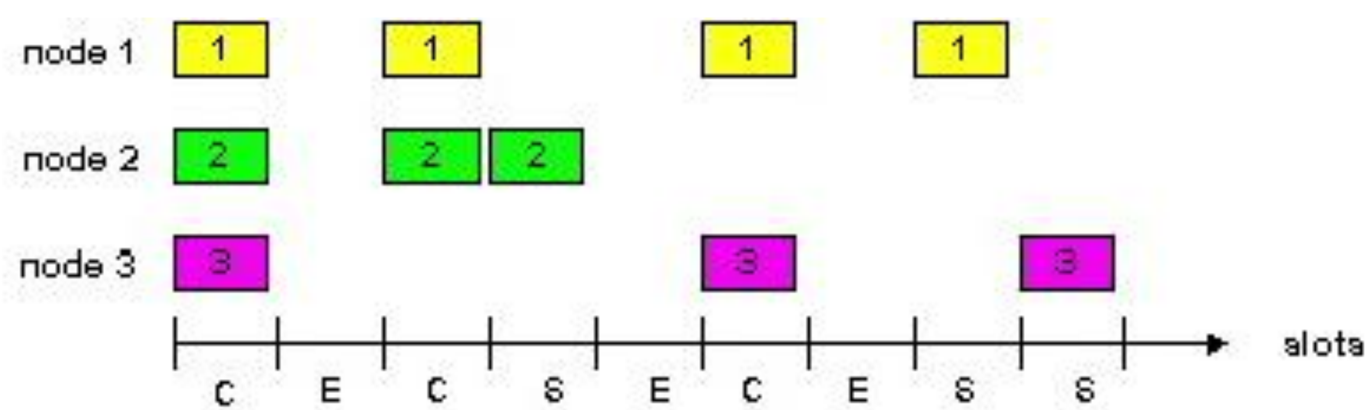
- 10Base5,2 are “broadcast networks”
- 10BaseT with hubs were the same, but just simpler “start” wiring
- CSMA-CD protocol used
  - Sense carrier
  - Start transmit if no carrier
  - Detect collision if more than one starts at once
  - Binary exponential backoff

Aloha – Basic Technique

- First random MAC developed
  - For radio-based communication in Hawaii (1970)
- Basic idea:
  - When you're ready, transmit
  - Receiver's send ACK for data
  - Detect collisions by timing out for ACK
  - Recover from collision by trying after random delay
    - Too short → large number of collisions
    - Too long → underutilization

Slotted Aloha

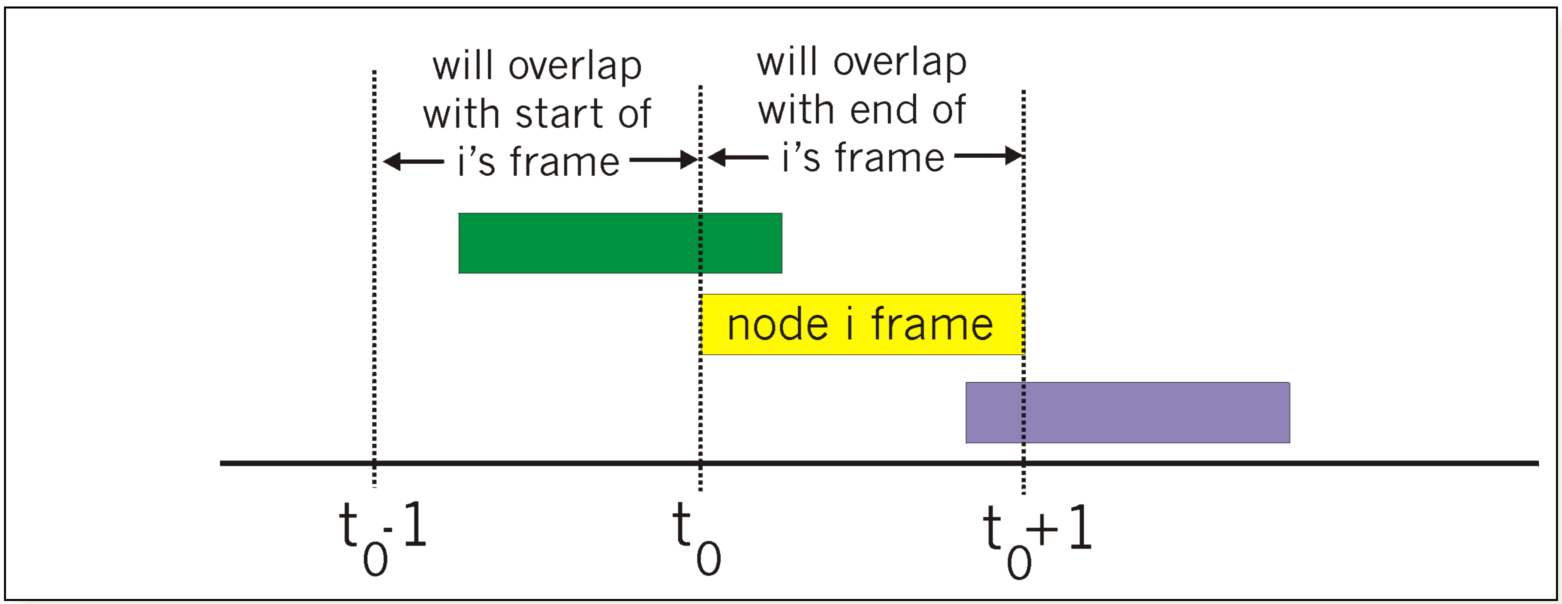
- Time is divided into equal size slots (= pkt trans. time)
- Node (w/ packet) transmits at beginning of next slot
- If collision: retransmit pkt in future slots with probability p, until successful



Success (S), Collision (C), Empty (E) slots

Pure (Unslotted) ALOHA

- Unslotted Aloha: simpler, no synchronization
- Pkt needs transmission:
  - Send without awaiting for beginning of slot
- Collision probability increases:
  - Pkt sent at t_0 collide with other pkts sent in $[t_0-1, t_0+1]$



Slotted Aloha Efficiency

Q: What is max fraction slots successful?

A: Suppose N stations have packets to send

– Each transmits in slot with probability *p*

– Prob. successful transmission *S* is:

by single node: $S = p\,(1-p)^{(N-1)}$

by any of N nodes

S = Prob (only one transmits)

$= N\,p\,(1-p)^{(N-1)}$

To be stable p ≤ 1/N so…

$= 1/e = .37$ *as* N -> *infty*

At best: channel use for useful transmissions 37% of time!

Pure Aloha (cont.)

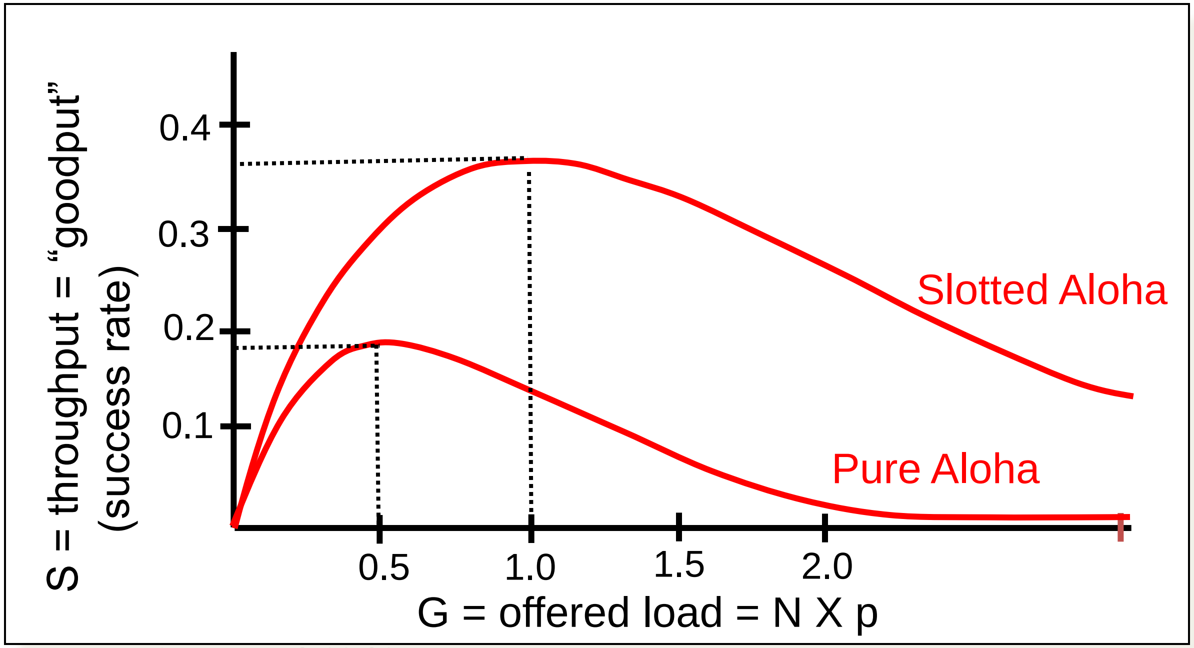
P(success by given node) = P(node transmits) X P(no other node transmits in $[p_0\text{-}1, p_0]$ X P(no other node transmits in $[p_0\text{-}1, p_0]$

$= p \text{ X } (1\text{-}p)^{(N\text{-}1)} \text{ X } (1\text{-}p)^{(N\text{-}1)}$

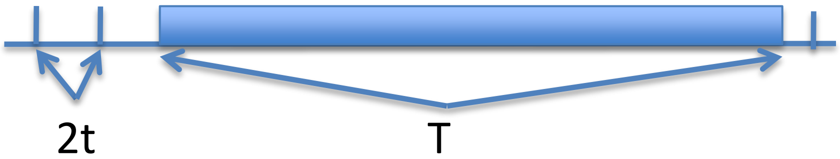
P(success by any of N nodes) = $N\, p \text{ X } (1\text{-}p)^{(N\text{-}1)} \text{ X } (1\text{-}p)^{(N\text{-}1)} = 1/(2e) = .18$

… choosing optimum p as N → infty → p = 1/2N …

protocol constrains effective channel throughput!



CSMA/CD



- View CSMA/CD as Aloha system with slot size “2t” which then reserves the channel for many slots…
- 2t is “the collision window” – time for packet to propogate all the way across the network and someone else to just start sending as it arrives – when do I see the collision?
- Interval between sucessful packets

$$I = (e\text{-}1)2t + T + t$$

= Delay to acquire mini-slot
+ Data transmission delay
+ Wait for next mini-slot

Efficiency

- Efficiency of CSMA/CD

$$E = T/\,I \qquad ;$$
$$= T\,/\,(T + t(2e\text{-}1))$$
$$\approx 1/(1 + 4.4\gamma), \text{ where } \gamma = t/T$$

- Say 10Mbps network with 500m diameter
  - T = 1000 bits = 10^{-4}s
  - t = 1000m/signal speed in Cu= $5\text{x}10^{-6}$s

$$E \approx 82\%$$

  - What happens at 100Mbps, 1Gbps?

Bridges to switches

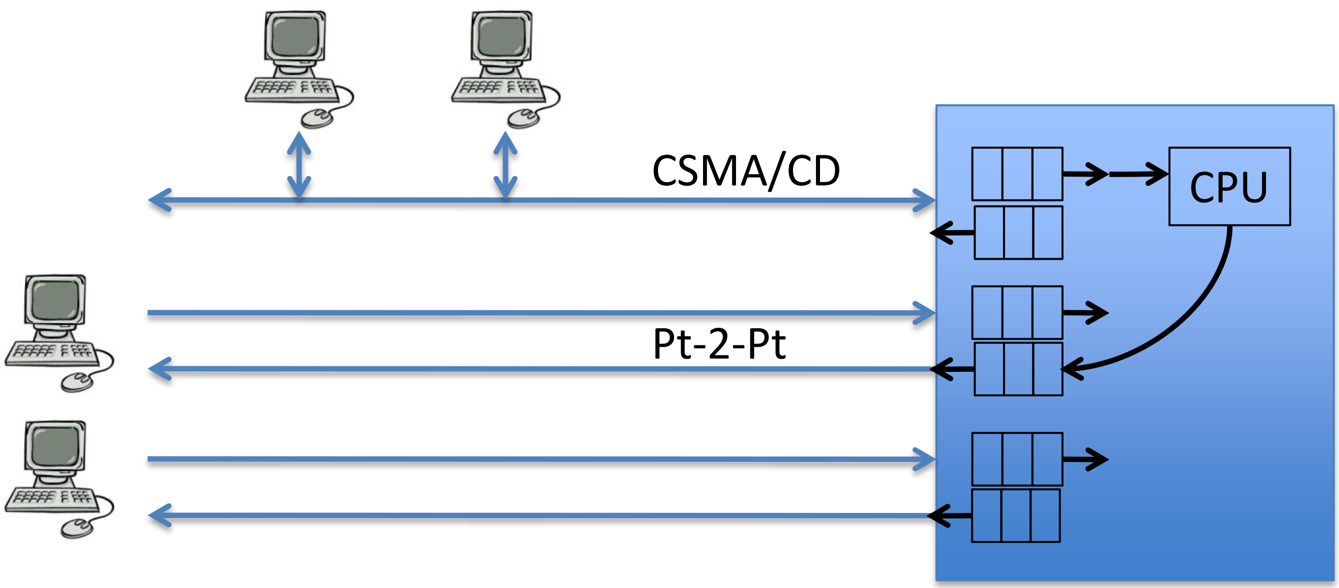
- CSMA/CD works well for certain packet sizes / physical networksize / speed trade-offs
- 100Mbps for normal network traffic profile started to stress CSMA/CD in larger LAN deployments
- Needed to move away from pure CSMA/CD
- Started with “bridges” to partition CSMA/CD into smaller segments – bridges originally connected two CSMA/CD segments…

… then to full blown switching

Buffering…

- Attach CSMA/CD segment to switch, or
- Use point to point links from host to switch
- Buffer incoming packets in simple FIFO queues in switch
- “Contention” resolved by scheduling packets to output queues
- Simple queuing in software possible at low speeds

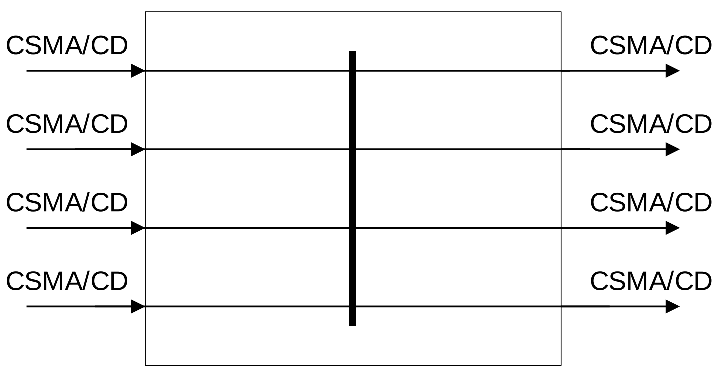
e.g.:



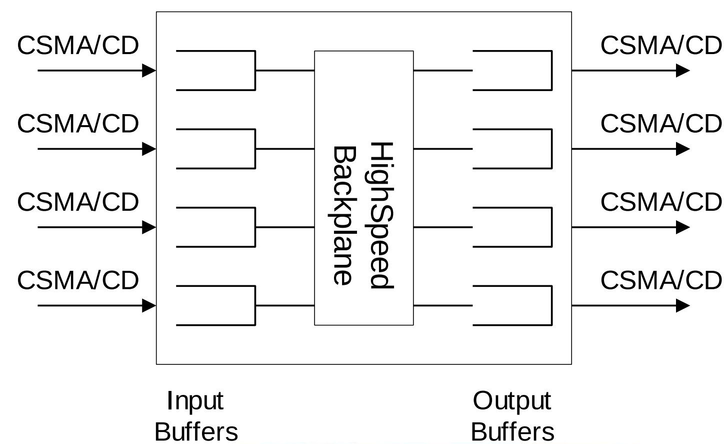
Ethernet Hubs vs. Ethernet Switches

- An **Ethernet switch** is a packet switch for Ethernet frames
    - Buffering of frames prevents collisions.
    - Each port is isolated and builds its own collision domain
- An **Ethernet Hub** does not perform buffering:
    - Collisions occur if two frames arrive at the same time.

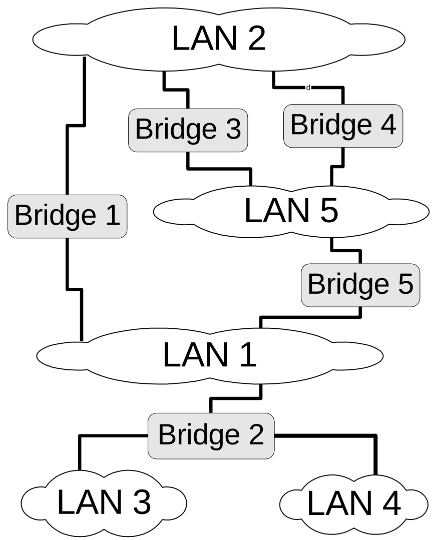
Hub



Switch



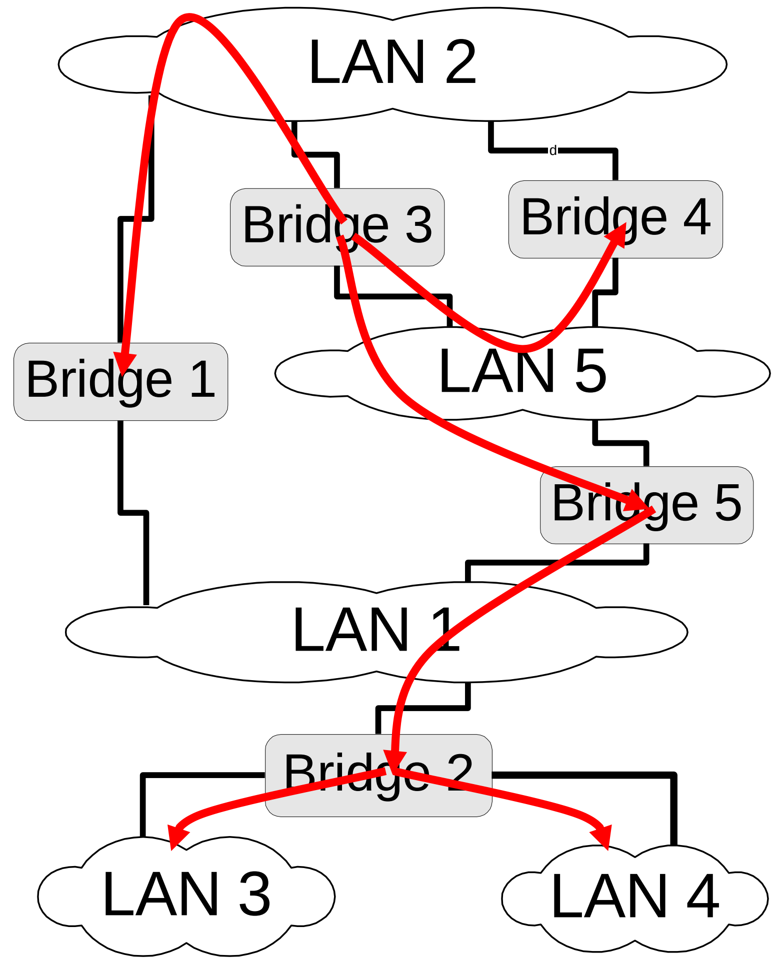
Issues



- Introducing switching introduces multipath:
  - Fools will wire switches with loops by accident
  - Smart people will wire with loops to increase redundancy
- So:
  - Where do we forward the packets
  - How to support broadcast and multicast
- Can we flood incoming packet onto all outgoing links?
  - Packets circulate forever unless
    - Add a hopcount?
    - Maybe by remembering them all
- Really need another solution – Spanning Tree

Spanning Tree Protocol (IEEE 802.1d)

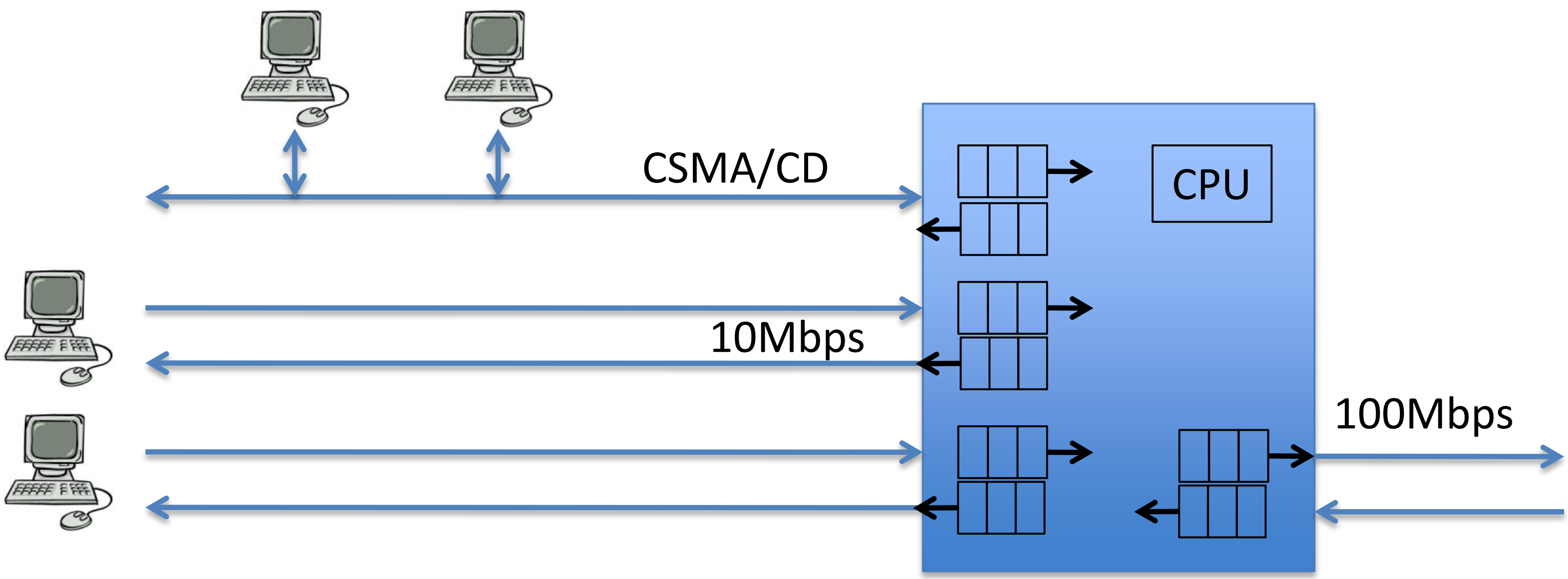
- The Spanning Tree Protocol (SPT) is a solution to prevent loops when forwarding frames between LANs
- The SPT is standardized as the IEEE 802.1d protocol
- The SPT organizes bridges and LANs as spanning tree in a dynamic environment
  - Frames are forwarded only along the branches of the spanning tree
  - Note: Trees don't have loops
- Bridges that run the SPT are called transparent bridges
- Bridges exchange messages to build the tree.



Decoupling rates…

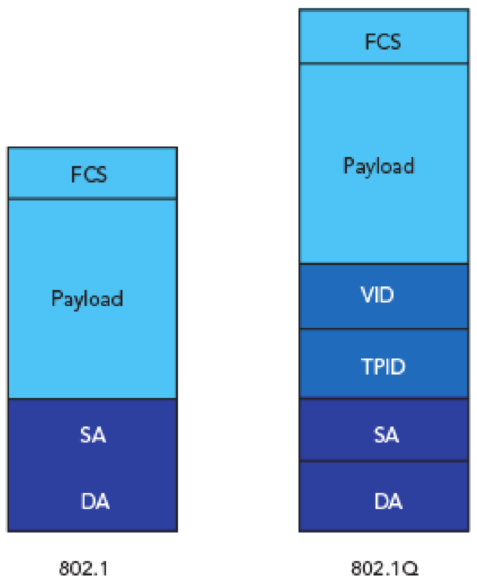
- Once we have introduced buffering we have decoupled the rates of transmission on the different switch ports
- Very common to see hosts atatched at one speed and faster “uplinks” from switch to backbone network.

- e.g.:



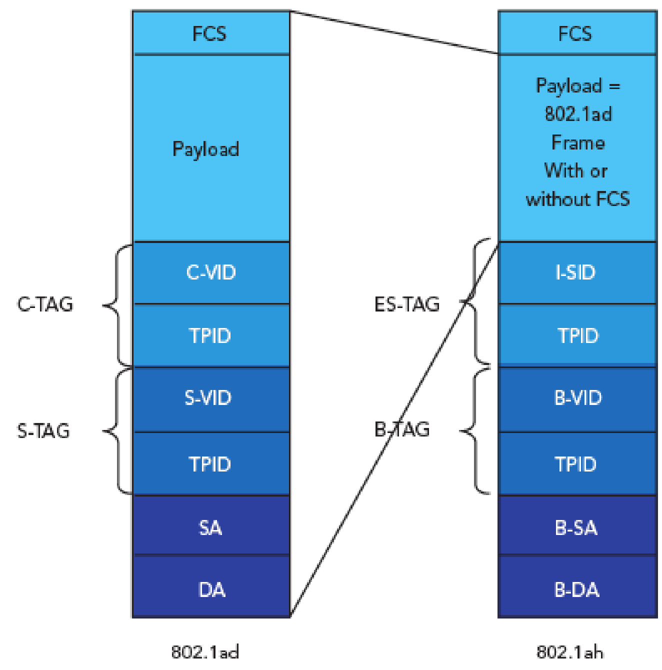
802.1Q VLAN

- VLAN – Aim to support multiple virtual LANs on a single physical infrastructure
- Add VLAN tag that allows up to 64K virtual networks
- Label each spanning tree and forwarding table with VLAN tag
- Works on all 802 networks including 802.11
  - e.g. UoN wireless from one set of base stations…



Carrier Grade Ethernet

- Provide Ethernet service to customers who might already be using VLANs
- Desire to use Ethernet switching in the core network
- First try VLAN in VLAN - 802.1ad
  – Then find customers already using VLAN in VLAN
- So do full 802.1ad encapsultauion in another 802.1ad frame – 802.1ah



Data Centre Ethernet

- Desire to replace 3 networks with one:
  - Storage (e.g. Fibrechannel)
  - InterProcess Communications (e.g. Myrinet)
  - LAN (e.g. Ethernet)

- Need some isolation between traffic types, maybe…
  - Ethernet with QoS
  - 802.1p