

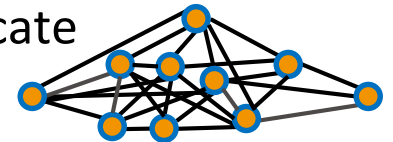
Communications Networks Review

Why digital communications *networks*?

Early telegraph and telephone *connections* were individual *links*



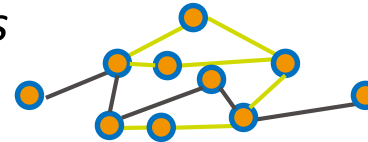
However, it is impossible (or at least very inefficient)
to directly connect every 2 entities that need to communicate



Instead, one builds a **network**

Networks are arbitrary connected *graphs*

- nodes are called **Network Elements**
- edges are **links**



End-points (customers) are nodes, and are called *peers* or *hosts*

Nodes that are not end-points include:

- *switches*
- *routers*
- *gateways* (between different networks)
- *middleboxes*

Network services

Theodore Vail (father of the telephone *network*)
organized telephony as a **service** (like the *postal service*!)

Vailism is the philosophy that public services
should be run as closed centralized monopolies for the public good

In the *Bell-Watson model* a customer pays for equipment
and is responsible for installation, operations, maintenance

In the *Vail model* the customer pays a monthly fee
and the **Service Provider** assumes responsibility for everything

Today basic communications services are available free of charge
but one pays for **Quality of Service** guarantees

Mechanisms to measure and monitor QoS are called
Operations, Administration, and Maintenance

A common mechanism to ensure availability is called
Automatic Protection Switching

QoE and QoS

Customers are willing to pay for the quality of the service received

This should ideally be the **Quality of Experience**

the acceptability of a service as perceived subjectively by the end-user

However, it is difficult to directly measure QoE
and QoE depends on application

So instead, we use **Quality of Service**

objective network parameters that are easily measured

as a proxy for QoE

Service Providers and customers agree on a minimum level of QoS
in a **Service Level Agreement**
that typically include financial penalties for *breaches*

QoS parameters in SLAs include:

- Connectivity parameters (e.g., availability, time to repair)
- Performance parameters (e.g., **P**acket **L**oss **R**atio, data-rate, delay, PDV)

OAM

QoS monitoring is carried out using

Operations, **A**dministration, **M**aintenance (**OAM**)

The difference between connectivity and performance parameters leads to two types of OAM :

- **Fault Monitoring** required for maintenance of connectivity (availability)
 - Continuity Check** checks that data arrives at intended destination
 - Connectivity Verification** checks that it doesn't go to unintended destination
- **Performance Monitoring** required for maintenance of all other QoS parameters
 - 1-way delay measurement,
 - roundtrip delay measurement
 - token bucket parameters measurement

OAM is a ***user-plane*** function

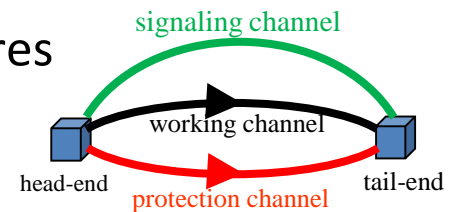
but may influence control and management plane operations

for example

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

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
- exists in 1+1, 1:1, 1:n, m:n, and $(1:1)^n$ flavors



APS includes :

- detection of *failures* (signal fail or signal degrade) on a *working channel* via *OAM – CC messages*
- switching traffic *transmission* to a *protection channel* (except in 1+1)
- selecting traffic *reception* from the protection channel
- (optionally) reverting back to the working channel once failure is repaired

Rings are often used for APS since they provide 2 paths to destination and may be used with 1+1 or 1:1 protection

An alternative to APS is **Fast ReRoute** (defined for MPLS and IP) which provides local detour instead of end-to-end switching

QoE = f(QoS)

One would ideally pay for **Quality of Experience**, e.g.,

- **M**ean **O**pinion **S**core (MOS) between 1 and 5 for voice
- Apdex between 0% and 100% for web applications

QoE may be measured directly using human subjects

but many objective measures have been found

mostly based on psycho-physical models (PSQM, PESQ, POLQA, PEVQ)

QoE for a given application (voice, video, browsing)

is a function of QoS parameters

$$\text{QoE} = f(\text{service}; \text{QoS}_1, \text{QoS}_2, \dots \text{QoS}_n)$$

Researchers have found various functional forms

for the dependence of QoE on a particular QoS parameter

- VQmon
- R-value

Protocols

In communications theory a **protocol** is a recipe for parties to connect and exchange information

Standardization of protocols are needed for communications

- with equipment from different vendors
- with network operated by different service providers

Certain issues need to be solved by most if not all protocols, e.g.,

- peer discovery
- negotiation (handshake)
- keep-alive (heartbeat)
- extensibility (TLVs, objects) and versioning
- framing, formatting, data representation
- resilience
- security

yet protocols are not (yet) universally designed for re-use

This is one of the motivations behind **Software Defined Networking**

Protocols and Algorithms

Once there was no overlap
between *communications* (telephone, radio, TV)
and *computation* (computers)
but this dichotomy has blurred

The differentiation can still be seen in the terms *algorithm* and *protocol*

In communications theory a **protocol** is a recipe for parties
to connect and exchange information

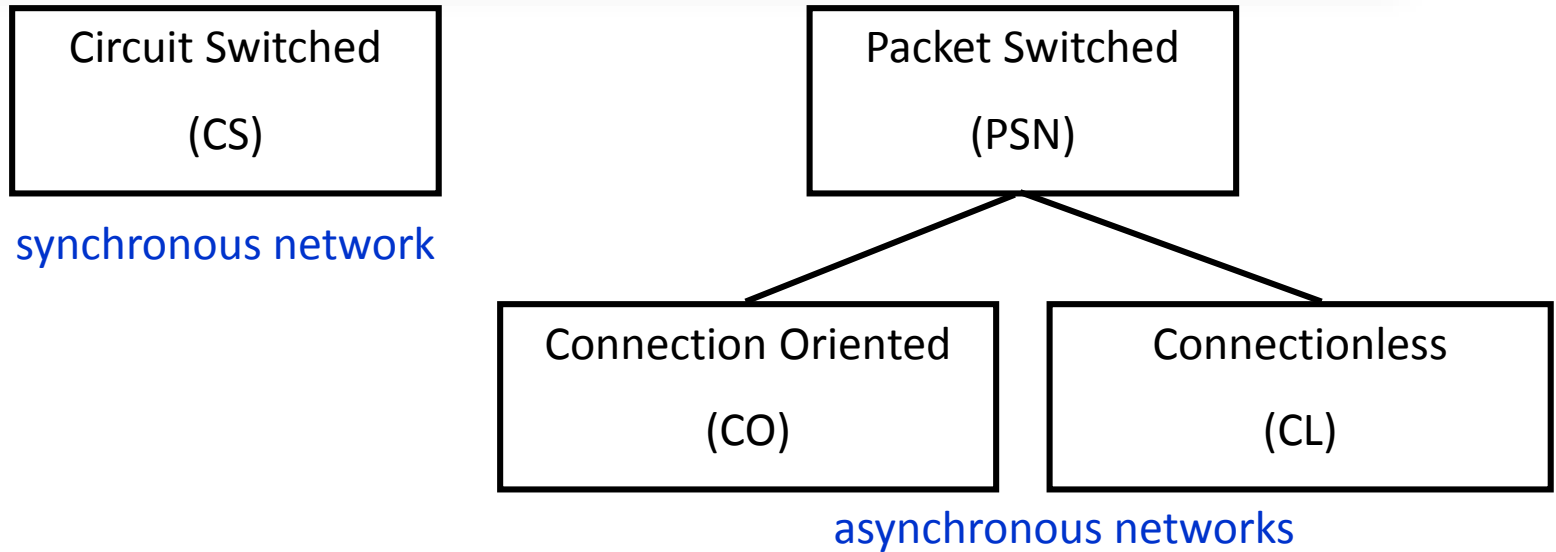
In computer science an **algorithm** is a recipe for a computational device
to carry out a task

Protocols are to *communications* what *algorithms* are to *computation*

SDN advocates replacing protocols by algorithms

Network Functions Virtualization advocates running these algorithms
on standard computational platforms

CS, CO, CL



Circuit Switched networks sent data at a **Constant Bit Rate**)

Packet Switched Networks send packets of data only when needed

PSNs have replaced CS ones, because of the efficiency of Statistical Multiplexing

Today there are two global networks

- **Public Switched Telephony Network (CS)**
- **Internet (CL)**

but the Internet runs over CO networks

OAM for PSNs

OAM is more complex and more critical for PSNs

In addition to previous problems, such as

- 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

OAM remains 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

The PSTN – the first network

The **P**ublic **S**witched **T**elephone **N**etwork

- was the first communications network
- was not planned, but rather grew by mergers and acquisitions
- is not a single network, but rather an internetwork of regional networks

The PSTN was originally an analog network

but migrated to become a digital core with analog subscriber lines

Many innovations were invented for the PSTN, including:

- addressing, hierarchical network topology, optimal path computation
- multiplexing (mux, duplexing, inverse mux, **M**ultiple **A**ccess)
- 3 planes (user, control, management), network planning
- OAM, APS, frequency distribution
- billing, SLAs, customer service

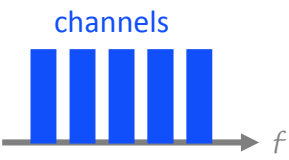
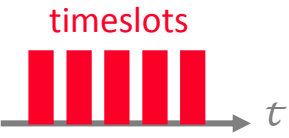
The PSTN employs many mux technologies (TDM, PDH, SDH)

The PSTN is presently being phased out (being replaced by the Internet)
but many issues remain before it will be completely shut down

Multiplexing

While user *local loops* only need to support a single call between Central Offices the PSTN muxed multiple calls on a single *trunk*

The PSTN employed several different mux methods

- FDM divides analog signal in the frequency domain 
- TDM divides digital signals in the time domain 
- PDH muxes high rate digital signals with *plesiochronous* clocks using stuffing bits to compensate for different
(E1=2.048Mbps, E2=8.448Mbps, E3=34.368Mbps
T1=1.544 Mbps, T3=6.312Mbps, T3=44.736Mbps)
- SDH muxes high rate digital signals with *plesiochronous* clocks using pointers to indicate the beginning of the data
(STM-1≈155Mbps STM-4≈622Mbps, STM-16≈2.488Gbps, STM-64≈9.953Gbps)

Multiplexing terminology

There are mechanisms to efficiently utilize *links* in a *network*

- **Duplexing** (half/full duplex)
sending information in both directions on same link
examples: FDD, TDD
- **Multiplexing**
sending multiple flows of information on same link
examples: FDM, TDM
- **Inverse multiplexing**
sending a single flow of information on multiple links
examples: LAG, link bonding, ECMP, VCAT
- **Multiple Access**
multiplexing uncoordinated users
examples, FDMA, TDMA, CDMA



Path computation and routing

Finding an optimal path through a packet network can be performed by

- path computation : centrally (by a God-box)
- routing : distributed routing protocols

The PSTN is based on path computation

The Internet is based on routing, but is starting to include PC

Distributed routing protocols are limited to

- finding simple connectivity
- minimizing number of hops

but can not perform more sophisticated operations, such as

- guaranteeing isolation
- optimizing paths under constraints (e.g., security)
- setting up non-overlapping backup paths (Suurballe problem)
- integrating networking functionalities (e.g., NAT, firewall) into paths


This is why MPLS created the **Path Computation Element** architecture and recently **Software Defined Networking** is being studied

Data, control, and management planes

It is worthwhile to distinguish between :

- forwarding
- routing (i.e., learning how to forward)
- administration (setting policy, service commissioning, monitoring, billing, ...)

This leads to defining three *planes* – *data* (or *user*), *control*, and *management*



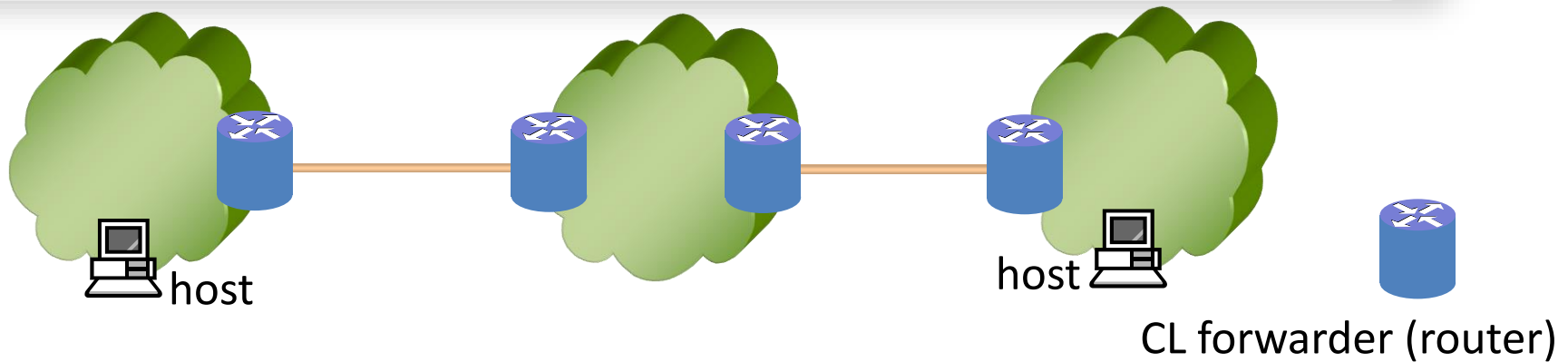
management plane	control plane
centralized (in N etwork O perations C enter)	distributed
based on human intervention (alarms)	automatic (e.g., routing protocols, APS)
relatively slow	relatively fast

Over time intelligent software replaced human intervention
erasing much of the control/management distinction

The difference that remains is that

- the management plane is slow and *centralized*
- the control plane is fast and *distributed*

Connectionless Forwarding (e.g., IPv4, IPv6)



A PSN is connectionless (CL) if

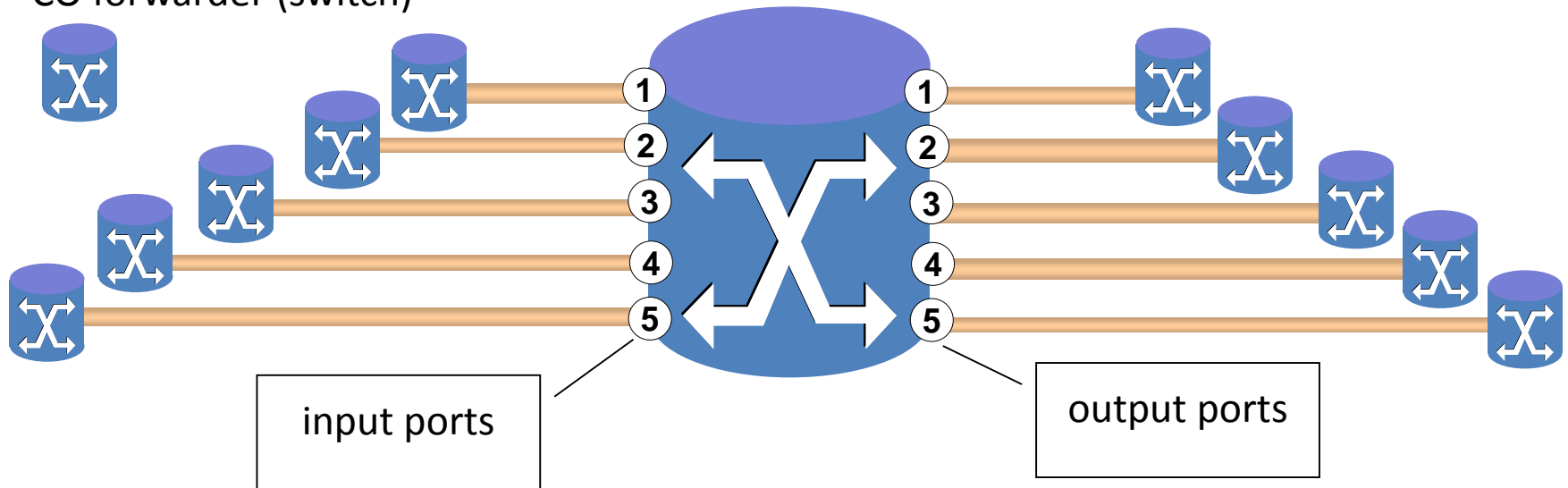
- no setup is required before sending a packet
each router makes an independent forwarding decision
- packets are self-describing
packet inserted anywhere will be properly forwarded
- IP forwarding detailed in RFC 1812 (IP forwarding walk-through)
hundreds of *software* cycles per packet (even with hardware switch fabric)

Note:

- the address **must** have *global significance*
- IP only runs between routers
it relies on a L2 protocol (Ethernet, PPP) from router to host

Connection Oriented Forwarding (e.g., ATM, MPLS)

CO forwarder (switch)



For CO forwarding global addresses are not required

Each forwarder maintains *forwarding table* (or table per input port)

Forwarding is simple, and can be performed by *hardware*

The control plane :

- route must be *set-up* (table must be updated) before data sent
- set-up may be manual or signaled
- once route no longer needed it should be *torn-down*

The OSI 7-layer model (X.200)

When Kleinrock designed CL packet switching
the tremendous complexity was apparent
(as compared to CS or CO networks)

So he separated the problem into *layers*

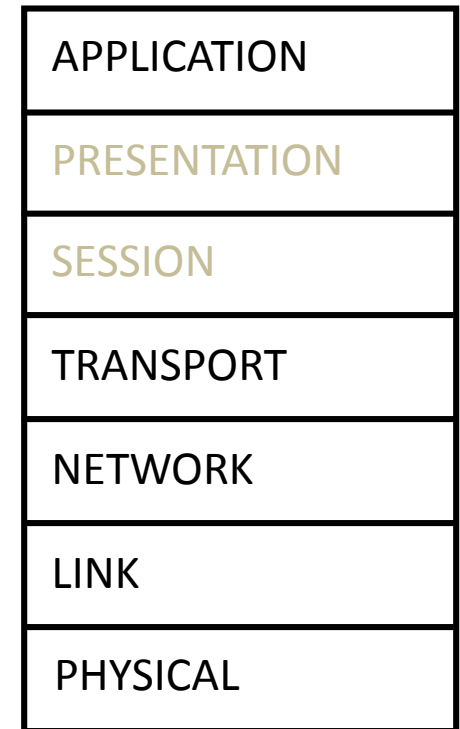
Each layer solved on problem
and required the other layers for all the rest

Each expert knew his layer well
and how to interface to the layers above and below

But this layering introduced extreme inefficiencies
(e.g., in VoIP 100B on-the-wire for a 10B payload)

This layering was never fully implemented
(e.g., Ethernet is actually 2 or 3 layers, MPLS does not fit anywhere)

Also, it is difficult to add new layers
for problems not originally envisioned (e.g., OAM, mux, security)



The G.80x model

A lesson learned as the PSTN evolved was the importance of **layer networks**

Each layer network is an independent network in its own right
independently designed and maintained

There must be an *operational reason* for each layer network

All layer networks should be described using the same tools
and there must be a client/server relationship between neighboring layers

Layer violations should be avoided, as they can lead to

- misrouting or loss of information
- billing avoidance
- information theft, information tampering

Each layer network needs its own

- addressing and forwarding mechanisms
- OAM mechanisms to guarantee QoS for its client
- control protocols
- management
- security mechanisms

Layer Networks

In the new framework

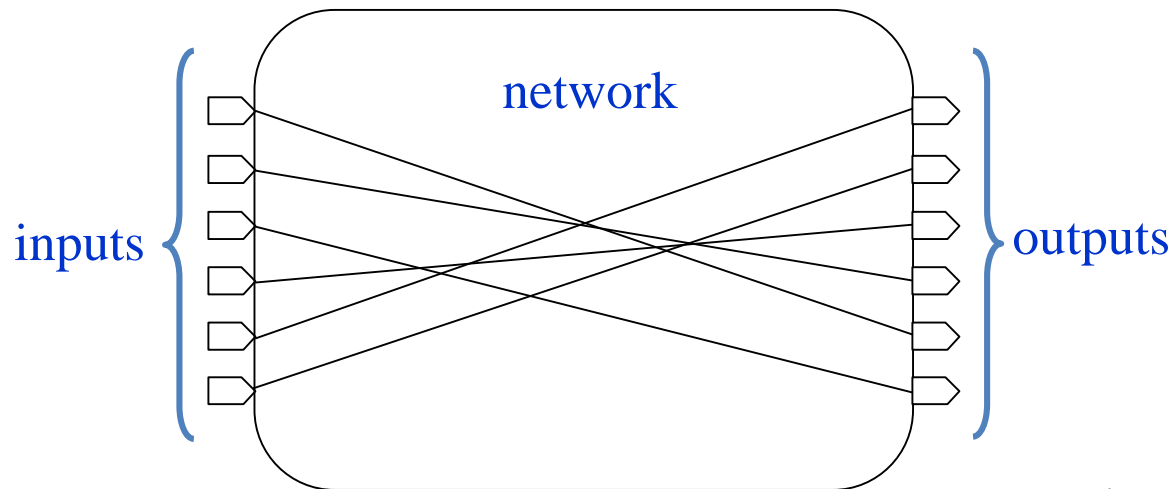
- each layer is an independent network with all functionalities
- called a **layer network**
 - because it exists at one layer
 - because it is a network unto itself

The goal of a layer network is to transport CI with minimal degradation

The association of an input with an output

is called a connection in a CO layer network

is called a flow in a CL layer network



G.80x modeling

G.80x provides

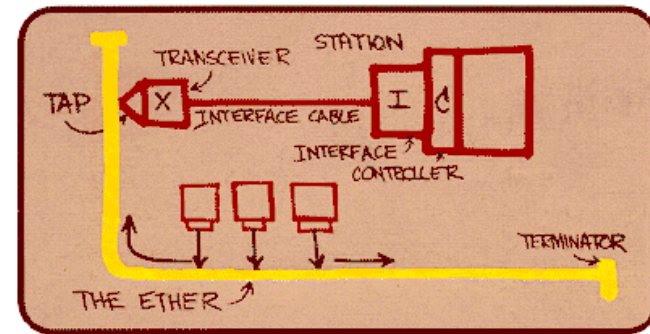
- a graphic modeling framework, defining
 - reference points (connection points, trail termination points, etc.)
 - links and connections
 - trails (connections plus OAM) and trail termination functions →▷
 - adaptation functions ◁
 - mux
 - interworking (tunneling and service interworking)
 - traffic conditioning (policing, shaping)

and providing mechanisms for

- proving correctness
 - adding functionality
- a set of maintenance tools
 - trail termination
 - anomalies, defects, faults, alarms
 - actions
- precise equipment specifications

Ethernet – PSN #1

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

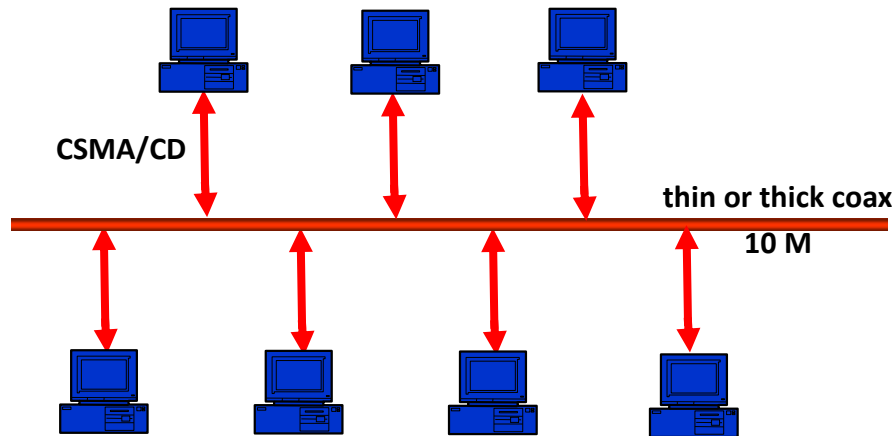


Metcalfe's original sketch of Ethernet

We may use the term today to describe

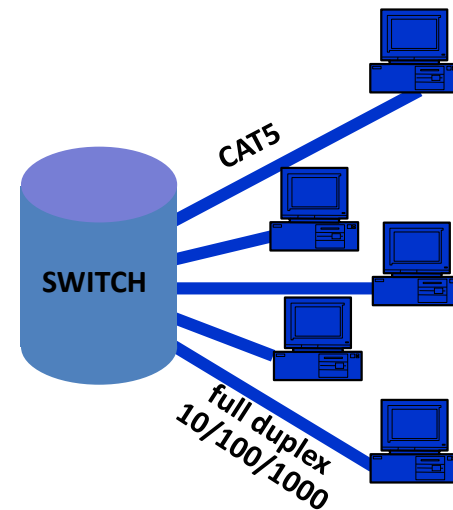
- full duplex 10G point-to-point optical links
- wireless Ethernet (WiFi) hot spots
- *Ethernet in the first mile* DSL access
- passive optical *GEAPON* networks
- metro Ethernet networks
- carrier-grade Ethernet services
- Ethernet **V**irtual **P**rivate **N**etworks
- etc.

Ethernet LANs, then and now



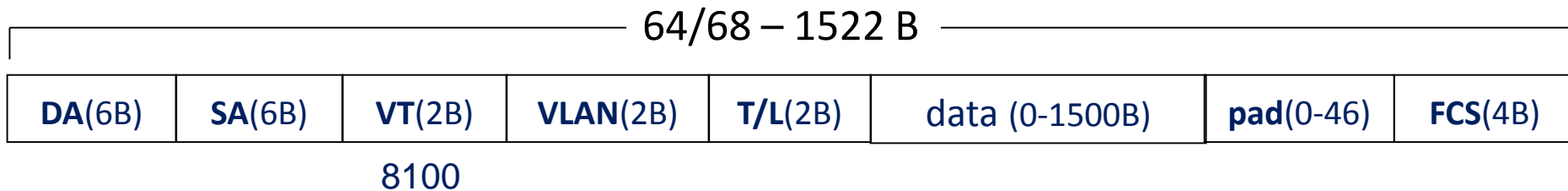
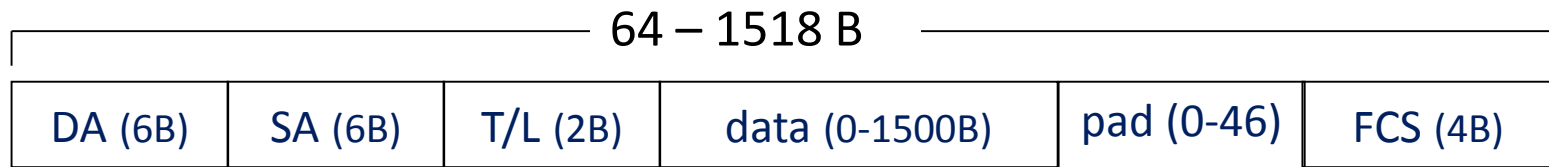
Bus topology
Single *collision domain*

Star topology
Independent full-duplex transmission
Active switch with buffers



MAC frame format

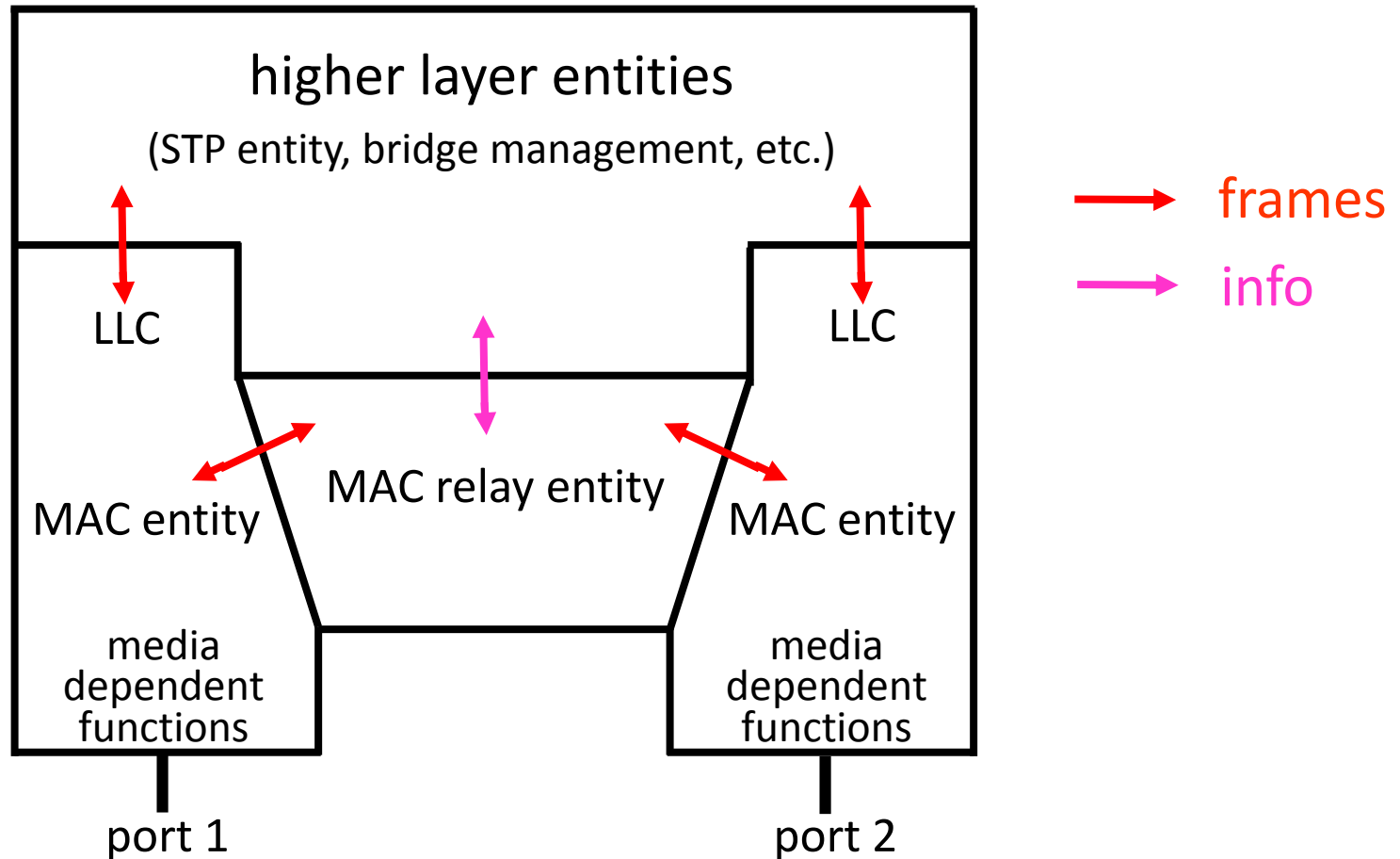
a *MAC frame* uses either of the following frame formats :



T/L is *Ethertype* or *Length*

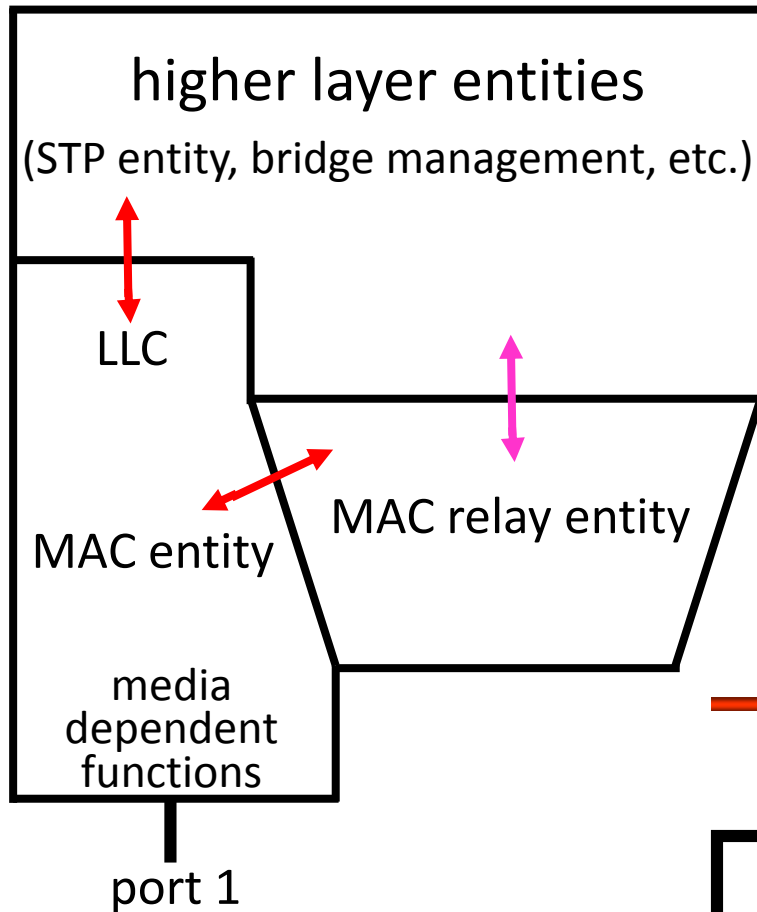
802.3as expanded frame size from 1500 to 2000B (since September 2006)

Baggy pants model



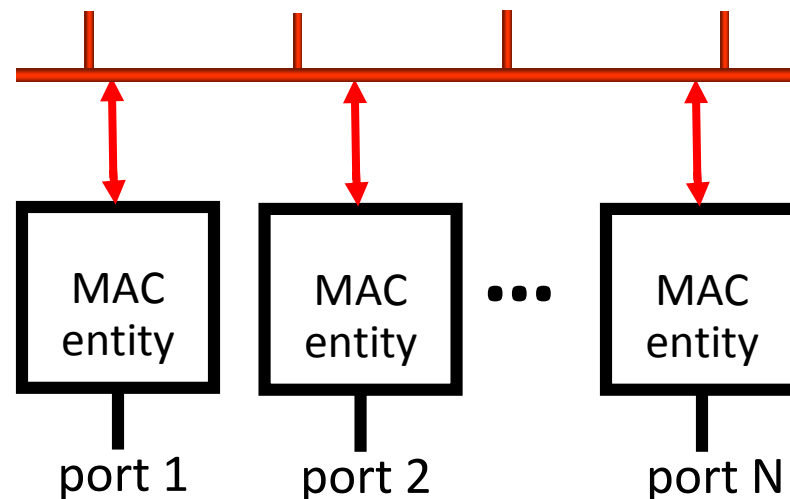
Note: a bridge must have at least 2 ports
here we depict exactly 2 ports

Extension to N ports



Ethernet bridges are not *forwarding* devices
but rather *filtering* devices
There is never a decision as to *where* to forward
only *whether* to forward

802 standards do not enforce mechanisms
as long as *external operation* is correct
This led to more efficient Ethernet **switches**



Layer 2 control protocols

The IEEE (and others) have defined Ethernet *control protocols* (L2CPs) :

- **Spanning Tree Protocol** (RSTP, MSTP)
- **Slow protocols**:
 - **Link Aggregation Control Protocol**
 - **Link Aggregation Marker Protocol**
 - **Link (EFM) OAM**
 - **Ethernet Synchronization Message Channel**
- **802.1X**
- **PAUSE**
- **E-LMI**
- **GARP** (GMRP, GVRP)
- **Link Layer Discovery Protocol**

“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

The **MEF forum** and **ITU-T** defined such mechanisms, e.g.

- OAM
- deterministic (Connection-Oriented) connections
- service attributes (frame loss, frame delay, BW profiles)
- protection switching (G.8031 linear APS, G.8032 for rings)
- synchronization
- security

(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** (shaping/policing)
per EVC, per CoS, per UNI
specified by **CIR**, CBS, EIR, EBS, ...

Token buckets

A BW profile is defined 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

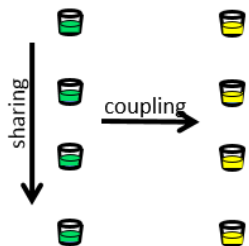
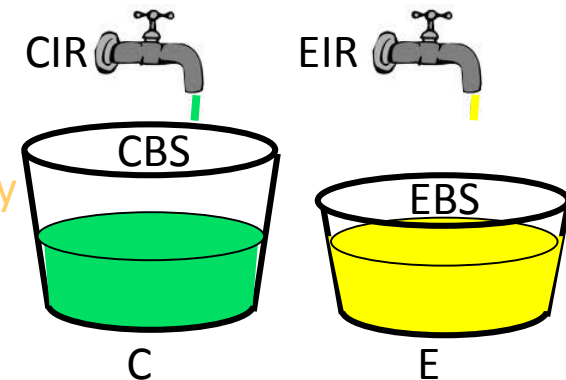
frame is **yellow** and its length of tokens is debited from E bucket

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



If there are multiple QoS levels (PCP field)
then we have may have sharing and/or coupling

Ethernet OAM functionality

Carrier Ethernet relies on Ethernet OAM

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

The functions provide include:

- Continuity Check / Connectivity Verification
- LoopBacks (intrusive, nonintrusive)
- defect notification (AIS, RDI)
- performance monitoring (loss, delay, delay variation, bucket parameters)

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 rings ?

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

The standard open loop APS is G.8032

- strives for 50 ms protection (< 1200 km, < 16 nodes)
- standard Ethernet format but incompatible with STP
- uses Y.1731 CCM for failure detection
- employs Y.1731 extension for R-APS signaling (opcode=40)
- complex due to requirement to avoid loop creation

IP – PSN #2

IP defines 3 NEs: hosts, routers, middleboxes

End-to-end (E2E) principle

*All functionality should be implemented
only with the knowledge and help of the application at the end points*

The *hourglass* model

- *IP (I3) is the common layer*
- *below IP (L3) is not part of IP suite, above is*

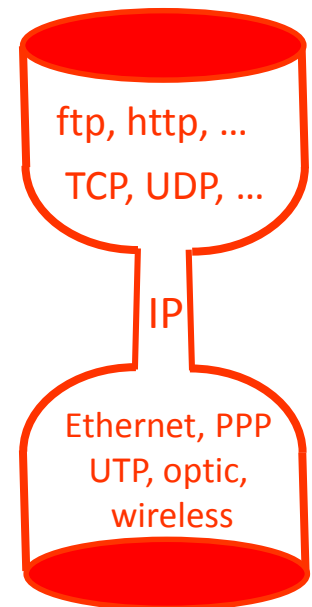
IP forwarding is

- connectionless
 - and thus Best Effort – since resources can't be reserved
- on a hop-by-hop basis

Unicast IP forwarding is based on a **Destination Address (DA)**

IP addresses are aggregated into subnetworks

Thus unicast forwarding uses Longest Prefix Match



FIBs

A router looks deduces how to forward a packet by (RFC 1812)

- observing the IP header
- performing a sequence of sanity checks
- deducing the **F**orwarding **E**quivalence **C**lass
- consulting the **F**orwarding **I**nformation **B**ase

The next router does the same thing all over again

(and hopefully the packet will arrive at its destination)

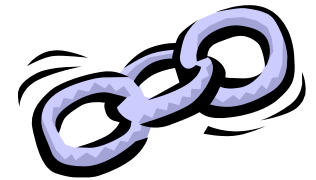
The FIB associates address prefixes with **N**ext **H**ops (NHs)

(and, to save an additional lookup, usually with L2 addresses as well)

The FIB is built from **R**outing **I**nformation **B**ases, static routes, and policy

IP Routing types

- **Distance Vector** (Bellman-Ford), e.g. RIP, RIPv2, IGRP, EIGRP
 - send <addr,cost> to neighbors
 - routers maintain cost to all destinations
 - need to solve “count to ∞ problem”
- **Path Vector**, e.g. BGP
 - send <addr,cost,path> to neighbors
 - similar to distance vector, but w/o “count to ∞ problem”
 - like distance vector has slow convergence*
 - doesn’t require consistent topology
 - can support hierarchical topology => exterior protocol (EGP)
- **Link State**, e.g. OSPF, IS-IS
 - send <neighbor-addr,cost> to all routers
 - determine entire flat network topology (SPF - Dijkstra’s algorithm)
 - fast convergence*, guaranteed loopless => interior routing protocol (IGP)



*convergence time is the time taken until all routers work consistently
before convergence is complete packets may be misforwarded, and there may be loops

What is the relationship between all these routing types ?

The Internet is composed of **Autonomous Systems** run by network operators

Each AS is truly autonomous

- AS is a single entity to the outside world
- routers in the same AS obey a common policy, and *trust each other*
- one AS can *request* another to forward a packet, but can not *force* it to

Inside an AS topology information is shared

- Link State routing (OSPF, IS-IS)

Between AS's topology information is on a need-to-know basis

- Path Vector routing (BGP)

Each AS has at least one **AS Border Router**

- one leg inside the AS (OSPF/IS-IS)
- one leg between AS's (BGP)
- transit AS has at least 2 ASBRs
- ASBR application uses policy to decide what to advertise for IDR
- peering relationships influence policy

Forwarding Equivalence Classes

A FEC is the set of all packets that are to be treated in the same way

By the equivalence class theorem every packet belongs to one unique FEC

Packets in the same FEC should follow the same path

but in IP this is not directly *enforced*

since each successive router reclassifies the packet's FEC

If the router could insert information into the packet
informing the next router of its FEC

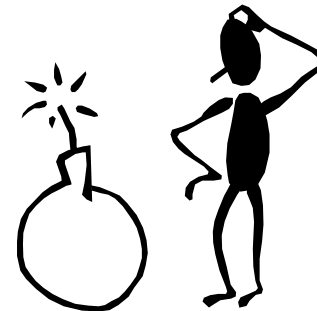
- this would save a lot of processing at the following routers
- the subsequent forwarding would be CO instead of CL

This leads us to define MPLS

Problems with IP routing

IP is great – but not perfect !

- scalability
 - router table overload
 - routing convergence slow-down
 - increase in queuing time and routing traffic
 - problems specific to underlying L2 technologies
- hard to implement load balancing
- QoS and Traffic Engineering
- problem of routing changes
- difficulties in routing protocol update
- lack of VPN services



MPLS – PSN #3



label switching adds the strength of CO to CL forwarding

label switching involves three stages:

- routing (topology determination) using L3 protocols
- path setup (label binding and distribution) perhaps using new protocol(s)
- data forwarding

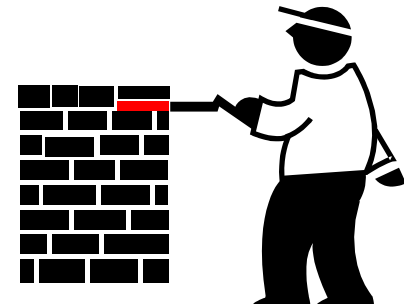
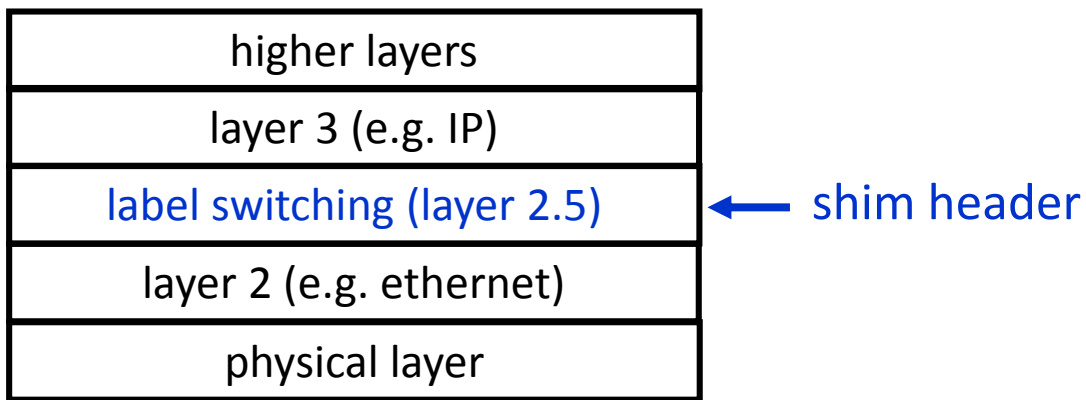
label switching the solution to *all* of the above problems

- speeds up forwarding
- decreases forwarding table size (by using local labels)
- enables support for QoS and arbitrary granularity FECs
- load balancing by explicitly setting up paths
- complete separation of routing and forwarding algorithms
 - no new routing algorithm needed
 - but new signaling algorithm may be needed

Where is it?

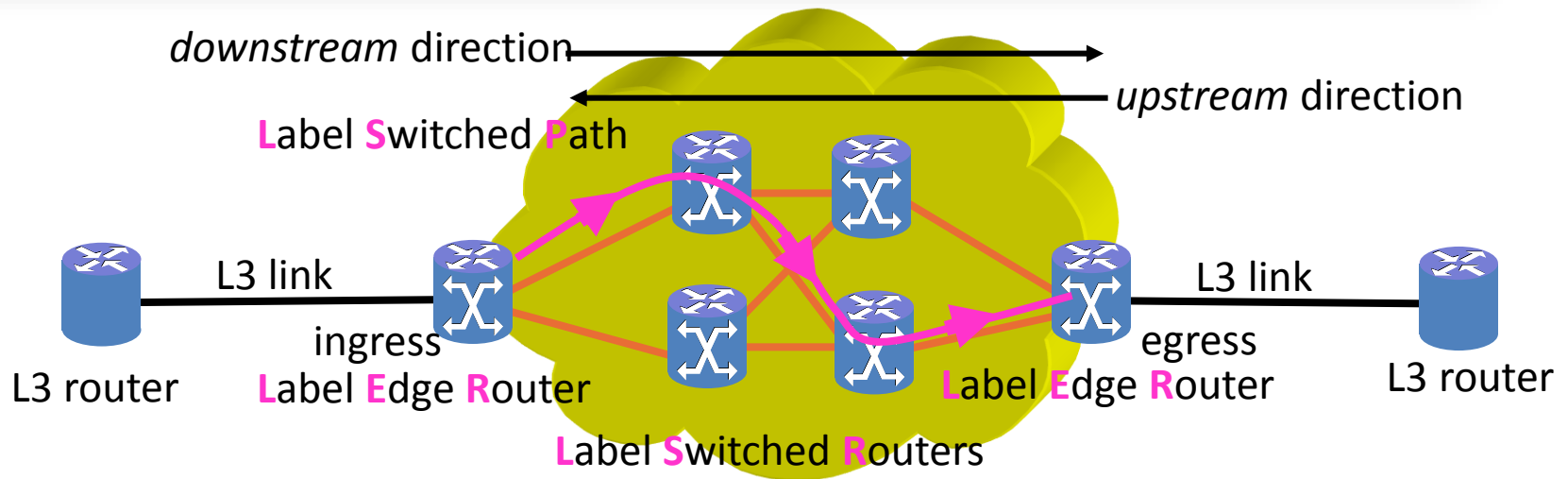
Unlike TCP, the label switching CO layer lies *under* the CL layer

If there is a broadcast L2 (e.g. Ethernet), the CO layer lies *above* it



Hence, label switching is sometimes called **layer 2.5 switching**

Label Switching Architecture



Label switching is needed in the **core**, access can be L3 forwarding*

Core interfaces the access at the edge (ingress, egress)

LSR router that can* perform label switching

LER LSR with non-MPLS neighbors (LSR at edge of core network)

LSP unidirectional path used by label switched forwarding (ingress to egress)

* not every packet needs label switching

Label Stacks

Since labels are structure-less, the label space is flat

Label switching can support arbitrary levels of hierarchy by using a *label stack*



Label forwarding based only on **top label**

Before forwarding, three possibilities (listed in NHLFE) :

- **read** top label and **pop**
- **read** top label and **swap**
- **read** top label, **swap**, and **push** new label(s)

Label stacks are needed for **Fast ReRoute**, **Virtual Private Networks**, and **PWs**

MPLS *Shim* Header



The shim format is:

Label there are 2^{20} different labels (+ 2^{20} multicast labels)

Traffic Class (ex-EXP)

was **CoS** in Cisco Tag Switching
may influence packet queuing
QoS may be via E-LSP or L-LSP

Stack bit S=1 indicates bottom of label stack

TTL decrementing hop count
used to eliminate infinite routing loops
and for MPLS traceroute
generally copied from/to IP TTL field

Special (reserved) labels

- 0 IPv4 explicit null
- 1 router alert
- 2 IPv6 explicit null
- 3 implicit null
- 13 MPLS-TP GAL
- 14 Y.1711 OAM label

S=0	top label
S=0	another label
S=0	yet another label
S=1	bottom label

MPLS Single Forwarding Algorithm

IP uses different *forwarding* algorithms
for unicast, unicast w/ ToS, multicast, etc.

LSR uses one *forwarding* algorithm (LER is more complicated)

- read top label L
- consult Incoming Label Map (forwarding table) [\[Cisco terminology LFIB\]](#)
- perform label stack operation (pop L , swap $L - M$, swap $L - M$ and push N)
- forward based on L 's Next Hop Label Forwarding Entry

LER's forwarding algorithm is slightly more complex

- check if packet is labeled or not
- if labeled
 - then forward as LSR
 - else
 - lookup destination IP address in FEC-To-NHLFE Map
 - if in FTN
 - » then prepend label and forward using LSR algorithm
 - » else forward using IP forwarding

MPLS flavors

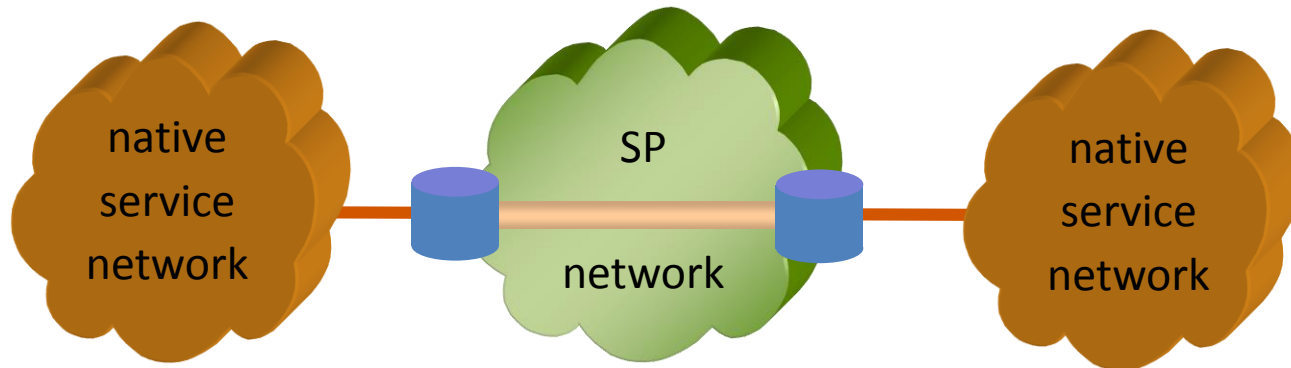
We can now distinguish *four* flavors of MPLS :

1. plain **vanilla MPLS** (usually with LDP, perhaps with RSVP-TE for FRR)
not true CO – pinned to route not to NEs
used in Internet core
2. MPLS for **L3VPN** services (RFC 4364 <ex-2547> using BGP)
used to deliver VPN services to businesses
3. MPLS-TE (currently with RSVP-TE)
true CO with resource reservation
used when SLA guarantees given
4. MPLS-TP (usually with management system, can use RSVP-TE)
does not assume the existence of IP forwarding plane
does not require control plane – can work with management OSS
implements OAM and APS functionality

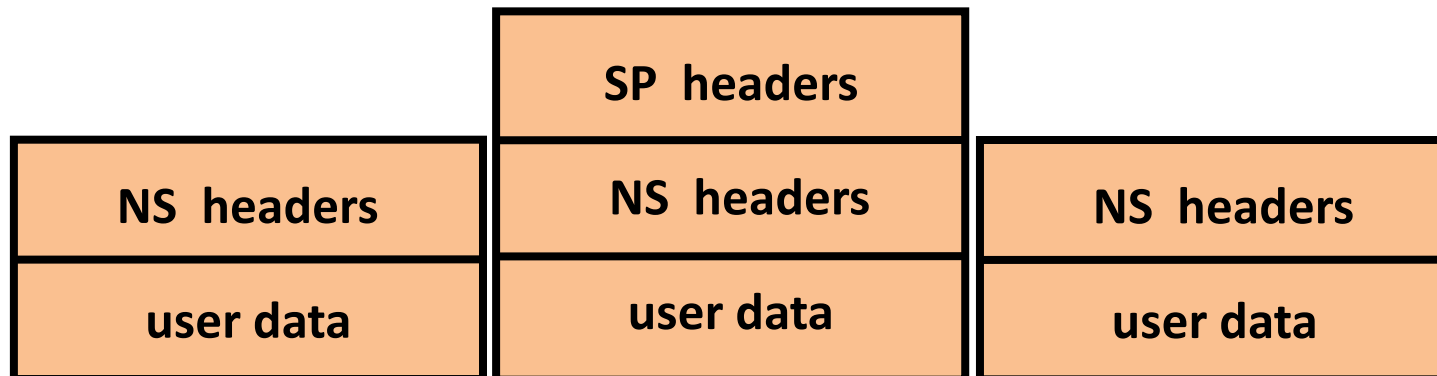
Tunneling

Tunneling is a simple way of interworking that is applicable when both end-points sit on the same native network

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



Note that the native service protocol is not terminated



Pseudowires

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

PWs for transport of:

- TDM/SDH
- ATM/frame relay
- Ethernet

have been defined to run over :

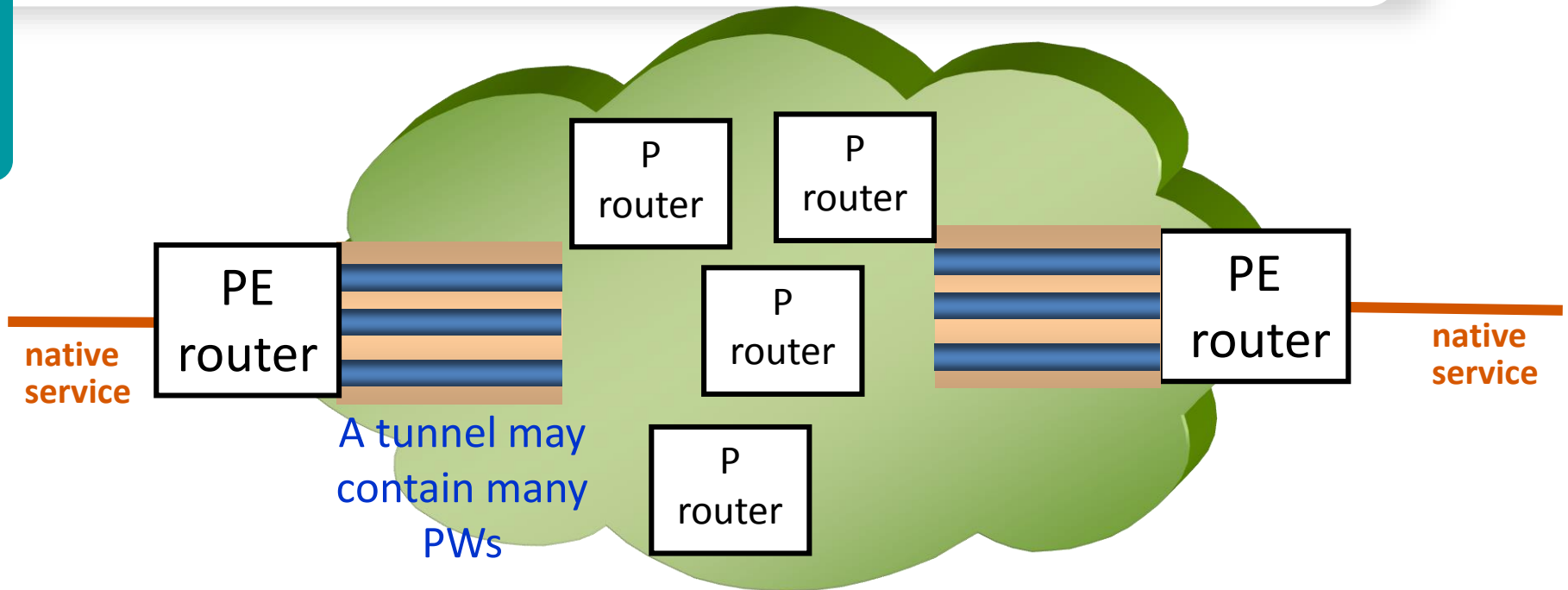
- MPLS
- L2TPv3 over IP
- Ethernet (only TDM PWs)
- UDP/IP (only TDM PWs)

PWs are bidirectional (unlike MPLS LSPs)

MPLS is a natural PSN for PWs :

- transport MPLS tunnel set up between PEs
- multiple PWs may be set up inside tunnel
- native packet/frame encapsulated with 2 labels
- PW label is **Bottom of Stack** (S=1)
- LSRs completely unaware of individual PWs, only the PE knows!

Pseudowires



PW label is not a *real* label

it just identifies native service instance

P routers don't know about PWs

just how to get to egress PE

With MS-PWs, PW labels becomes real labels

tunnel label
PW label
PW control word
payload

Communications security

Communications security (COMSEC) means preventing unauthorized access to communications infrastructure and communicated messages, while still providing the communications service between intended parties

There are many important security mechanisms:

- Physical security – preventing access to communications devices and links
- Emission security – preventing interception and jamming
- Authorization – preventing unauthorized access to resources
- Source authentication – confirming the source of a message
- Integrity – preventing tampering with messages
- Confidentiality – preventing eavesdropping
- DoS blocking – preventing Denial of Service
- Topology hiding – thwarting traffic analysis
- Anti-hacking – preventing injection of computer malware
- Privacy – protecting user's personal data from mining and directed collection

Security needs to be applied at every network layer!

Authorization

AAA (user) Authentication, Authorization, and Accounting
means any mechanism for controlling access to resources

Authorization (warning – often called *authentication*) means

- verifying the identity of a *supplicant* (party requesting service)
- to an *authenticator* (party authorizing service)
- which may be *three-factor authentication*
 - something you are (fingerprint, retina scan)
 - something you have (cellphone, password generator)
 - something you know (password, identify questions)

In order to avoid storing password in the clear, we use crypto-hashes

1. maps strings (vectors) of arbitrary length to strings (vectors) of fixed length
2. ensures that small change in input map to large changes in output
3. calculating the hash function is computationally easy
finding an input that produces a given output is computationally hard

CHAP

We can't use passwords to access a remote authenticator, since

- sending the password in-the-clear discloses it to an eavesdropper
- sending the crypto-hash also discloses it to a eavesdropper

Instead, we use **C**hallenge **H**andshake **A**uthentication **P**rotocol

- *authenticator* sends a *challenge* message to the *supplicant*
- supplicant responds with crypto-hash of challenge+password
- authenticator compares response with expected hash value
- if match supplicant admitted, else rejected

Well-known crypto-hashes include :

- MD5 (no longer considered secure)
- Secure Hash Algorithm SHA1 (widely used, no longer considered secure)
- Secure Hash Algorithm SHA2
- Secure Hash Algorithm SHA3 (new)

Extensible **A**uthentication **P**rotocol is a authentication *framework*
and runs over links layers (PPP, Ethernet, WiFi) without needing IP

802.1X is an IEEE standard for authenticating users of a LAN or WLAN

Integrity

We can provide *integrity* using a Message Authentication Code (MAC)

A MAC is a short block of information that

- is uniquely determined by the message
- verifies that the message has not been modified
i.e., it is highly unlikely that a modified packet has the same MAC (collision)
- is difficult/impossible to forge

The MAC is inserted into packet headers
and is verified upon receipt

A Hash-based Message Authentication Code (HMAC)

uses a crypto-hash as a MAC (but block ciphers and other mechanisms can be used)

The simplest way to prevent forging MACs is by using a *shared key*

- Alice calculates MAC from message + key, and inserts into packet
- Bob calculates MAC using message + key, and compares to MAC in packet
- Eve, not knowing the key, can not forge the MAC

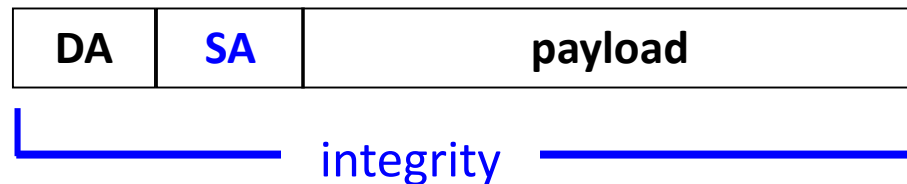
Source Authentication

Ethernet and IP packets contain Source Addresses (SA)
but these can be readily forged

MACs can also be used to *authenticate* a packet's *source*
that is, to prove that the SA correctly indicates the packet's source

We can reuse integrity mechanisms (e.g., MACs) to solve this problem !

All that is needed is to have the (H)MAC protect the SA
if the MAC is correct, then the SA indeed belongs to the claimed sender



Replay Attacks

Another type of attack is the replay attack

Here the MiM intercepts a packet and resends it multiple times
(transfer\$100 → transfer\$100, transfer\$100, transfer\$100, transfer\$100)

Integrity and source authentication mechanisms do not detect replay attacks
since the MACs calculate correctly

We can reuse integrity mechanisms (e.g., MACs) to solve this problem !

To defend against replay attacks

- add or utilize a packet **Sequence Number** field
- have the MAC protect the sequence number field
- if duplicate SN is received (or integrity violated), then discard packet



Confidentiality

An important attack vector is *eavesdropping*

i.e., the packet content is observed by unintended parties

The standard countermeasure is *encryption*

There are two very different types of encryption – *symmetric key* and *public key*

Symmetric key encryption is based on the sides having a shared secret

Truly random one-time pads are provably secure *symmetric key* systems

DES and AES are also symmetric key systems

Public key encryption algorithms rely on hard-to-perform calculations, e.g.,

- factoring large integers
- finding discrete logarithms
- elliptic curve logarithms

Public key methods can be used for

- symmetric key distribution (Diffie Hellman)
- digitally signing documents

but you still need to authorize the other side first (certificates or web of trust)

SDN and NFV

Software Defined Networks (SDN)

SDN advocates replacing standardized networking protocols with centralized software applications that configure all the NEs in the network

Advantages:

- easy to experiment with new ideas
- control software development is much faster than protocol standardization
- centralized control enables stronger optimization
- functionality may be speedily deployed, relocated, and upgraded

Network Functions Virtualization (NFV)

NFV advocates replacing hardware network elements with software running on COTS computers that may be housed in POPs and/or datacenters

Advantages:

- COTS server price and availability scales with end-user equipment
- functionality can be located where-ever most effective or inexpensive
- functionalities may be speedily combined, deployed, relocated, and upgraded

SDN abstractions

At a high enough level of abstraction

all network elements perform the same task

- receive a packet
- observe packet fields
- apply algorithms (classification, decision logic)
- optionally edit the packet
- forward or discard the packet

and can be implemented by **whitebox** switches

With full knowledge of network topology and constraints

path computation can be solved by a graph optimization algorithm
run at a central location (SDN controller, *God-box*)

Whitebox switches are directly *configured* by the SDN controller

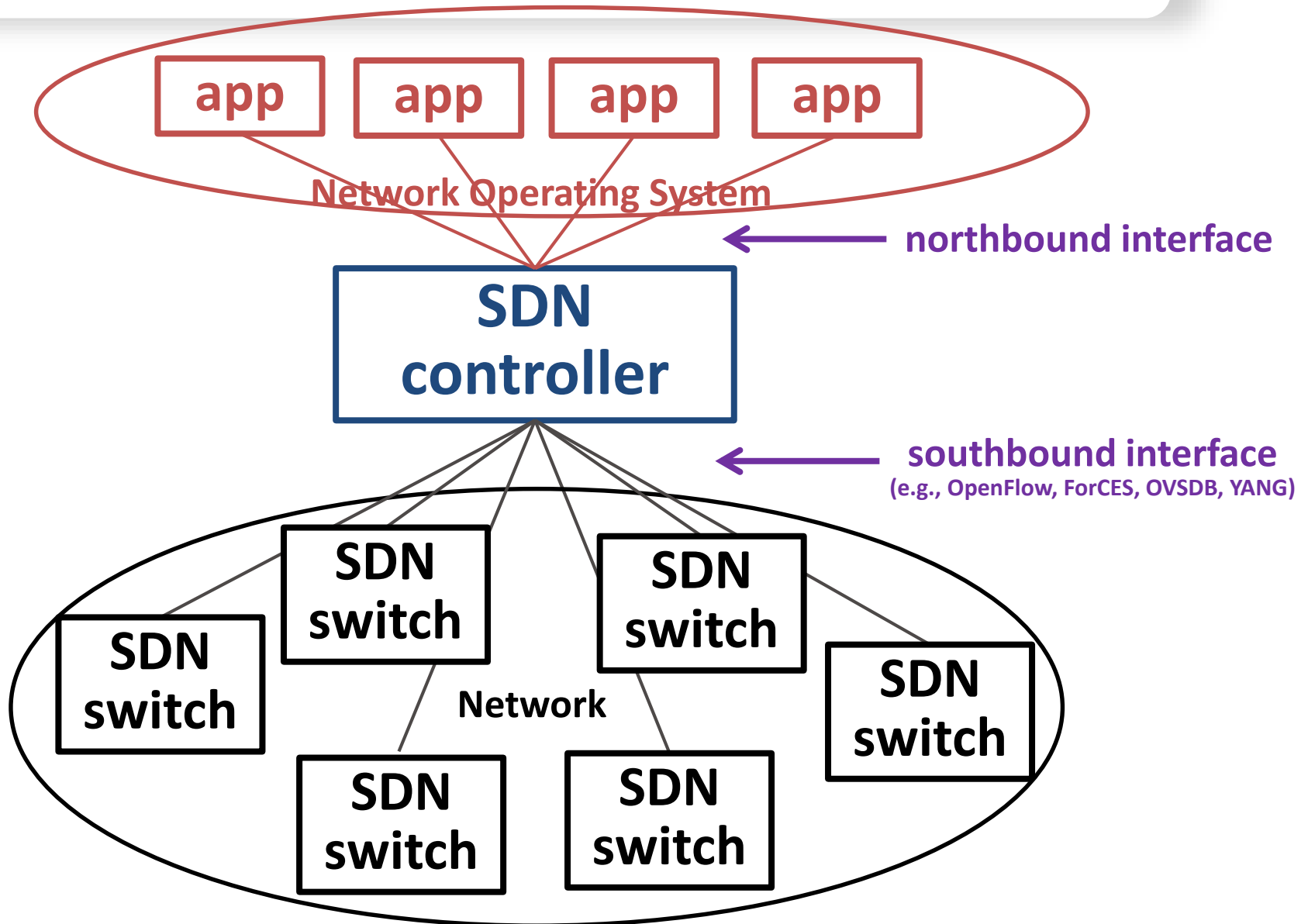
using a *southbound API* that is essentially a management protocol

Packets are associated with a flow to which they belong (FEC)

For even more flexibility, we use network applications

that sit above the SDN controller communicating via *northbound APIs*

SDN overall architecture



Virtualization

Virtualization here means the creation of a **virtual machine** (VM) that acts like an independent physical computer

A **VM** is software that emulates hardware (e.g., an x86 CPU) over which one can run software as if it is running on a physical computer

Many network elements (e.g., switches, routers, NATs, firewalls, IDS) may also be replaced by software running on a CPU or VM

This is called **NFV** and enables

- using standard COTS hardware (whitebox servers)
- fully implementing functionality in software
- consolidating equipment types
- optionally concentrating network functions in datacenters or POPs

Once a network functionality has been virtualized it is relatively easy to relocate it (e.g., SDN is an example of routing relocation)

Rich communications services

Traditional communications services are pure *connectivity* services
transport data from A to B

- with constraints (e.g., minimum bandwidth, maximal delay)

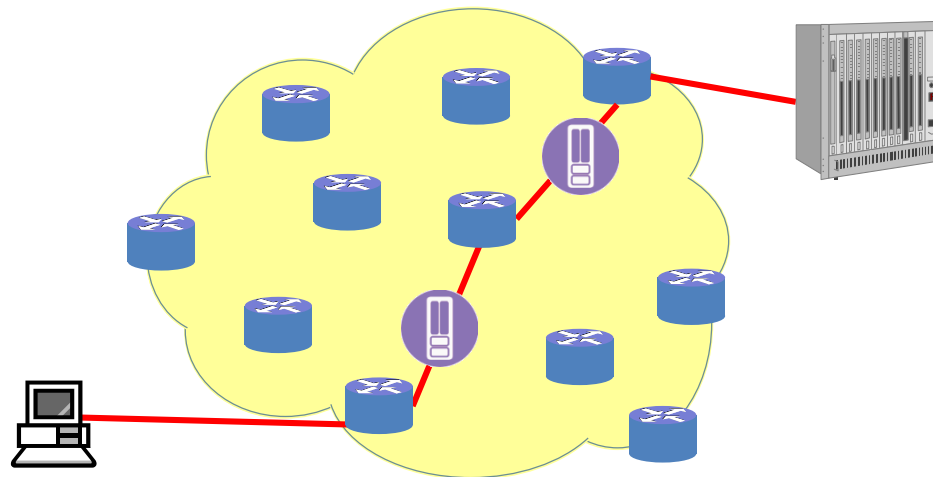
- with maximal efficiency (minimum cost, maximized revenue)

Modern communications services are richer

- combining connectivity and network functionalities

- e.g., firewall, NAT, load balancing, CDN, parental control, ...

Such services further blur the computation/communications distinction
and make service deployment optimization more challenging



Problems with rich services

Separate path computation and VNF placement is obviously suboptimal unless plentiful computational resources are available along the path

A solution was developed for the *joint PC/D-NFV optimization problem*






For rich services the end user still experiences *some* end-to-end QoE

- can we find easily measurable QoS parameters that determine this QoE ?
- can we find the precise relationship between QoE and these KPIs ?

It turns out that in general

- end-to-end QoS parameters such as packet loss rate and delay are useless
 - hop-by-hop QoS parameters are useless and even counterintuitive
- and new ideas are needed

Generations of cellular technologies

	1G	2G	3G	4G	5G
3GPP releases			4 - 7	8-9, 10-14	15, 16
era	1980s	1990s	2000s	2010s	2020s
services	analog voice	digital voice messages	WB voice packet data	voice, video Internet, apps	everything
devices					
data rate	0	100 kbps (GPRS)	10 Mbps (HSPA)	100+ Mbps (LTE/LTE-A)	10 Gbps (NR)
delay		500 ms	100 ms	10s ms	5 ms

What's wrong with 4G?

4G made possible:

- fast Internet access
- video reception and creation
- apps relying on location and identity
- always-on behavior

but suffers from numerous limitations:

- for some applications: data-rate too low
- for some applications: delay too high
- too few simultaneous connections (insufficient density)
- coverage too low / drop rate too high
- weak (if any) QoS guarantees
- price per bit too high (inefficient spectral use)
- power consumption too high (and thus battery life too low)
- poor support for new applications/markets (e.g., IoT, AR/VR, connected cars)
- no support for new mobility requirements (mobile hot spots, high speed)
- insufficient security/privacy

5G is being developed
to address 4G limitations

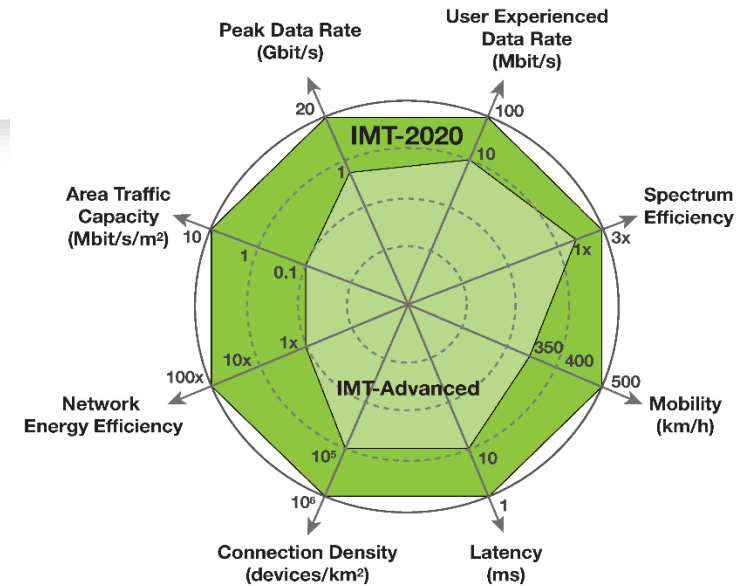
IMT-2020 goals

The ITU is defining performance targets for 5G that are 10 to 100 times *more* than 4G* :

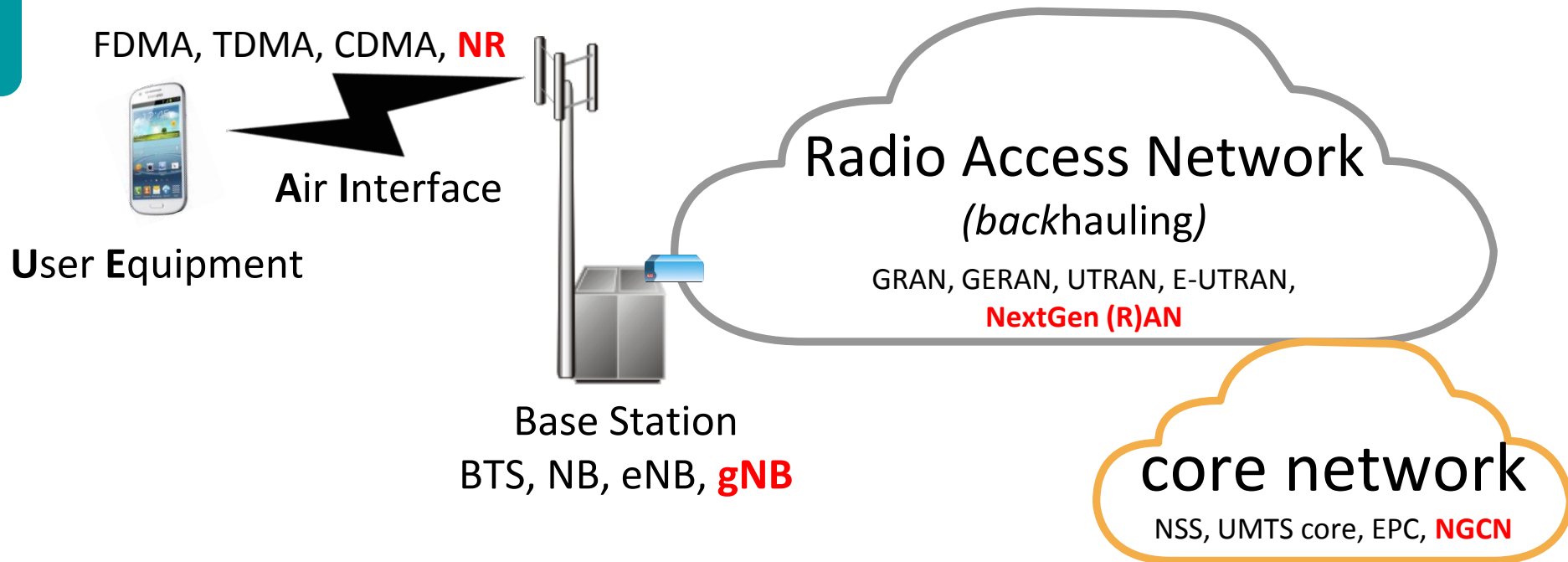
- **Peak data rate** (20 Gbps/device)
- **User experienced data rate** (100 Mbps)
- **Latency** (1 ms)
- **Mobility** (500 km/h and seamless transfer)
- **Connection density** (10^6 devices/km²)
- **Energy efficiency** (1/100 Joule/bit for both air interface and network)
- **Spectrum efficiency** (3 times the bps/Hz of LTE-A)
- **Area traffic capacity** (10 Mbps/m²)

However, it is not possible to attain all of these *at the same time* so 5G recognizes usage scenarios and includes mechanisms to separate traffic types (slicing)

* ITU-R M.2083-0



5G changes all cellular segments



Multiple 4G/5G co-existence options have been proposed

- Standalone (e.g., eNB-EPC and gNB-NGCN)
- Non-Standalone (gNB-LTE-EPC, gNB-EPC, eNB-NR, eNB-NGCN, ...)

Initial 5G will be Non-Standalone gNB connecting to LTE RAN and EPC

The End