Packet Switched Networks

The drive towards packet switching

With circuit switching, bandwidth (data-rate) is consumed
even when there is no information to be sent
For example, in voice calls each side speaks on average 50% of the time
but consumes 64 kbps even during silent intervals

In addition, the PDH/SDH hierarchies are efficient for transporting

- CBR clients of known bit-rates
- CBR clients of certain bit-rates

but are not designed to handle

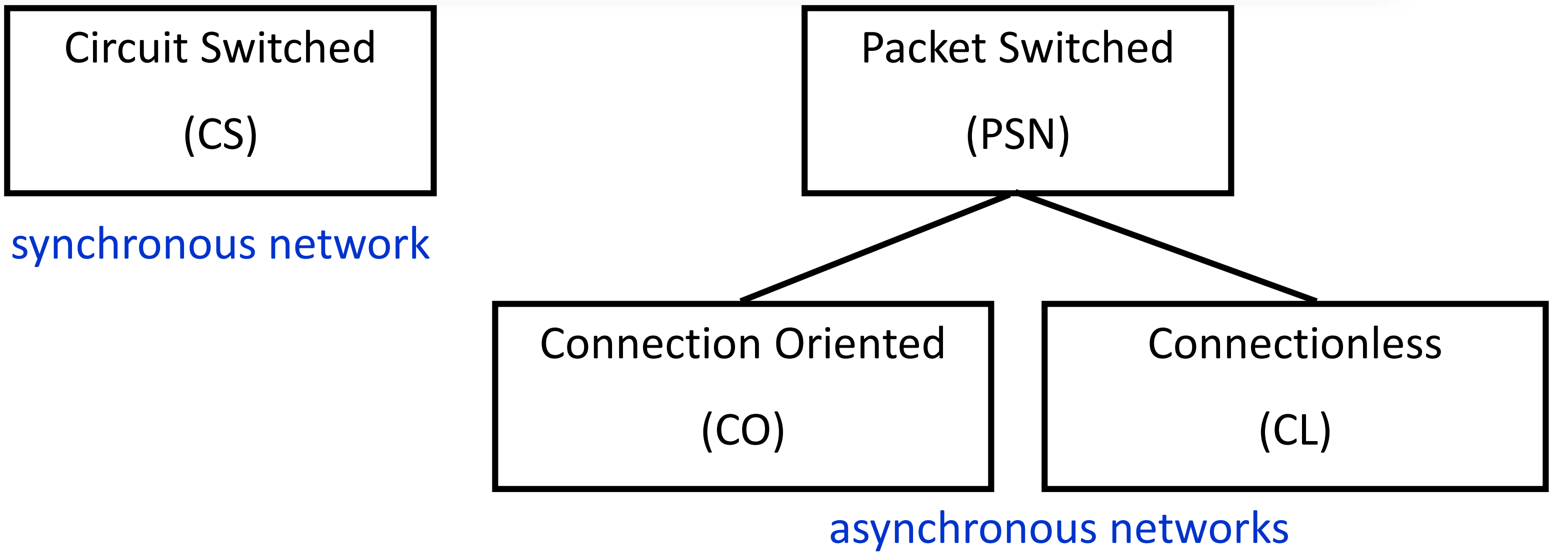
- CBR of arbitrary bit-rates
- Variable Bit Rate
- one-time messages

By encapsulating messages/files/streams in

- single packets
- packet flows
- real-time packet streams

one may achieve higher efficiencies through statistical multiplexing

Packet vs. Circuit Switching



Unlike CS networks where data is sent at a constant rate (**C**onstant **B**it **R**ate)
  with PSNs packets are only sent when information is available (VBR)

Packet switching enables statistical multiplexing, and thus is more efficient

CO packet forwarding is similar to circuit switching
  routing performed at set-up, forwarding is simple

CL packet forwarding is much more complex than circuit switching
  no set-up
  require expensive real-time forwarding computation

Network technologies

There are (were) many native network technologies for each mode

- CS: TDM, PDH, SDH, OTN
- CO: ATM, FR, MPLS, TCP/IP, SCTP/IP
- CL: UDP/IP, IPX, Ethernet, CLNP

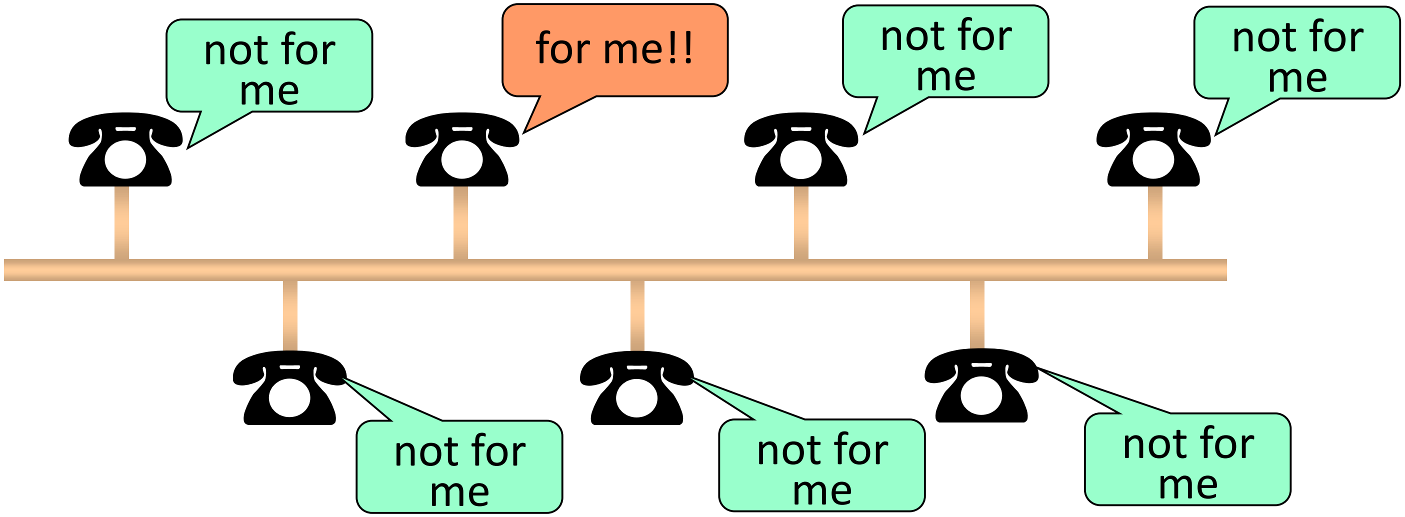
Over time a small number have prevailed (Ethernet, IP, MPLS)

Networks can be *layered* - any mode over any mode

but some layerings may involve performance loss

- CL over CS (e.g., IP over SDH) is easy
- CL over CO (e.g., IP over ATM) is easy
- CL over CO over CS (e.g., IP over ATM over SDH) is easy
- CO over CL (e.g., TCP/IP) is *harder*
- CS over CO (e.g., AAL1 over ATM) is *harder*
- CS over CL (e.g., TDMoIP) is *extremely hard*

Forwarding plane - Buses (e.g., Ethernet LANs)



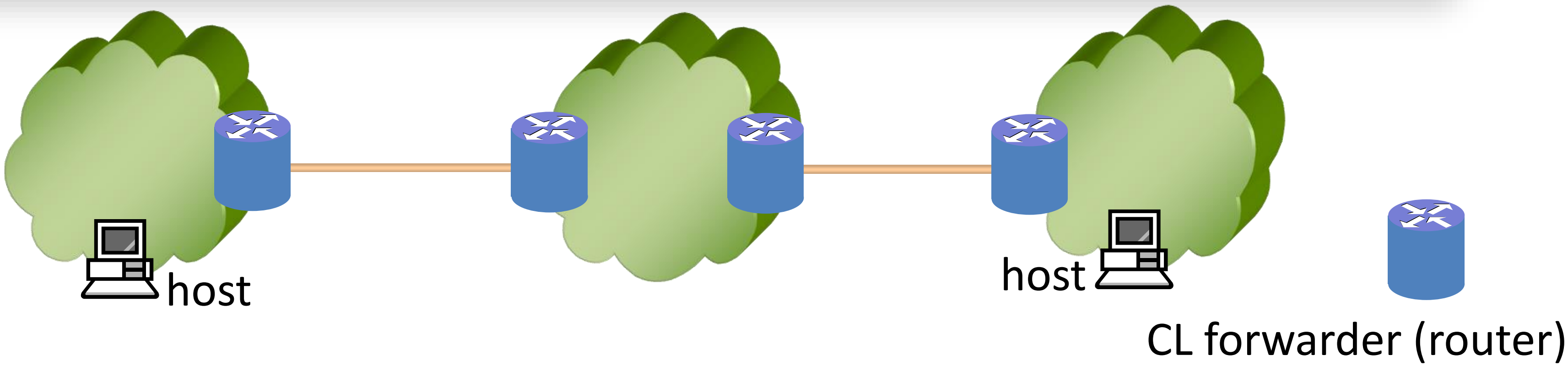
The simplest way to interconnect *hosts* is via a *bus*

- each message is sent (broadcast) to all possible recipients
- all receivers check the message's destination address
- if message destination address does not match receiver's address
  - message ignored (except in promiscuous mode)
- if message destination address matches receiver's address
  - message received

There is no forwarding in broadcast domains

Broadcast is limited to small **L**ocal **A**rea **N**etworks

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



A PSN is connectionless (CL) if

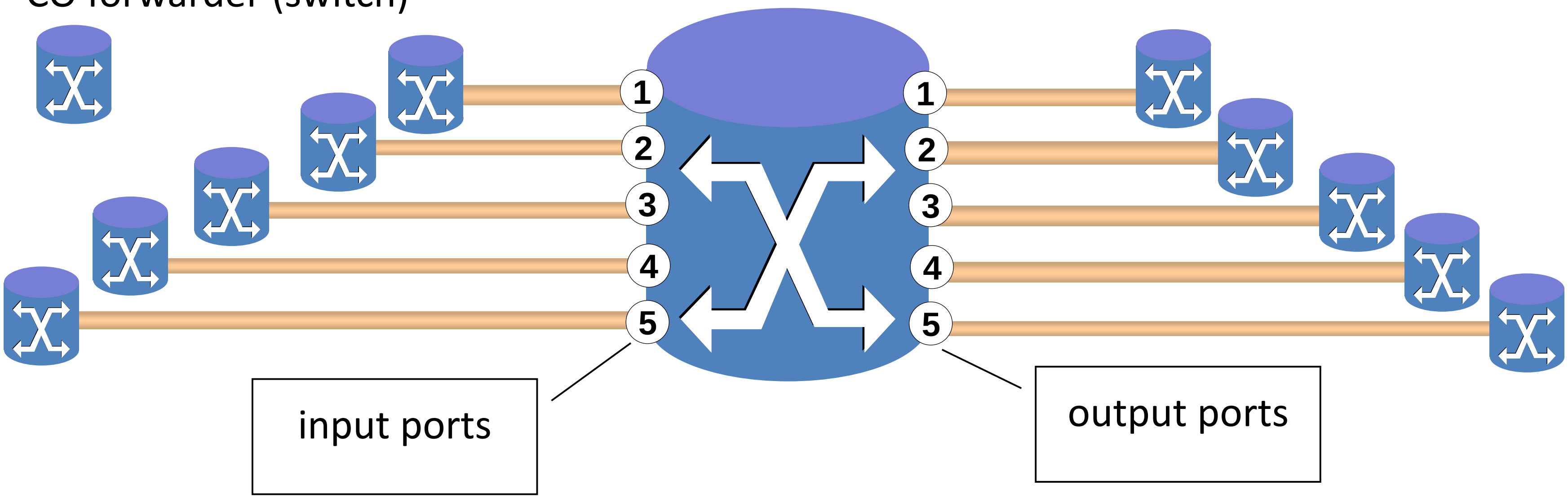
- no setup is required before sending a packet
  each router makes an independent forwarding decision
- packets are self-describing
  packet inserted anywhere will be properly forwarded
- IP forwarding detailed in RFC 1812
  hundreds of *software* cycles per packet (even with hardware switch fabric)

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*

PSN switching times

Unlike TDM/PDH/SDH switches
the residence time in a packet forwarding NE is *not* constant

The residence time includes processing time and queuing time

One important contribution is output queuing time

To estimate this time, let's assume Ethernet physical layer
with maximum packet size of 1500 Bytes
(actual networks may employ jumbo frames with 9KB frames)

The worst case queuing time for a prioritized packet is when
the packet waits for a single 1500 B packet that just started being output

- 10 Mbps 1.7 msec
- 100 Mbps : 170 μsec
- 1 Gbps : 17 μsec
- 10 Gbps 1.7 μsec

- times are lower for smaller packets
- but larger when packet needs to wait in queue
  for multiple packets to exit
- all processing times have been ignored

This is higher than TDM times for low bit-rates
but becomes negligible at high bit-rates

Traffic types

Communications between two points (point-to-point connections) may be

- unidirectional
- bidirectional

In addition to point-to-point, there may be communications

- from one point to all other points (broadcast)
- from one point to many points (point to multipoint, multicast)
- from one point to any of some set of points (anycast)
- between many points (multipoint-to-multipoint, LAN)

For PSNs there are 3 basic types of communications traffic itself

- message = 1 out of N symbols
- file = ordered (finite or infinite) sequence of messages
- stream = sequence of messages where time information is significant

Traffic can also be distinguished as

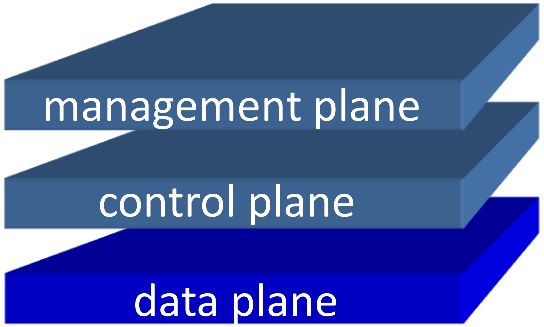
- non-real-time
- real-time non-interactive
- real-time interactive

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*

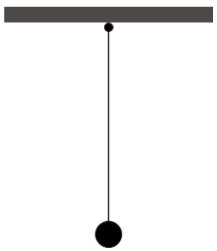
Path computation and routing

Finding an optimal path though a 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

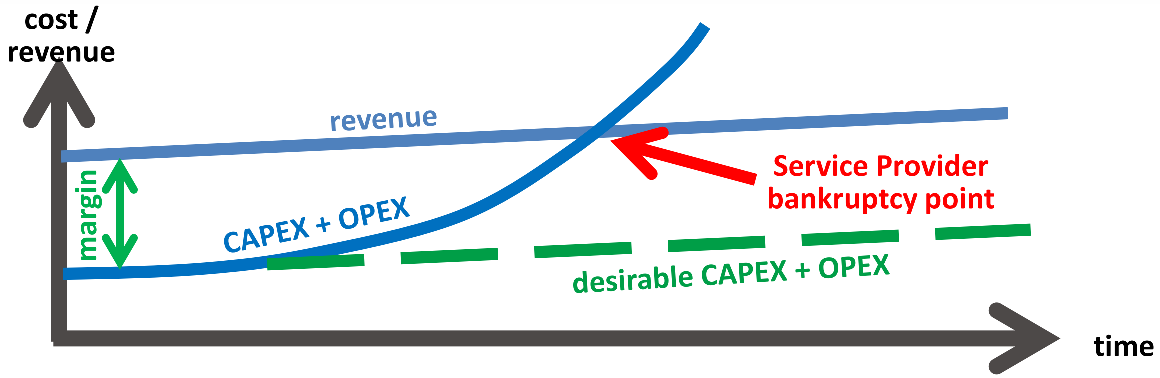
- finding simple connectivity
- minimizing number of hops

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 **P**ath **C**omputation **E**lement architecture and recently **S**oftware **D**efined **N**etworking is being studied

Research topics



- PSN SP expenses scale like data-rates, which are exponentially increasing while revenue scales like the number of subscribers, which is flat
  how can SPs avoid bankruptcy ?
- SDN theorists say that PSNs show be built like software
  - what abstractions are useful ?
  - what can be re-used ?
- A new trend is **N**etwork **F**unctions **V**irtualization
  - what network functionalities can/should be virtualized ?
  - what network functionalities can/should be relocated ?