Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Master Course
Computer Networks
IN2097

Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.

Chair for Network Architectures and Services
Institut für Informatik
Technische Universität München
http://www.net.in.tum.de

Outline

- Project announcements
- Node property fundamentals: delay, loss, throughput
- Internet Strucuture
- Network virtualisation

Project announcements

- Currently 30 teams

- SVN accounts: available by today - Monday evening, Nov 7th
- Submission 1 - Project plan - due by Tuesday evening, Nov 8th
- Submission 2 - IPv6 today - due by Tuesday evening, Nov 15th
- Submission 3 - Your own Site - due by Thursday Dec 15th

SVN Repositories

Information by Stephan Günther <guenther@net.in.tum.de>

https://projects.net.in.tum.de/svn-tum/mccn

"svn co --username <lrz_id> https://projects.net.in.tum.de/svn-tum/mccn"

The <lrz_id> is the 7-digit username that you submitted with your registration.

The password is the same one as for your MyTUM/TUMOnline account.

With the checkout you get the folders pub/ and team<xy>/ only, where <xy> denotes your team number (you don't need to know your team number in advance - checkout the repository and look at the folder name).

Note that you have

1) read access to the pub/ folder,

2) read access to the file team<xy>/grading.txt which will contain your grading for the assignment, and

3) write access to the folder team<xy>/assignment01/

Virtual Machine

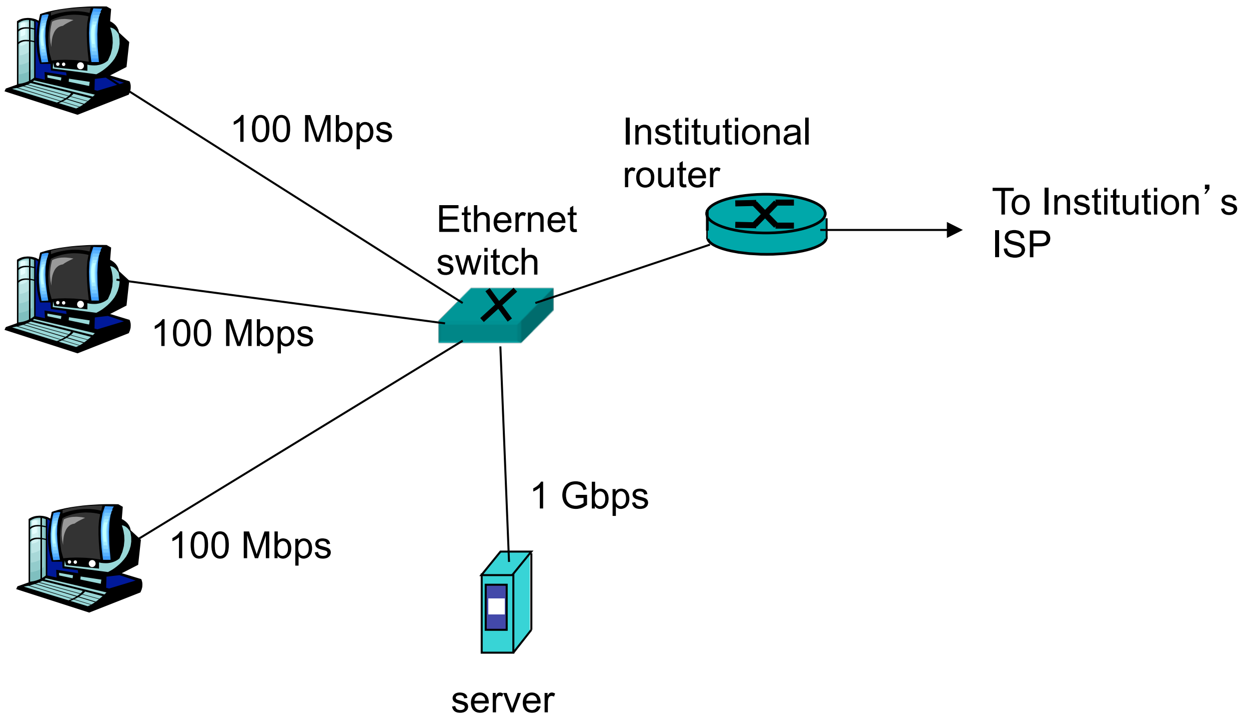
Furthermore, the folder team<xy>/ contains an RSA key pair that grants you access to your virtual machine used for the assignment. Keep the private key private (;. Further instruction on how to access the VMs are provided soon. TODO: For now, commit your project plan to a subfolder of team<xy>/assignment01/.

If you have problems accessing the SVN, please contact Stephan Günther <guenther@net.in.tum.de>

Delay, loss and throughput

Ethernet Internet access

- Typically used in companies, universities, etc
  - 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
  - Today, end systems typically connect into Ethernet switch

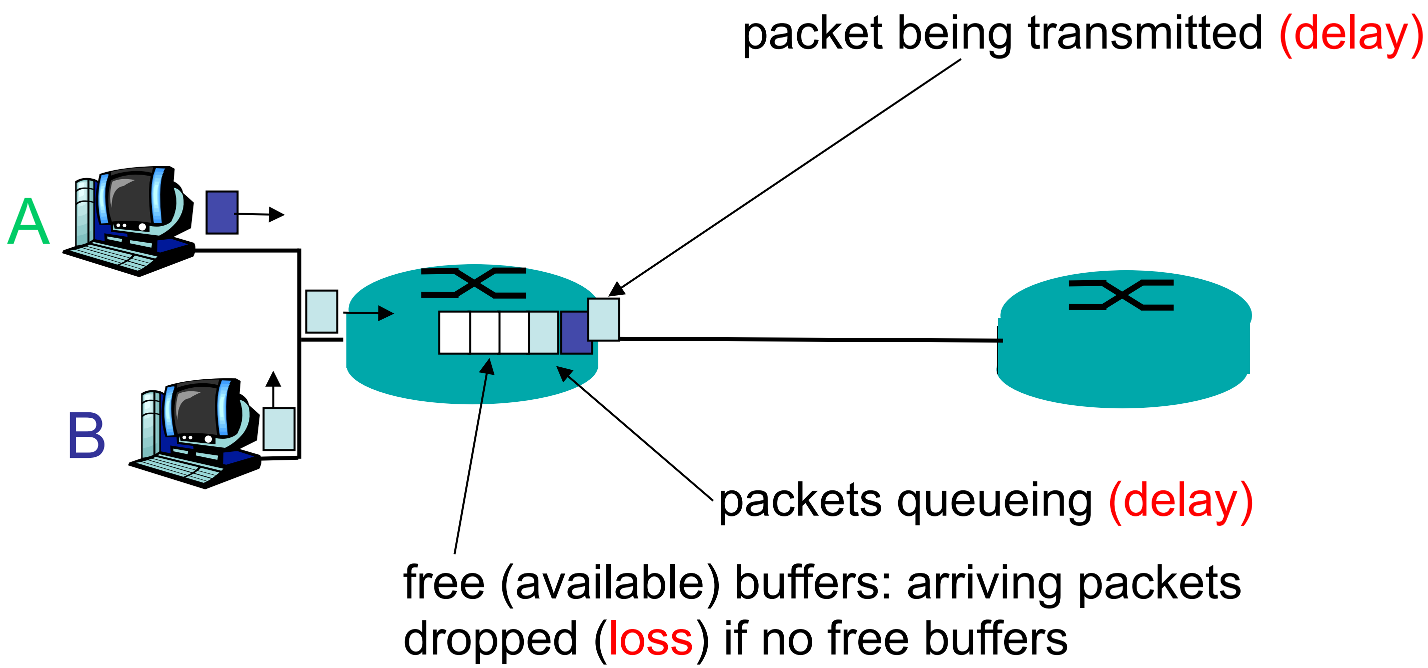


⇨ why?

Reasons for delay and loss

packets *queue* in router buffers

- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



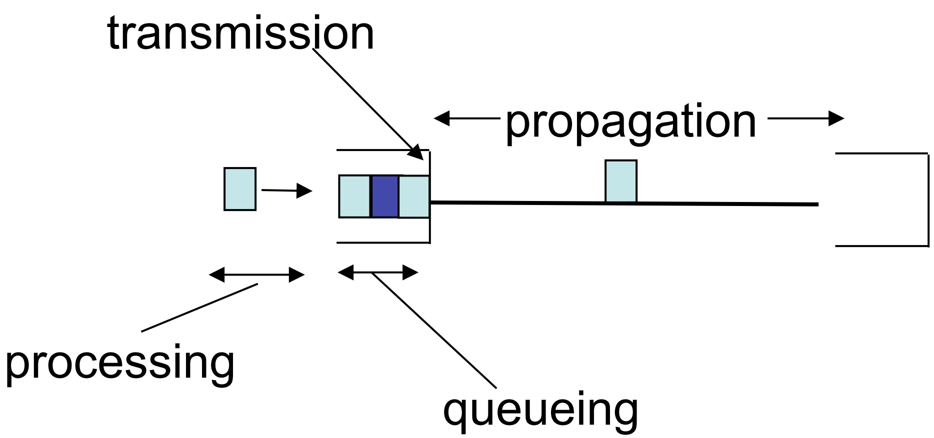
Background: Sources of packet delay

1. Processing delay:
   - Sending: prepare data for being transmitted
   - Receiving: interrupt handling

2. Queueing delay
   - time waiting at output link for transmission

3. Transmission delay:
   - L=packet length (bits)
   - R=link bandwidth (bps)
   - time to send bits into link = L/R

4. Propagation delay:
   - d = length of physical link
   - s = propagation speed in medium (~$2x10^8$ m/sec)
   - propagation delay = d/s



Nodal delay

- d_{proc} = processing delay
  - typically a few microseconds (µs) or less
- d_{queue} = queuing delay
  - depends on congestion - may be large
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - a few microseconds to hundreds of msecs

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Impact Analysis: Advances in Network Technology

Data rate	Delay (1bit)	Length (1bit)	Delay (1kbyte)	Length (1kbyte)
1 Mbit/s	1 us	200 m	8 ms	1600 km
10 Mbit/s	100 ns	20 m	0,8 ms	160 km
100 Mbit/s	10 ns	2 m	80 us	16 km
1 Gbit/s	1 ns	0,2 m	8 us	1600 m
10 Gbit/s	100 ps	0,02 m	0,8 us	160 m
100 Gbit/s	10 ps	0,002 m	80 ns	16 m

- Assessment
  - Transmission delay becomes less important
    ⇨ over time; in the core
  - Distance becomes more important
    ⇨ matters for communication beyond data center
  - Network adapter latency less important
    ⇨ Latency of communication software becomes important

Propagation Delay

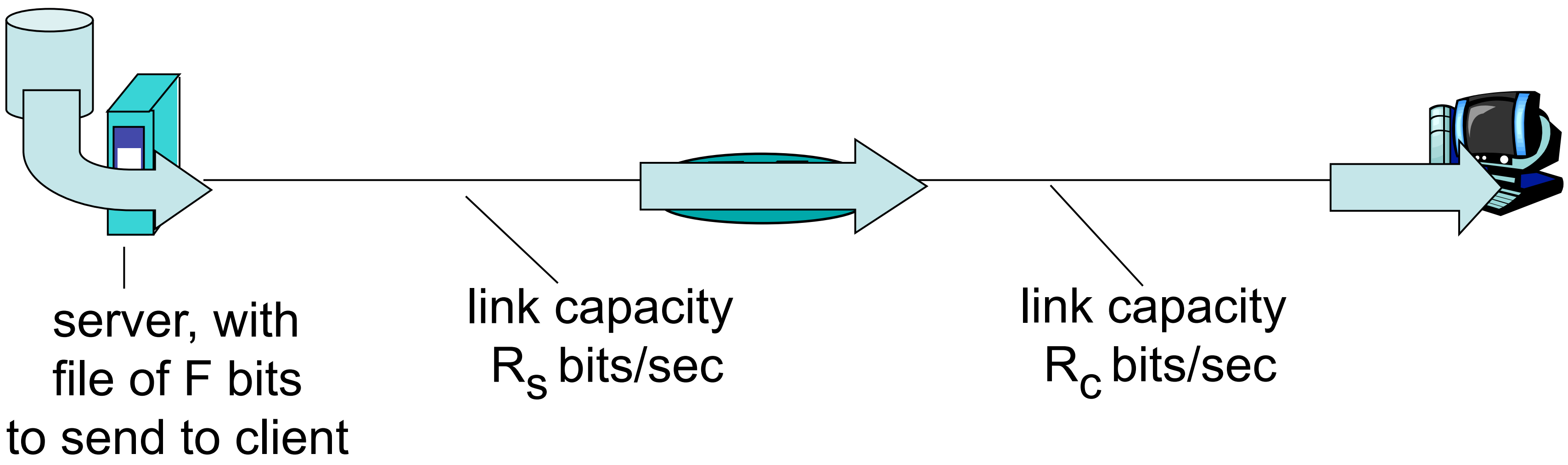
- Propagation speed: $2x10^8$ m/sec
- Transmission of 625 byte (= 5000 bit): t= L/R=5000 / 1Gbit/s = 5 us

Distance	Propagation Delay	equivalent Transmission Delay (625 byte)	CPU cycles per packet (1 GHz)	CPU cycles per byte (1 GHz)
100 m	500 ns	10 Gbit/s	500	<1
1 km	5 us	1 Gbit/s	5.000	8
10 km	50 us	100 Mbit/s	50.000	80
100 km	500 us	10 Mbit/s		800
1.000 km	5 ms	1 Mbit/s		8.000
10.000 km	50 ms	100 Kbit/s		80.000

- Suggestion for homework exercise: plot graphs

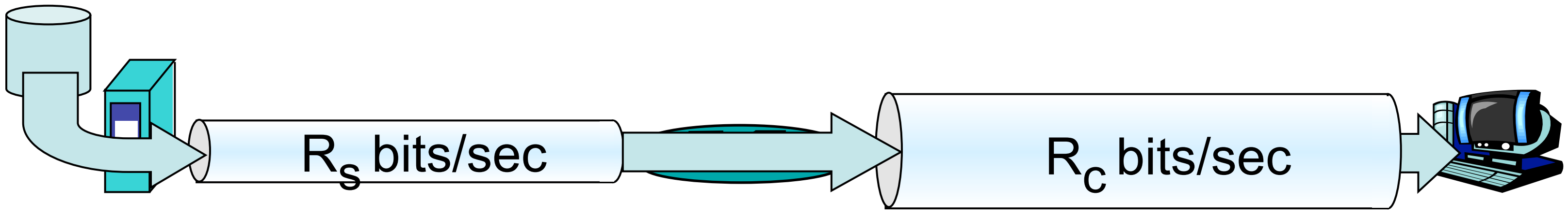
Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

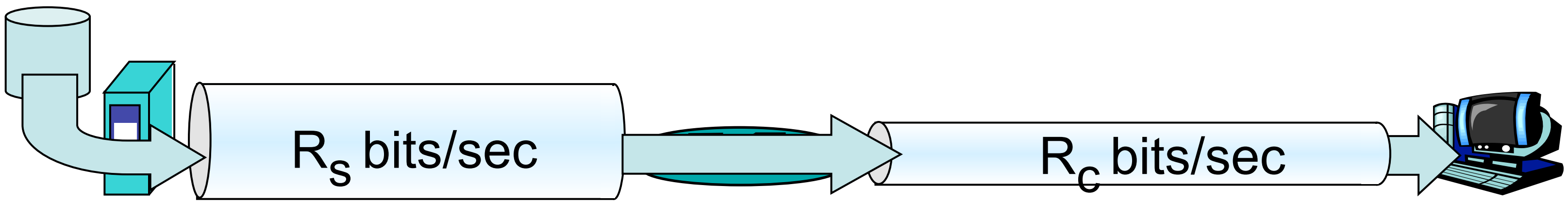


Throughput (more)

- $R_s < R_c$



- $R_s > R_c$

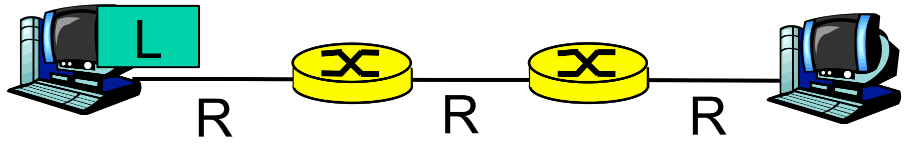


bottleneck link

link on end-end path that constrains end-end throughput

⇨ measurement challenge for networks with many nodes:
identify bottleneck interfaces, e.g. with packet-pair measurements

Store-and-Forward vs. Circuit Switching



- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: store and forward
- delay = 3L/R

Example: Large Message L

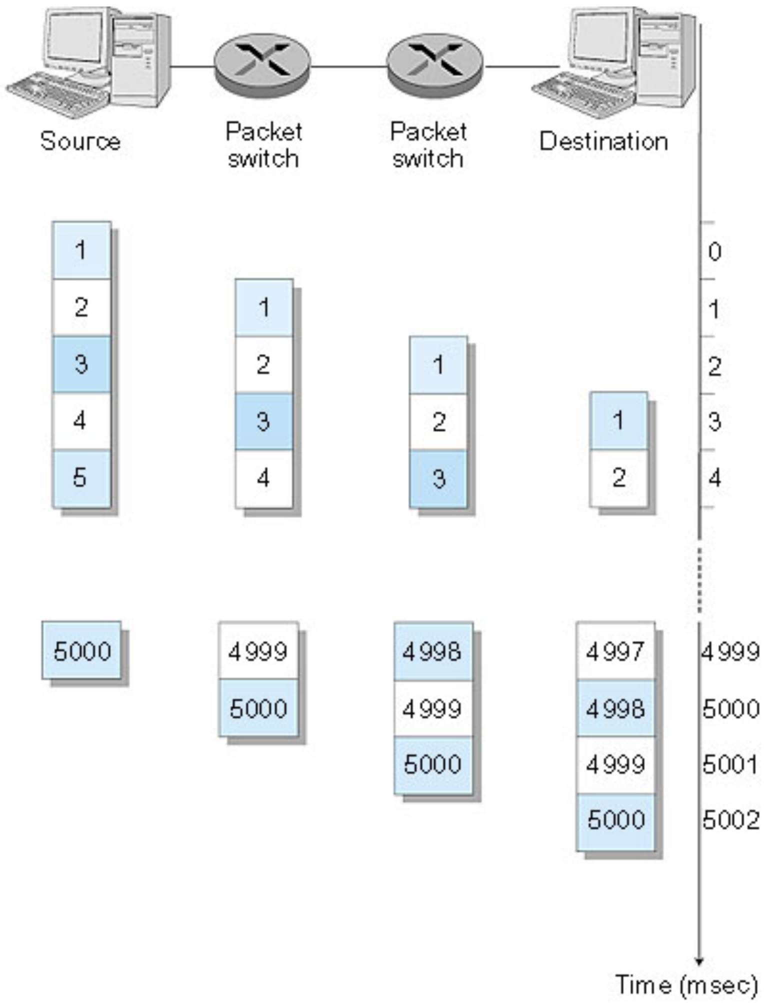
Circuit Switching:

- L = 7.5 Mbit
- R = 1.5 Mbit/s
- Transmission delay = 5 s

Store-and-Forward:

- L = 7.5 Mbit
- R = 1.5 Mbit/s
- Transmission delay = 15 s

Packet Switching: Message Segmenting



Now break up the message into 5000 packets

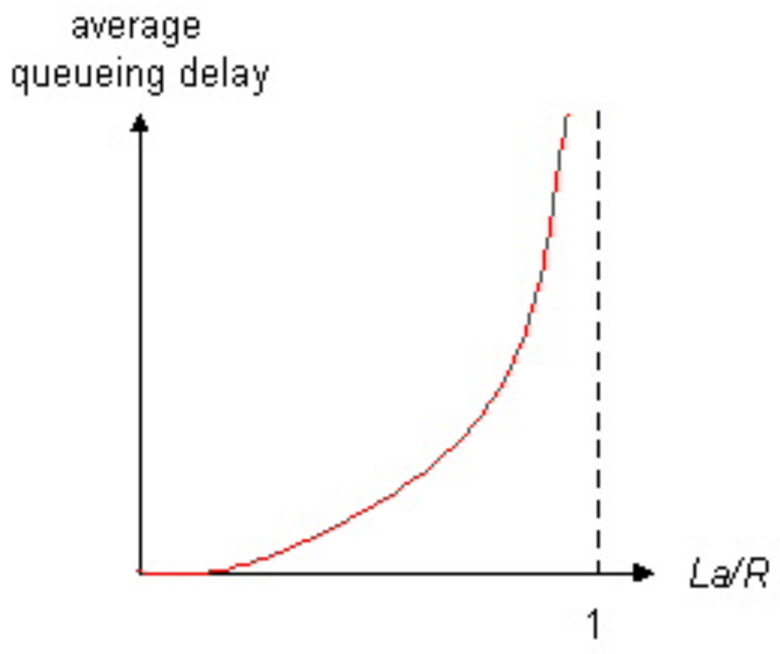
- Each packet 1,500 bits
- 1 msec to transmit packet on one link
- *pipelining:* each link works in parallel
- Delay reduced from 15 sec to 5.002 sec (as good as circuit switched)
- Advantages over circuit switching?
- Drawbacks (of packet vs. Message)

Queueing delay (revisited)

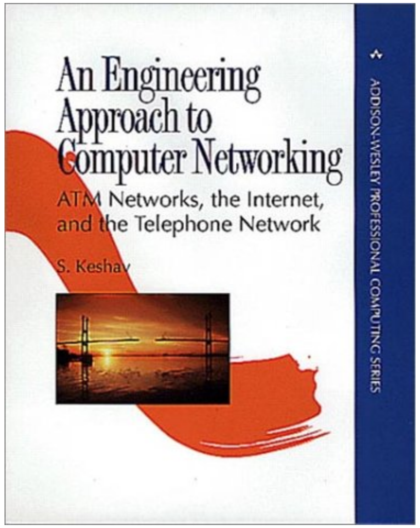
- R=link bandwidth (bit/s)
- L=packet length (bit)
- a=average packet arrival rate

traffic intensity = a·L/R

- a·L/R ~ 0: average queuing delay small
- a·L/R → 1: delays become large
- a·L/R > 1: more "work" arriving than can be serviced, average delay infinite!

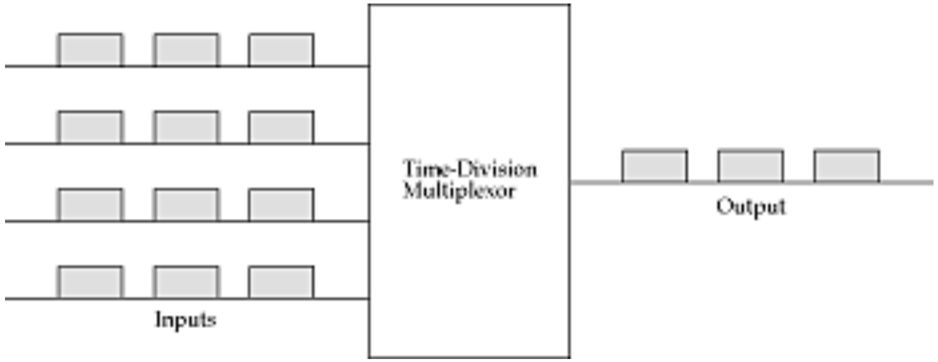


- S. Keshav: *An Engineering Approach to Computer Networking*. Addison-Wesley, 1999



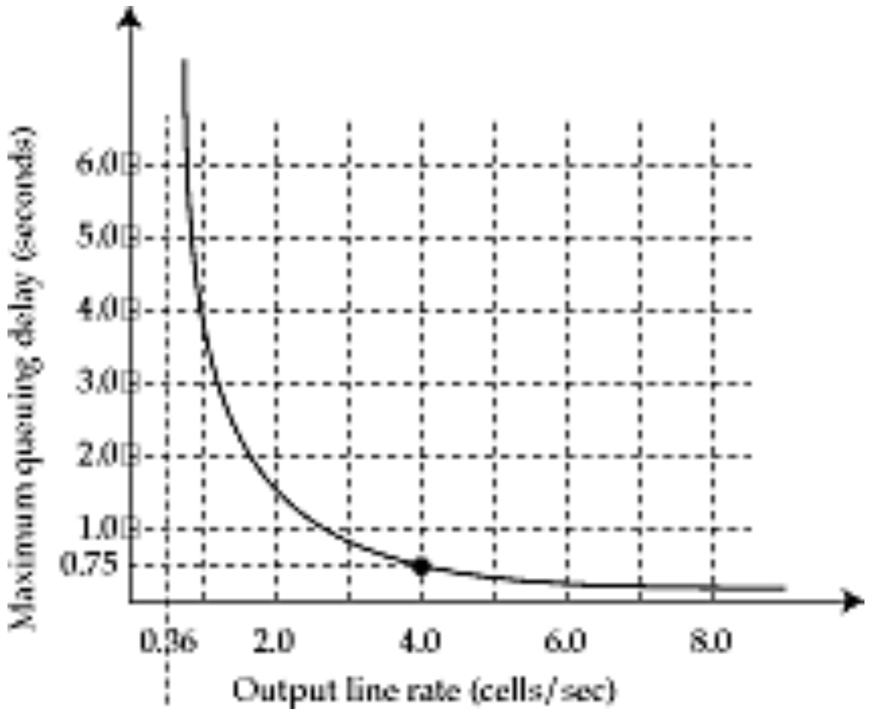
- Srinivasan Keshav - University of Waterloo

Statistical multiplexing



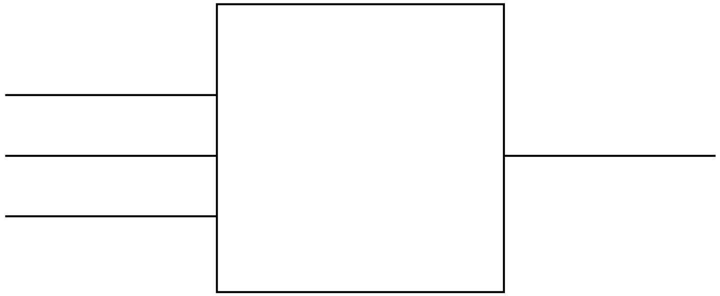
- Suppose packets/cells arrive in bursts
  - each burst has 10 packets/cells evenly spaced 1 second apart
  - gap between bursts = 100 seconds
- What should be service rate of output line?

Statistical Multiplexing



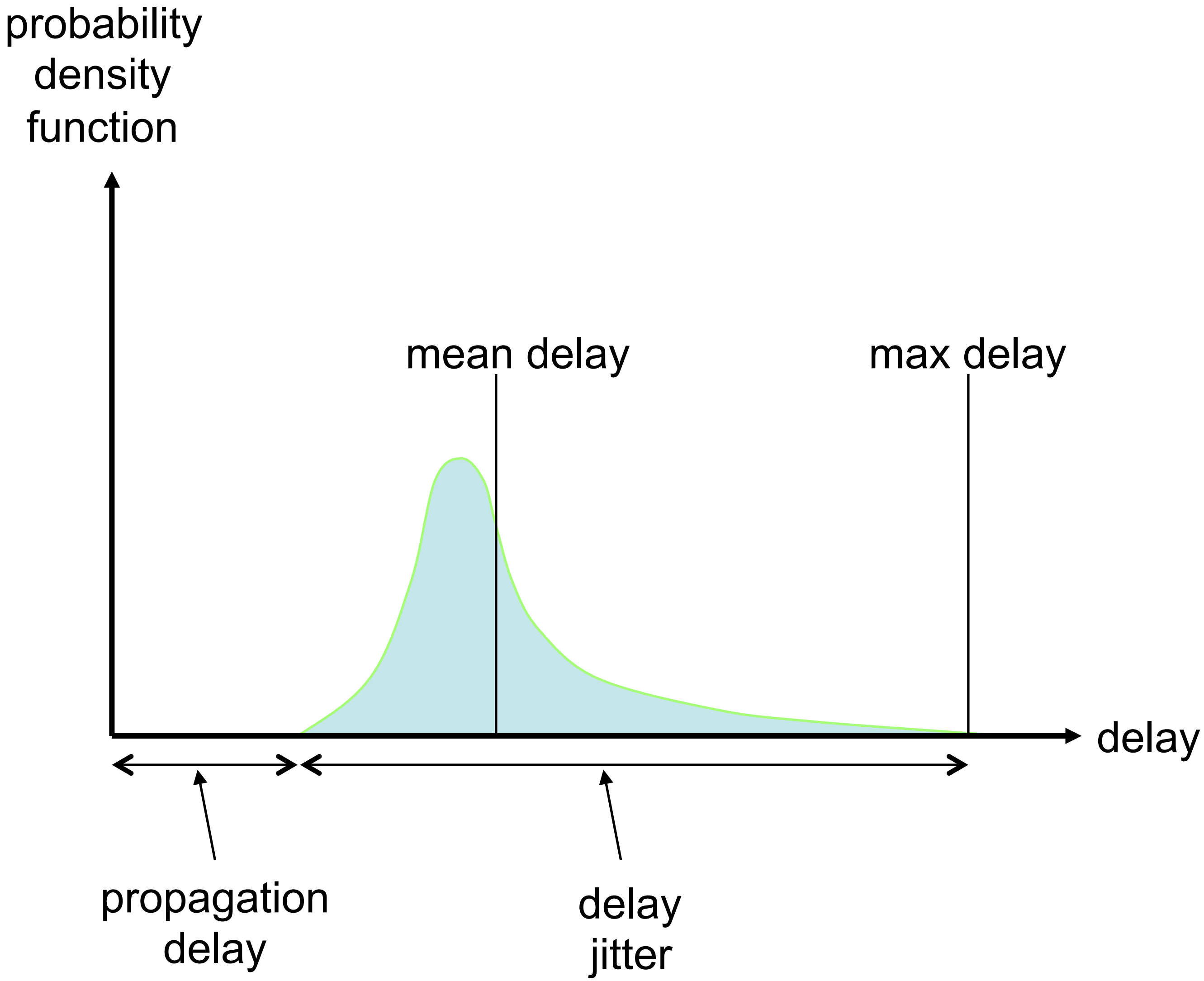
- We can trade off worst-case delay against speed of output trunk
- Statistical Multiplexing Gain
  = (sum of peak input rate)/(output rate)
- Whenever long term average rate differs from peak, we can trade off service rate for delay

Statistical Multiplexing



- Packets with L=625 byte; R=100 Mbit/s ⇒ d_{trans} = L/R = 50 us
- Input link: average load = 10%, i.e. a_{in} = 0.1
- Output link 200 Mbit/s: average load out = 15%, i.e. a_{out} = 0.15
  ⇒ d_{queue_max} = 2x 25 us = 50 us
- Output link 100 Mbit/s: average load = 30%, i.e. a_{out} = 0.3
  ⇒ d_{queue_max} = 2x 50 us = 100 us
- Output link 50 Mbit/s: average load = 60%, i.e. a_{out} = 0.6
  ⇒ d_{queue_max} = 2x 100 us = 200 us

Delay Distributions



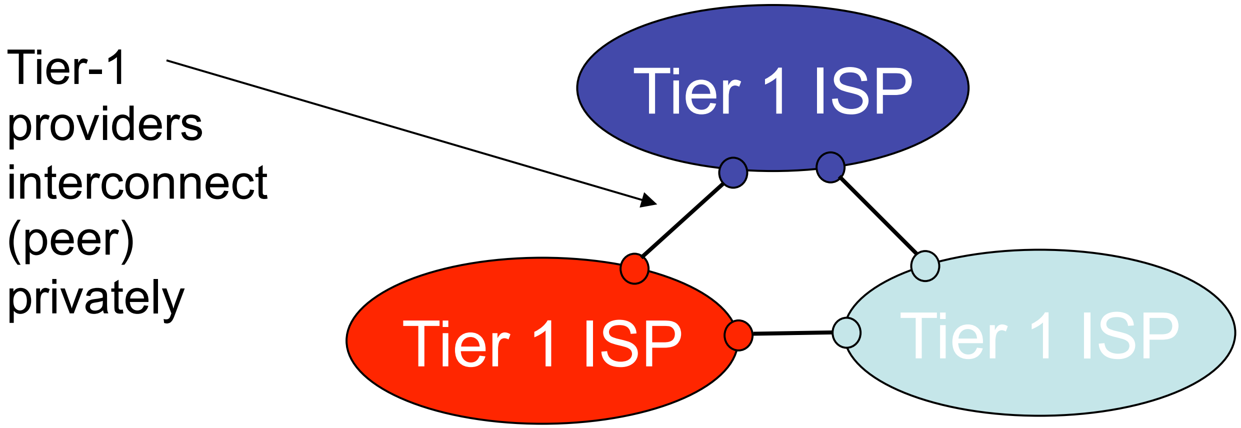
Discussion

- Can you „imagine“ a visualisation of packets being transmitted over different types of links?
- What is the role of statistical multiplexing
- What are the benefits of overprovisioning?
- What is the cost of tunneling?
- What is the role of header lengths?
- What is the role of compact headers / header compression?

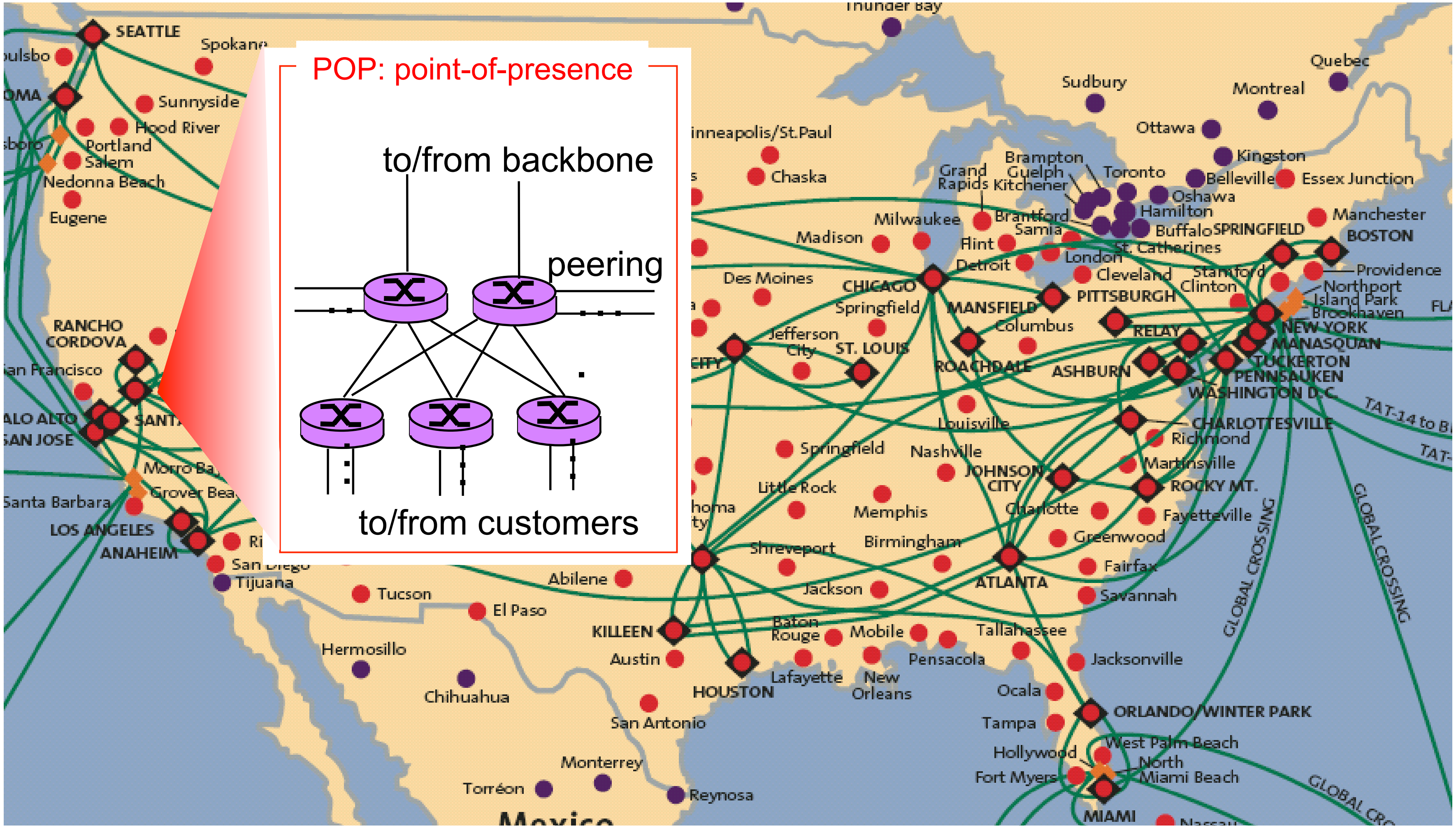
Internet Structure

Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (AT&T, Global Crossing, Level 3, NTT, Qwest, Sprint, Tata, Verizon (UUNET), Savvis, TeliaSonera), national/international coverage
  - treat each other as equals
  - can reach every other network on the Internet without purchasing IP transit or paying settlements



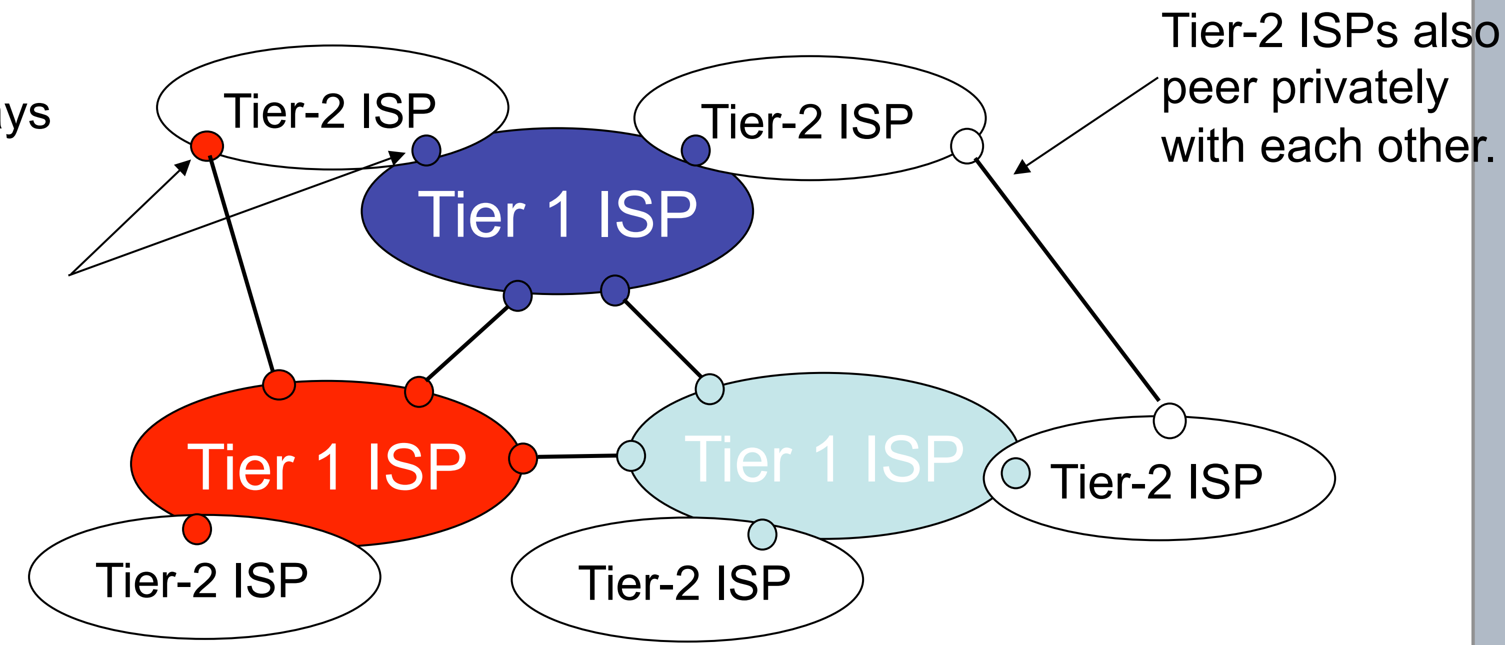
Tier-1 ISP: e.g., Sprint



Internet structure: network of networks

- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

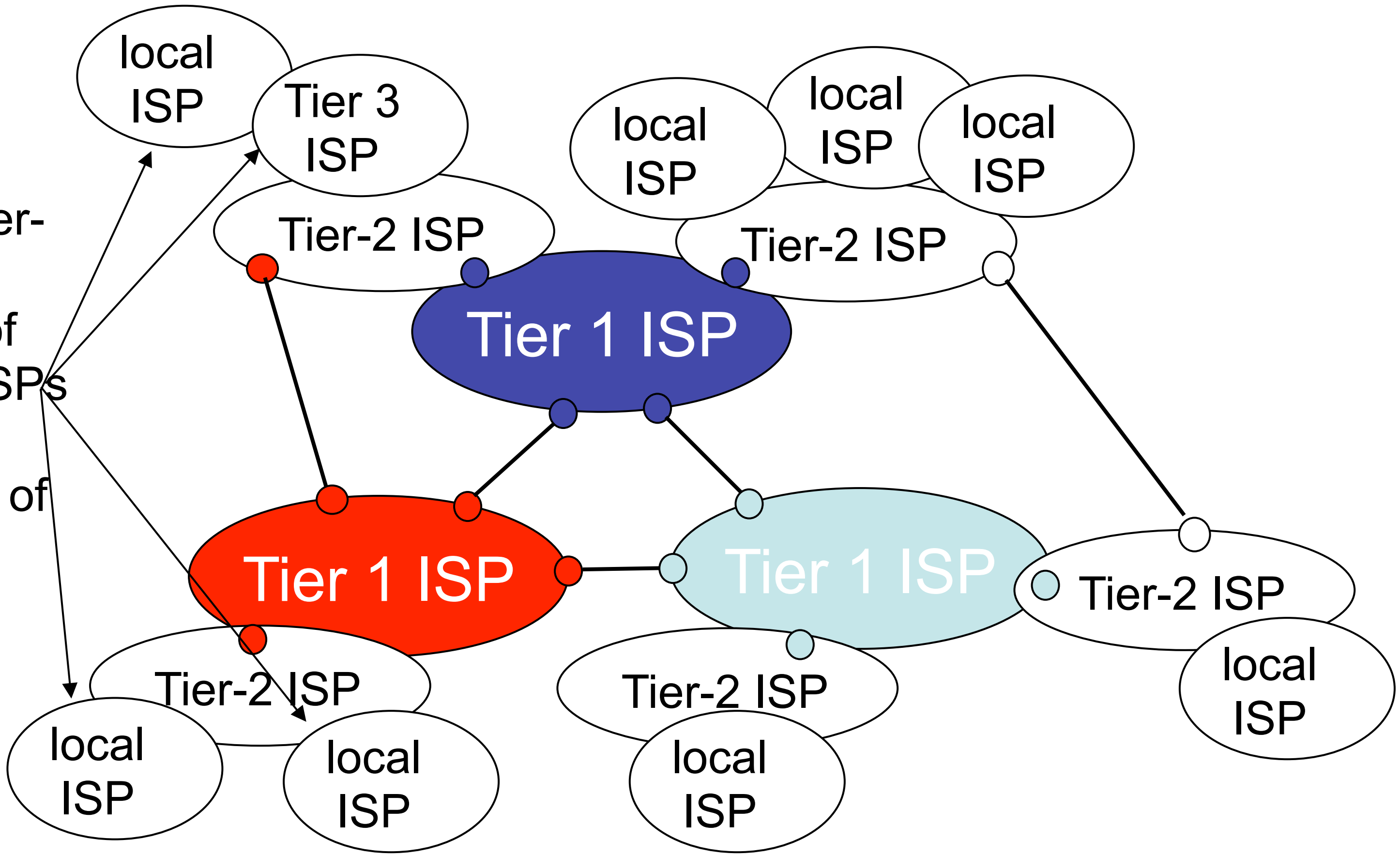
- Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet
- tier-2 ISP is *customer* of tier-1 provider



Internet structure: network of networks

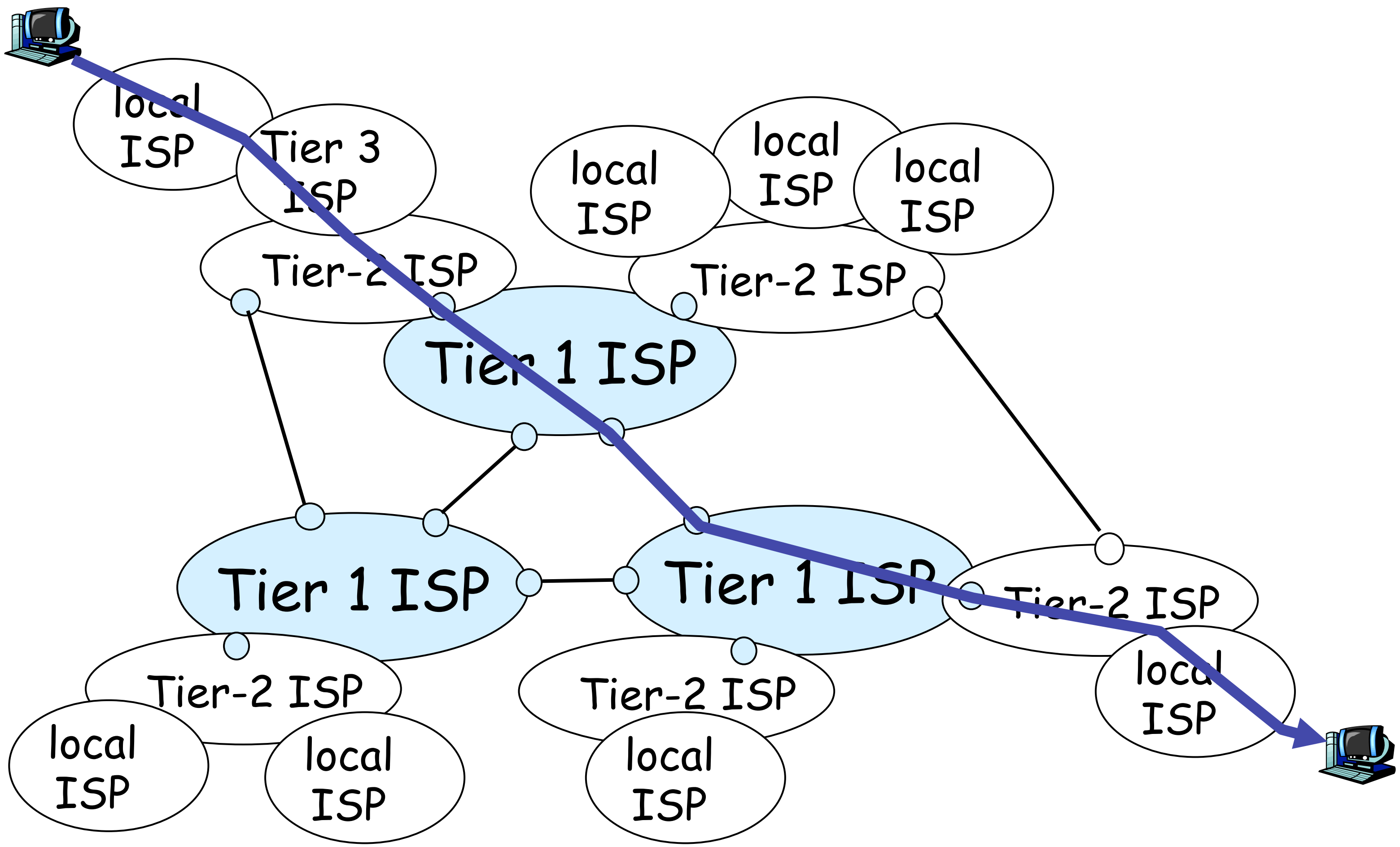
- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)

Local and tier-3 ISPs are *customers* of higher tier ISPs connecting them to rest of Internet

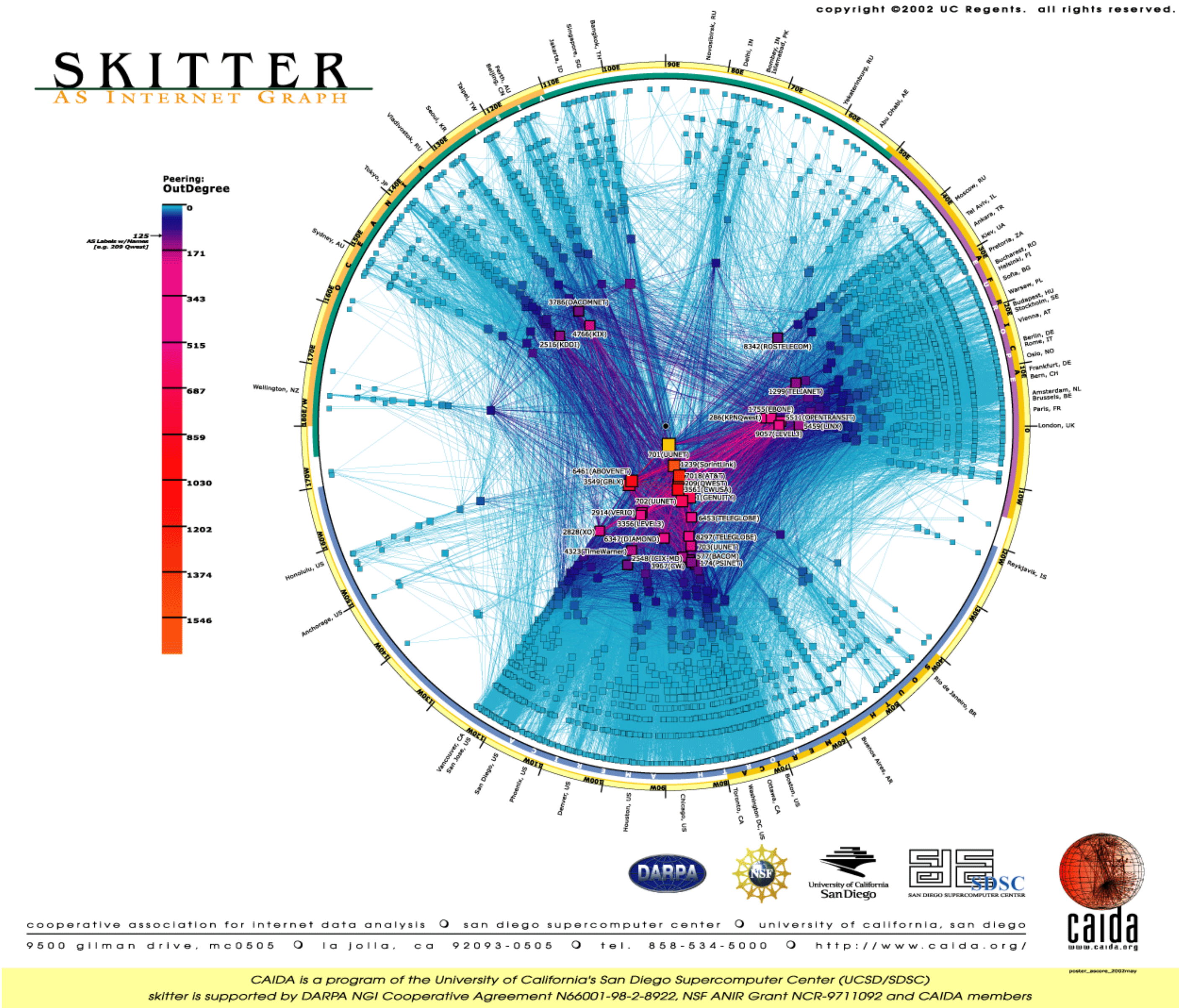


Internet structure: network of networks

- a packet passes through many networks!



ISP Peering Relations



Virtual Private Networks

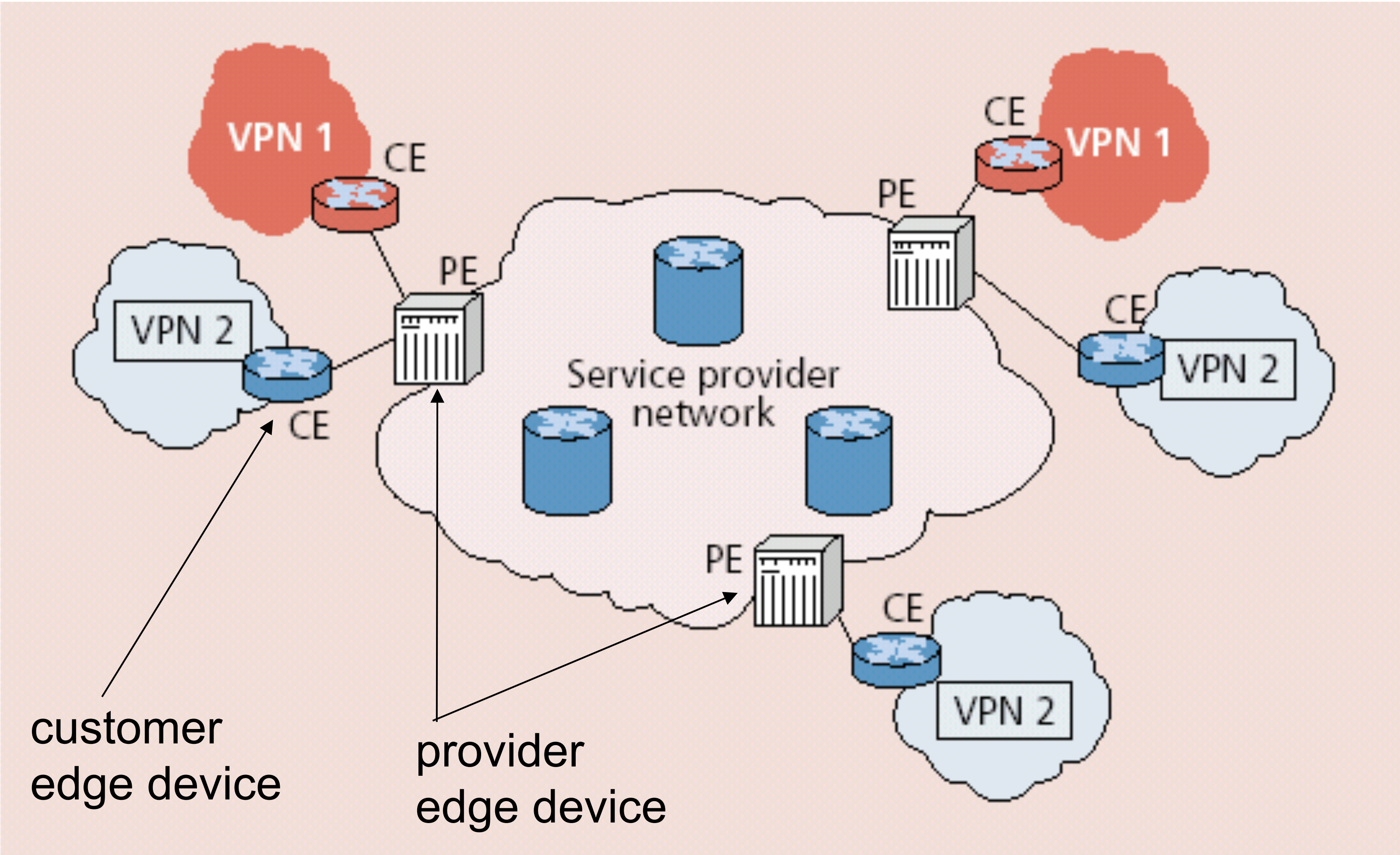
Virtual Private Networks (VPN)

VPNs

Networks perceived as being private networks by customers using them, but built over shared infrastructure owned by service provider (SP)

- Service provider infrastructure:
  - backbone
  - provider edge devices
- Customer:
  - customer edge devices (communicating over shared backbone)

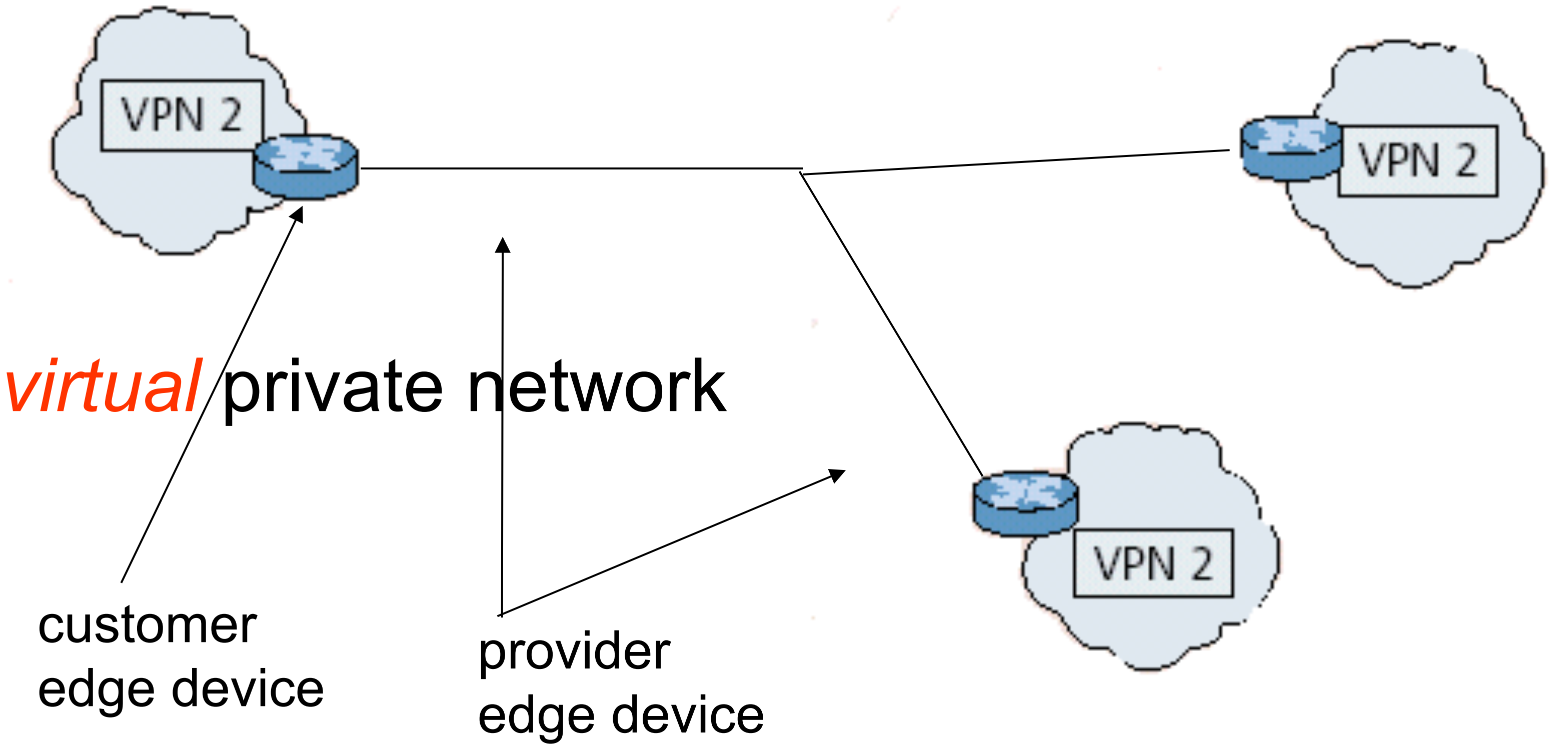
VPN reference architecture



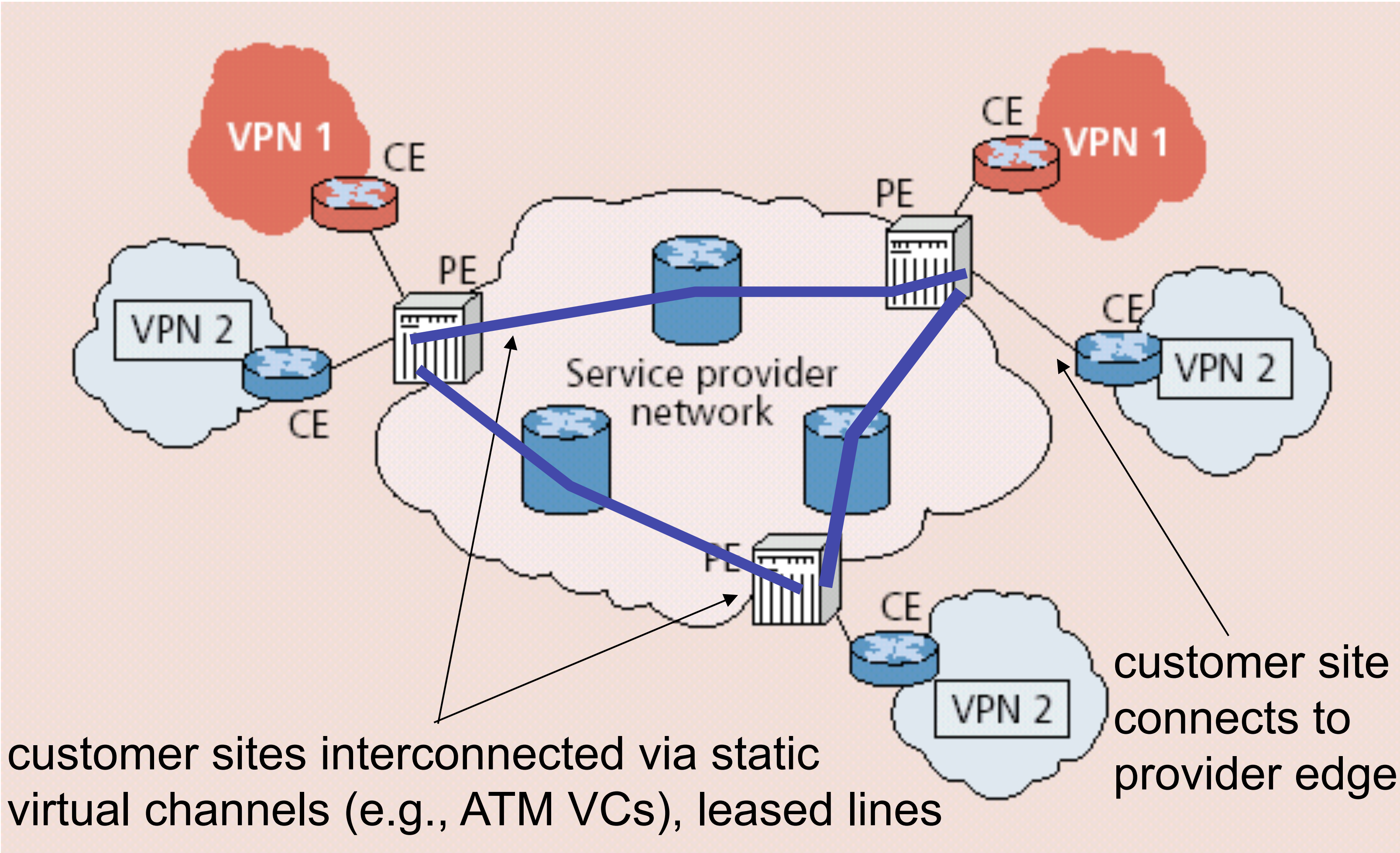
VPNs: why?

- Privacy
- Security
- Works well with mobility (looks like you are always at home)
- Cost: many forms of newer VPNs are cheaper than leased line VPNs
  - ability to share at lower layers even though logically separate means lower cost
  - exploit multiple paths, redundacy, fault-recovery in lower layers
  - need isolation mechanisms to ensure resources shared appropriately
- Abstraction and manageability: all machines with addresses that are "in" are trusted no matter where they are

VPN: logical view

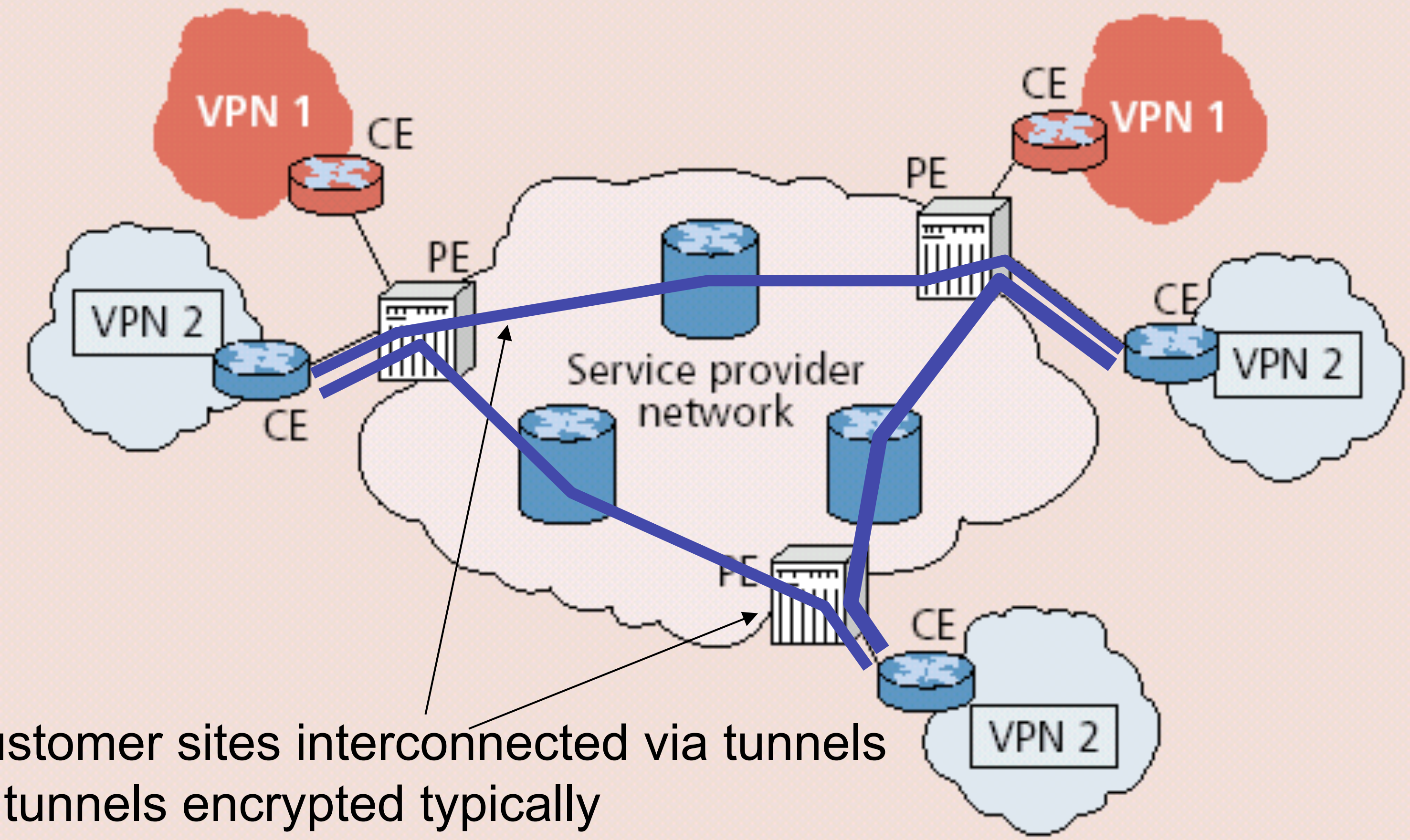


Leased-line VPN



Customer premise VPN

- all VPN functions implemented by customer



customer sites interconnected via tunnels

- tunnels encrypted typically
- SP treats VPN packets like all other packets

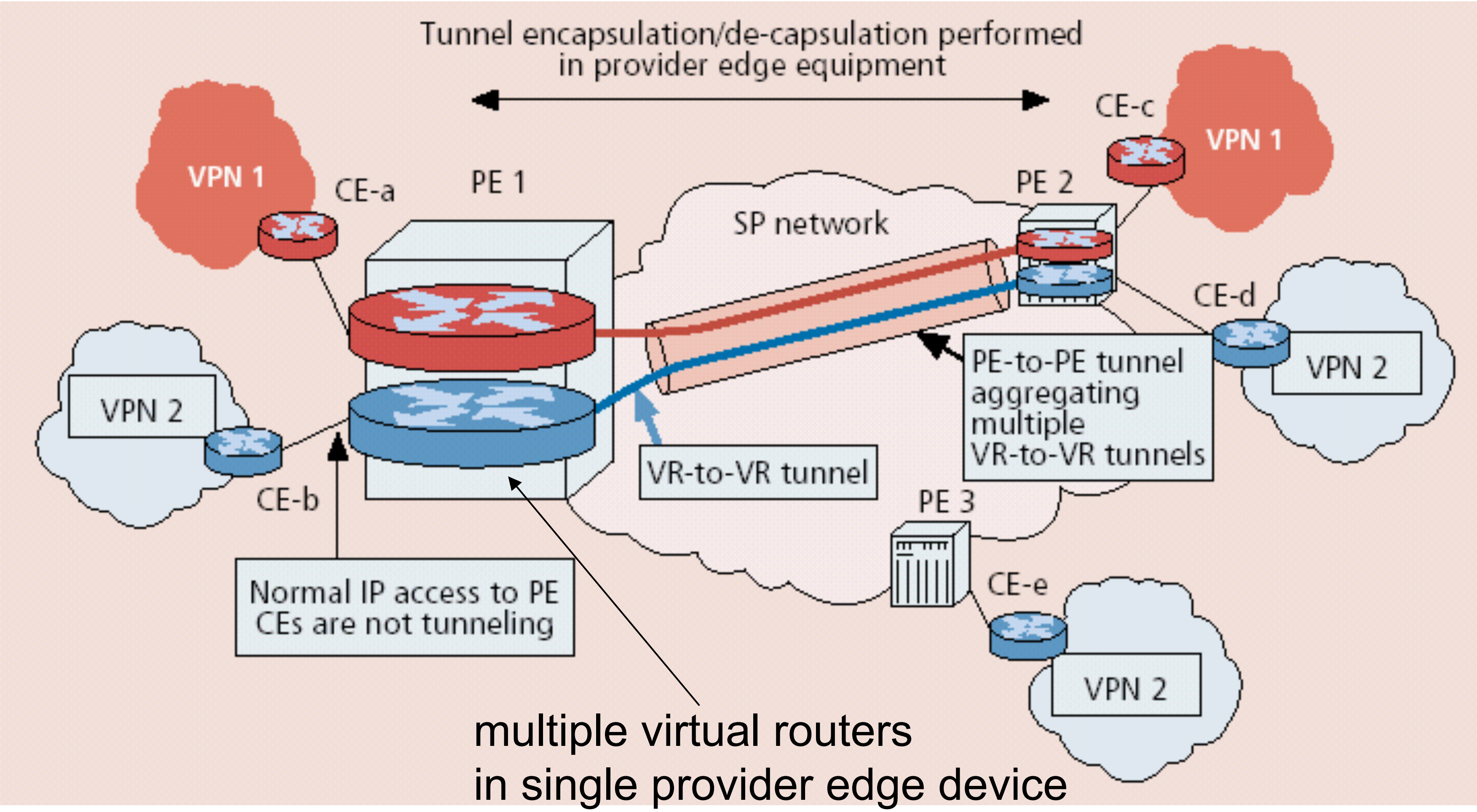
Drawbacks

- Leased-line VPN: configuration costs, maintainance by SP: long time, much manpower
- CPE-based VPN: expertise by customer to acquire, configure, manage VPN

Network-based VPN

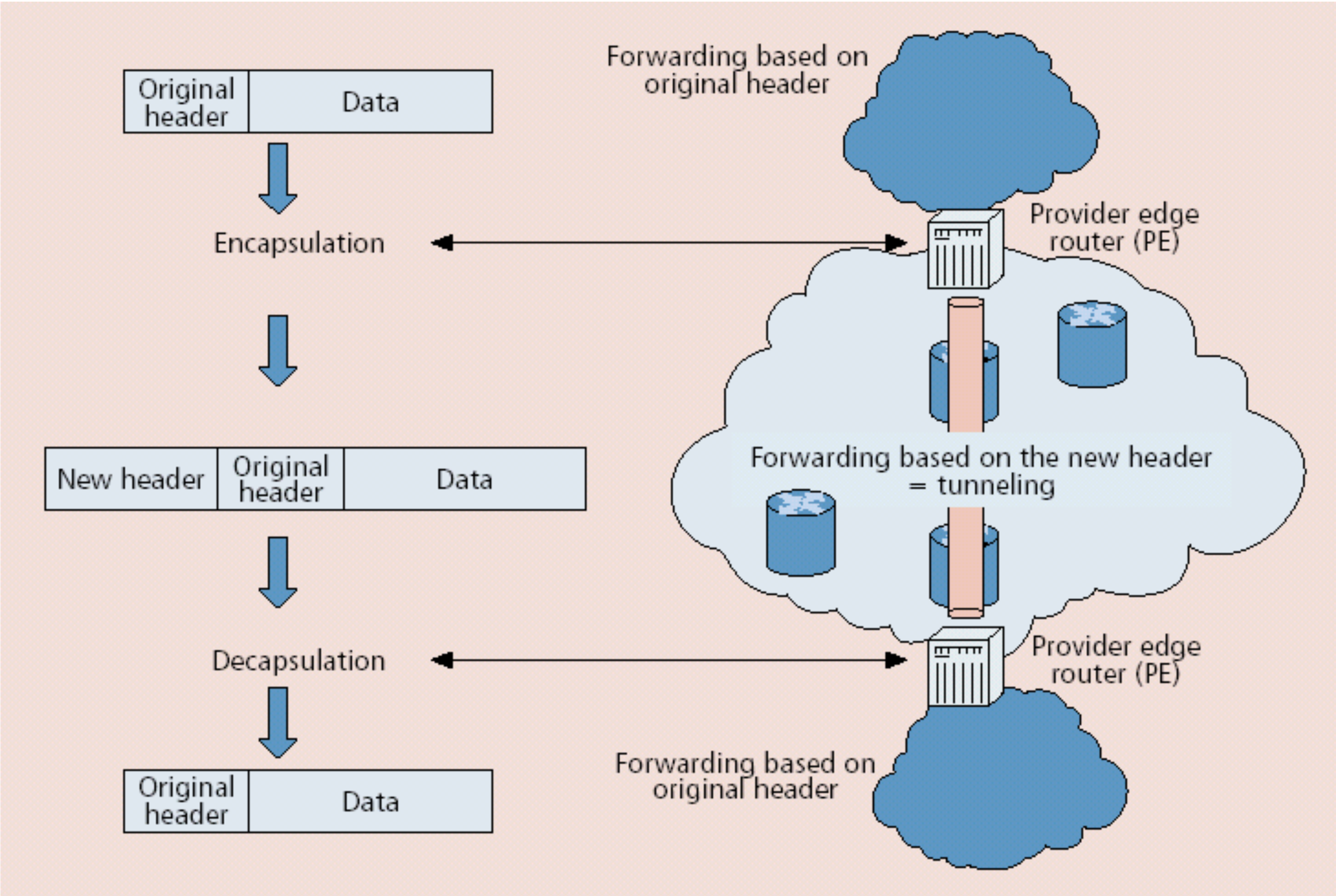
- Customer’s routers connect to SP routers
- SP routers maintain separate (independent) IP contexts for each VPN
  - sites can use private addressing
  - traffic from one VPN cannot be injected into another

Network-based Layer 3 VPNs



multiple virtual routers
in single provider edge device

Tunneling



Original header
Data
Encapsulation
New header
Original header
Data
Decapsulation
Original header
Data
Forwarding based on original header
Provider edge router (PE)
Forwarding based on the new header = tunneling
Provider edge router (PE)
Forwarding based on original header

Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Network Architectures

Virtual Circuit and Datagram Networks

Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Network Architectures

Link virtualization: ATM, MPLS

Technische Universität München

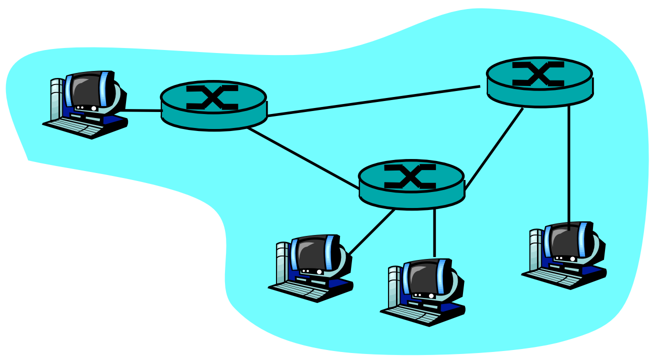
Virtualization of networks

- Virtualization of resources: powerful abstraction in systems engineering:
- computing examples: virtual memory, virtual devices
  - Virtual machines: e.g., java
  - IBM VM operation system from 1960's/70's
- layering of abstractions: don't sweat the details of the lower layer, only deal with lower layers abstractly

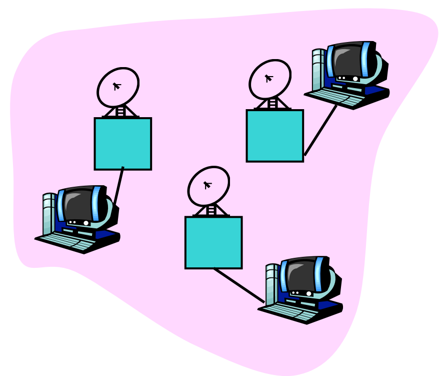
The Internet: virtualizing networks

- 1974: multiple unconnected nets
  - ARPAnet
  - data-over-cable networks
  - packet satellite network (Aloha protocol)
  - packet radio network

… differing in:
  - addressing conventions
  - packet formats
  - error recovery
  - routing



ARPAnet



satellite net

"A Protocol for Packet Network Intercommunication",
V. Cerf, R. Kahn, IEEE Transactions on Communications,
May, 1974, pp. 637-648.

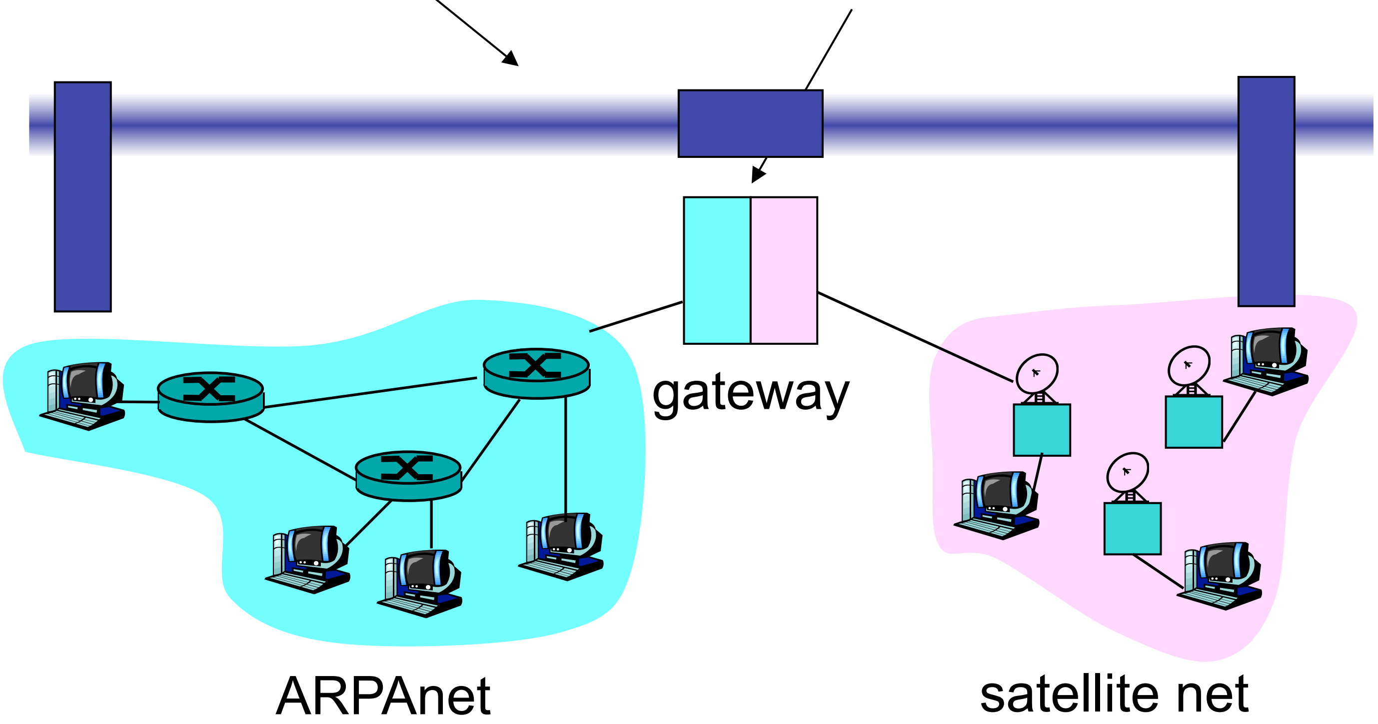
The Internet: virtualizing networks

Internetwork layer (IP):

- addressing: internetwork appears as single, uniform entity, despite underlying local network heterogeneity
- network of networks

Gateway:

- "embed internetwork packets in local packet format or extract them"
- route (at internetwork level) to next gateway



Cerf & Kahn's Internetwork Architecture

- What is virtualized?
- two layers of addressing: internetwork and local network
- new layer (IP) makes everything homogeneous at internetwork layer
- underlying local network technology
  - cable
  - satellite
  - 56K telephone modem
  - today: ATM, MPLS
- … "invisible" at internetwork layer. Looks like a link layer technology to IP!

ATM and MPLS

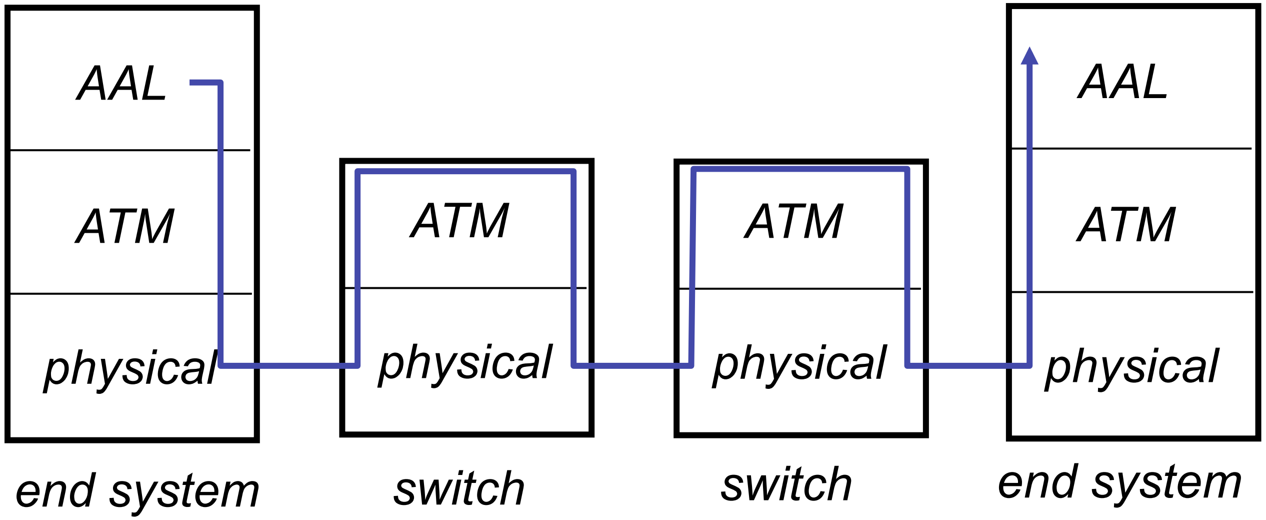
- ATM, MPLS separate networks in their own right
  - different service models, addressing, routing from Internet
- viewed by Internet as logical link connecting IP routers
  - just like dialup link is really part of separate network (telephone network)
- ATM, MPLS: of technical interest in their own right

Asynchronous Transfer Mode: ATM

- **1990's/00 standard for high-speed** (155Mbps to 622 Mbps and higher) *Broadband Integrated Service Digital Network* architecture

- Goal: *integrated, end-end transport of carry voice, video, data*
  - meeting timing/QoS requirements of voice, video (versus Internet best-effort model)
  - "next generation" telephony: technical roots in telephone world
  - packet-switching (fixed length packets, called "cells") using virtual circuits

ATM architecture



- adaptation layer: only at edge of ATM network
  - data segmentation/reassembly
  - roughly analagous to Internet transport layer
- ATM layer: “network” layer
  - cell switching, routing
- physical layer

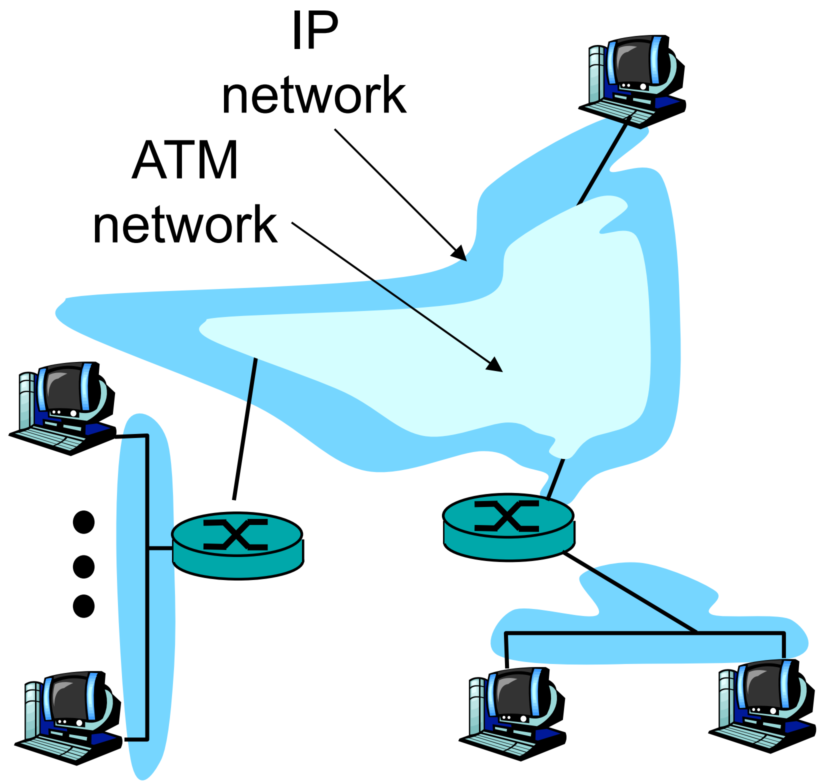
ATM: network or link layer?

Vision: end-to-end transport: "ATM from desktop to desktop"

- ATM *is* a network technology

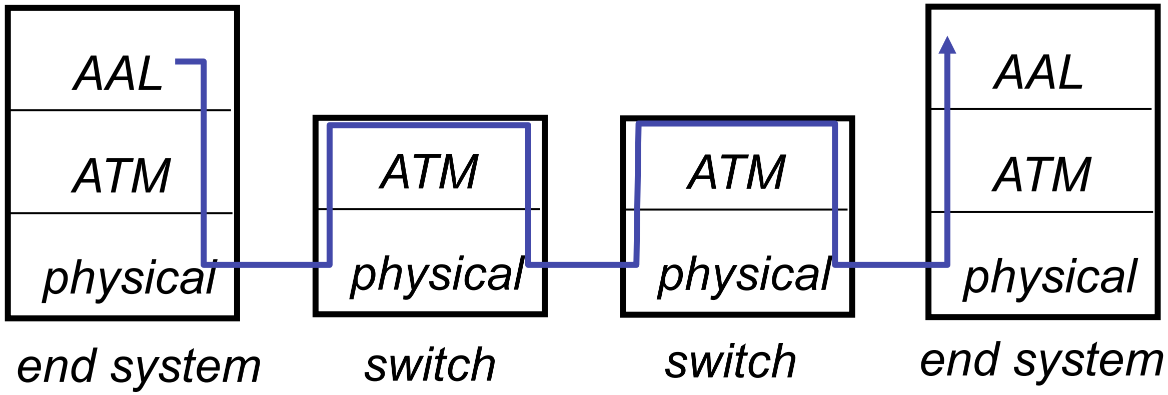
Reality: used to connect IP backbone routers

- "IP over ATM"
- ATM as switched link layer, connecting IP routers



ATM Adaptation Layer (AAL)

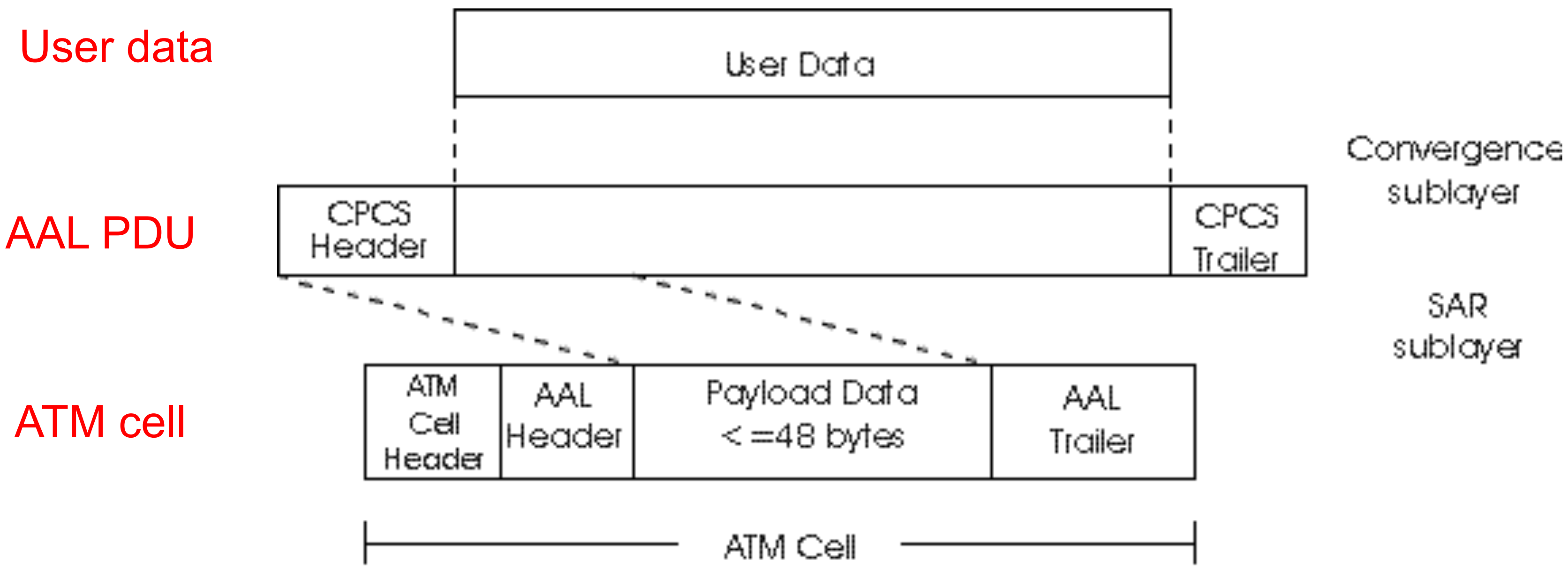
- ATM **Adaptation Layer** (AAL): "adapts" upper layers (IP or native ATM applications) to ATM layer below
- AAL present **only in end systems**, not in switches
- AAL layer segment (header/trailer fields, data) fragmented across multiple ATM cells
  - analogy: TCP segment in many IP packets



ATM Adaptation Layer (AAL) [more]

Different versions of AAL layers, depending on ATM service class:

- AAL1: for CBR (Constant Bit Rate) services, e.g. circuit emulation
- AAL2: for VBR (Variable Bit Rate) services, e.g., MPEG video
- AAL5: for data (eg, IP datagrams)



ATM Layer

Service: transport cells across ATM network

- analogous to IP network layer
- very different services than IP network layer

Network Architecture	Service Model	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

ATM Layer: Virtual Circuits

- VC transport: cells carried on VC from source to dest
  - call setup, teardown for each call *before* data can flow
  - each packet carries VC identifier (not destination ID)
  - *every* switch on source-dest path maintain “state” for each passing connection
  - link,switch resources (bandwidth, buffers) may be *allocated* to VC: to get circuit-like perf.
- Permanent VCs (PVCs)
  - long lasting connections
  - typically: “permanent” route between to IP routers
- Switched VCs (SVC):
  - dynamically set up on per-call basis

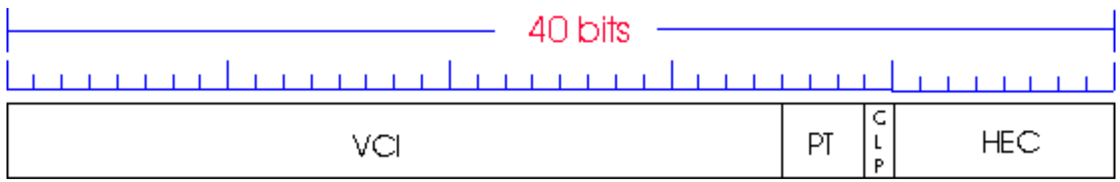
ATM VCs

- Advantages of ATM VC approach:
  - QoS performance guarantee for connection mapped to VC (bandwidth, delay, delay jitter)
- Drawbacks of ATM VC approach:
  - Inefficient support of datagram traffic
  - one PVC between each source/dest pair) does not scale (N*2 connections needed)
  - SVC introduces call setup latency, processing overhead for short lived connections

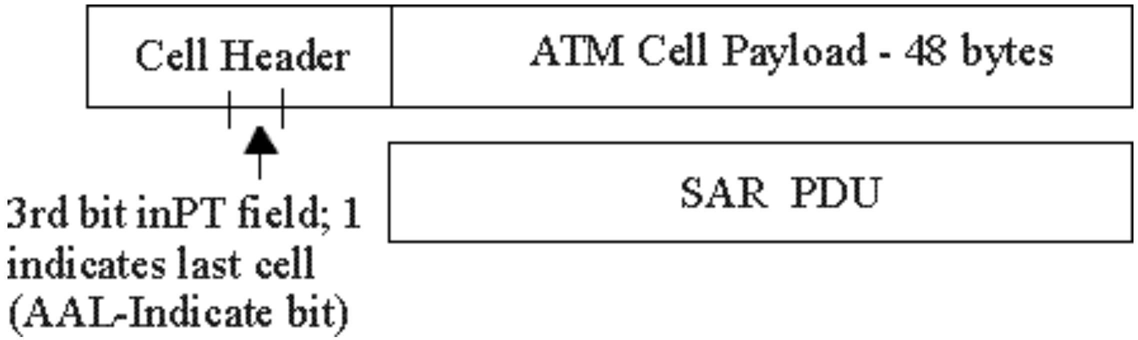
ATM Layer: ATM cell

- 5-byte ATM cell header
- 48-byte payload
  - Why?: small payload -> short cell-creation delay for digitized voice
  - halfway between 32 and 64 (compromise!)

Cell header

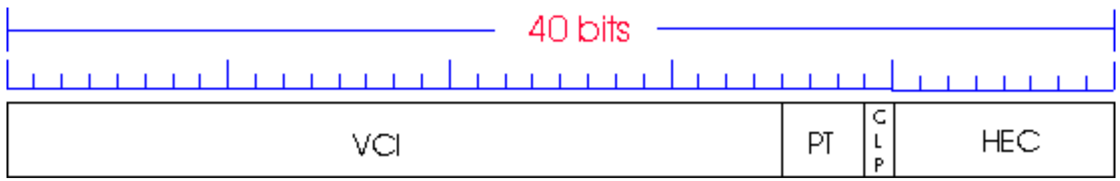


Cell format



ATM cell header

- **VCI:** virtual channel ID
  - will *change* from link to link thru net
- **PT:** Payload type (e.g. RM cell versus data cell)
- **CLP:** Cell Loss Priority bit
  - CLP = 1 implies low priority cell, can be discarded if congestion
- **HEC:** Header Error Checksum
  - cyclic redundancy check



Datagram or VC network: why?

Internet

- data exchange among computers
  - "elastic" service, no strict timing requirements
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at "edge"
- many link types
  - different characteristics
  - uniform service difficult

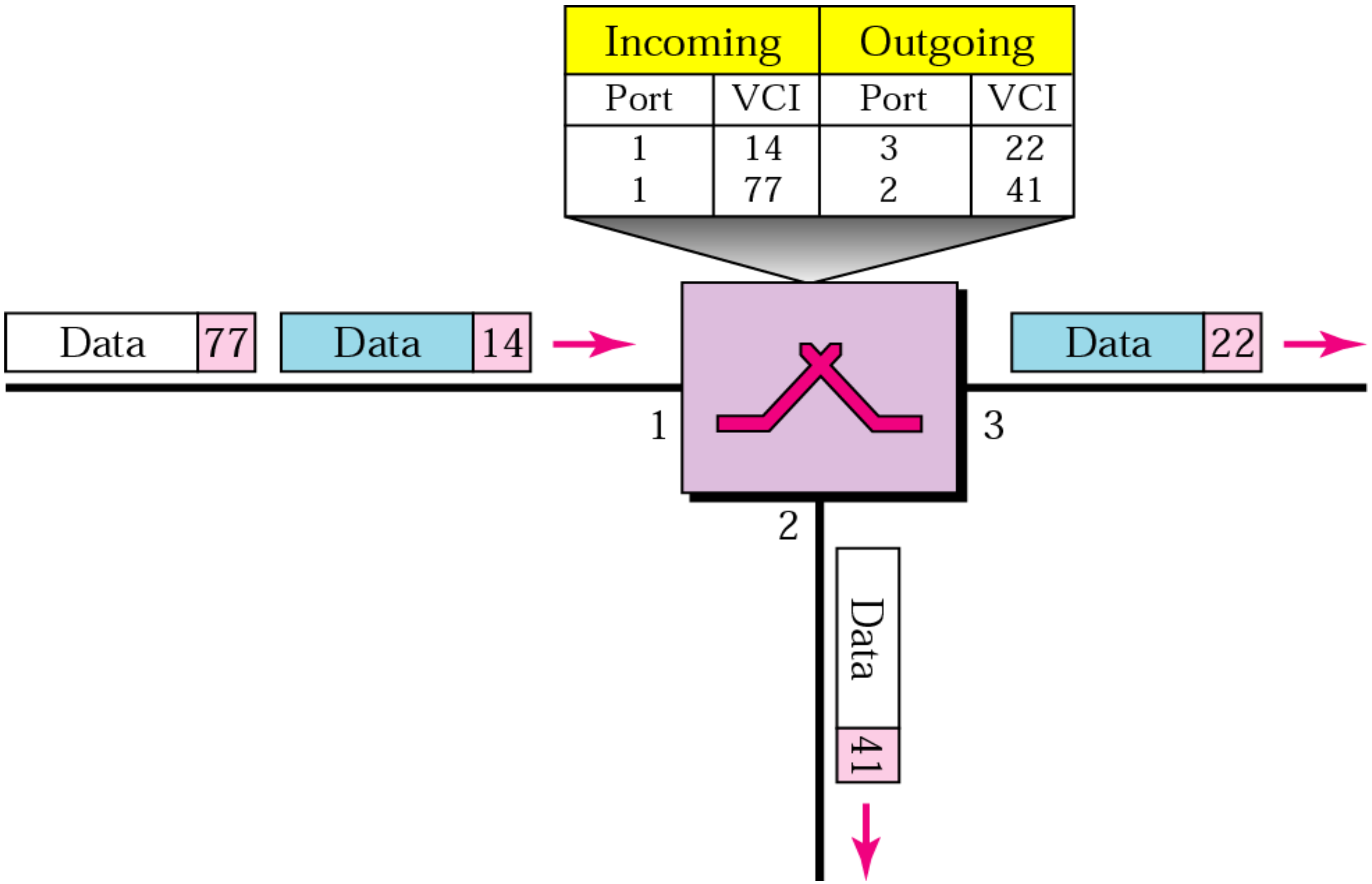
ATM

- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - complexity inside network

Virtual Circuits and Label Swapping

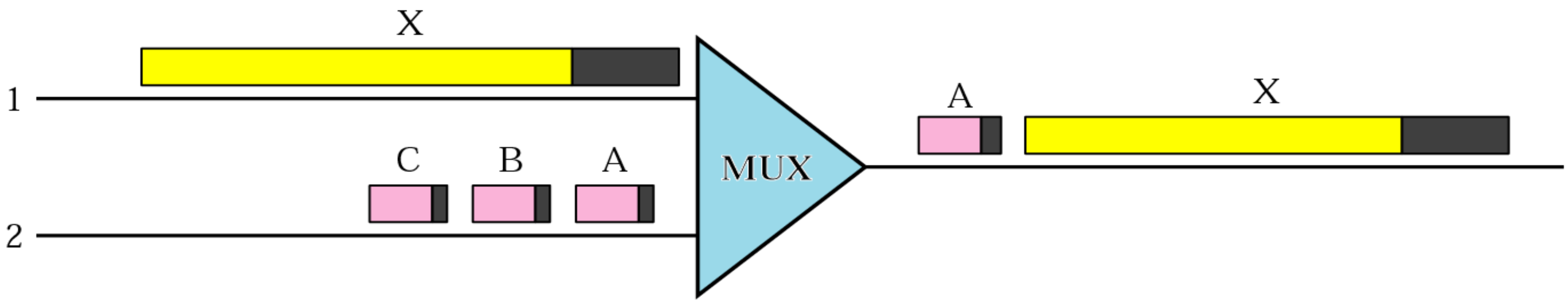
- Virtual Circuit Switching
- Multiplexing of Variable vs. Fixed Size Packets
- ATM Cell
- Virtual Path Identifiers and Virtual Channel Identifiers
- ATM Virtual Connections

Virtual Circuit Switching

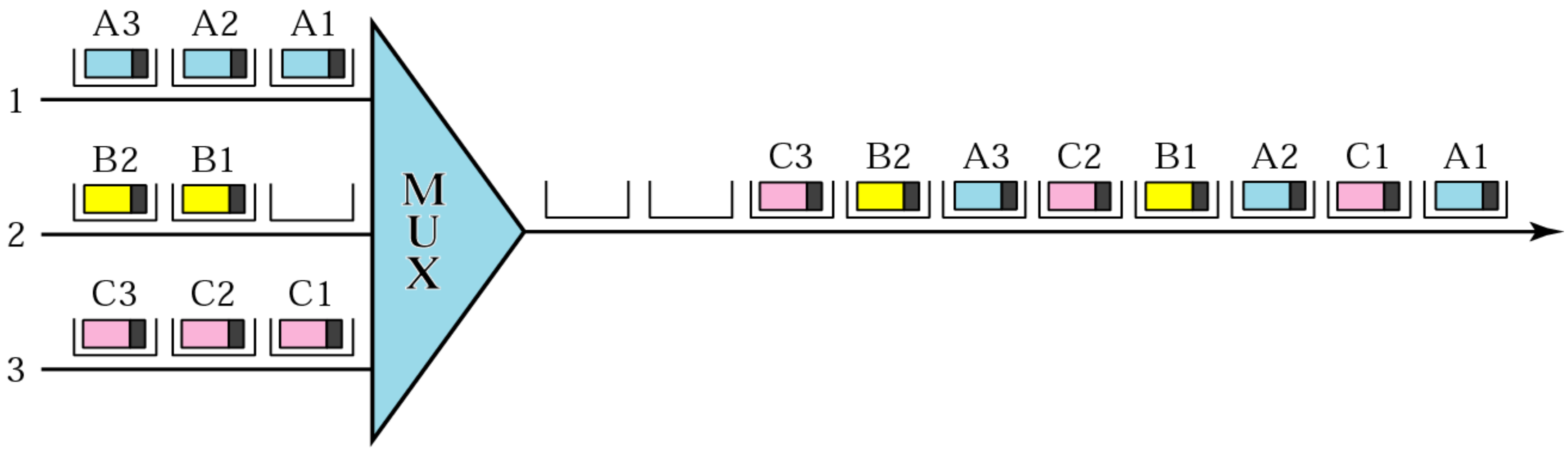


Multiplexing of Variable vs. Fixed Size Packets

- Multiplexing of variable size packets

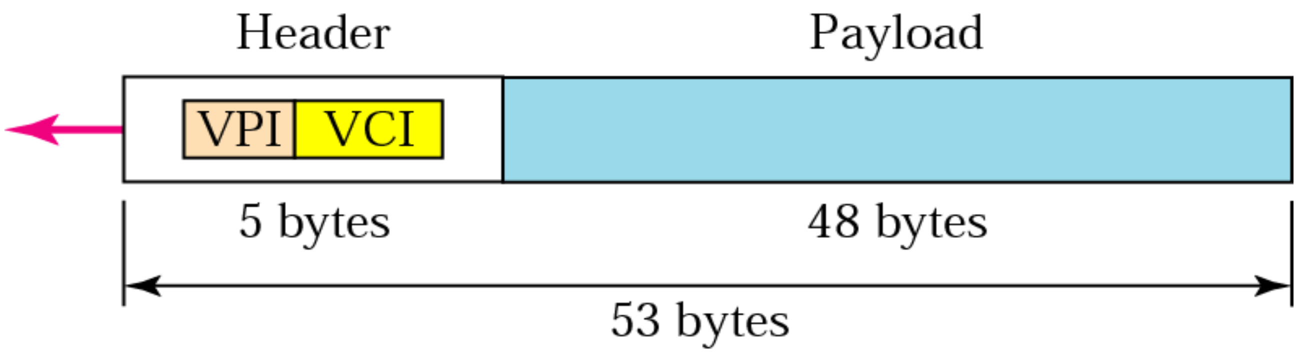


- ATM Multiplexing

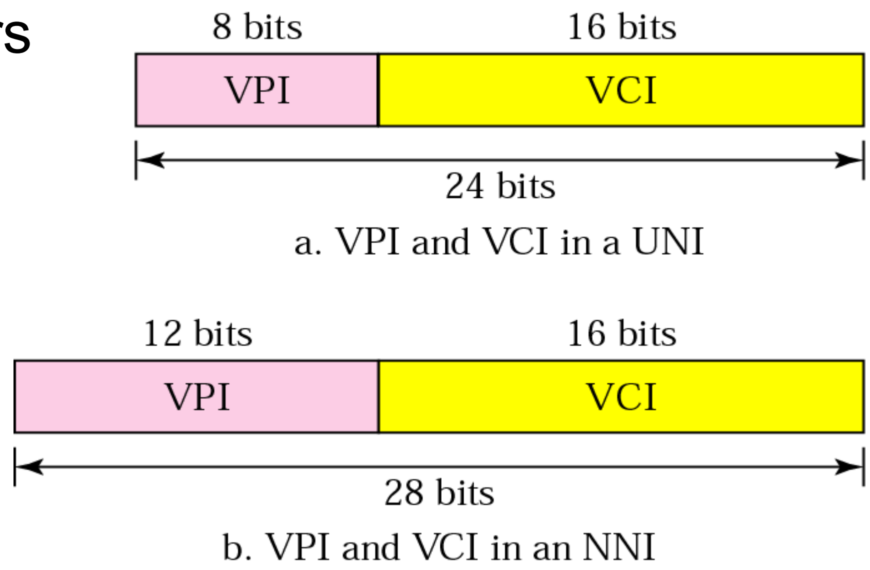


ATM Identifiers

- ATM Cell

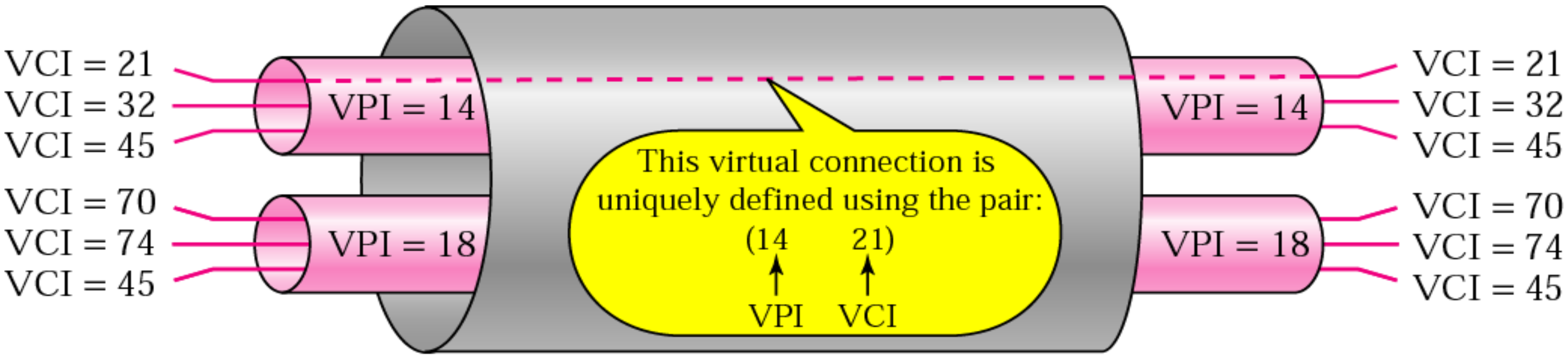


- Virtual Path Identifiers and Virtual Channel Identifiers



(UNI: User-to-Network-Interface
NNI: Network-to-Network-Interface)

ATM Virtual Connections



ATM Physical Layer (more)

Two pieces (sublayers) of physical layer:

- Transmission Convergence Sublayer (TCS): adapts ATM layer above to PMD sublayer below
- Physical Medium Dependent: depends on physical medium being used

TCS Functions:

- Header **checksum** generation: 8 bits CRC
- Cell **delineation**
- With "unstructured" PMD sublayer, transmission of **idle cells** when no data cells to send

ATM Physical Layer

Physical Medium Dependent (PMD) sublayer

- **SONET/SDH**: transmission frame structure (like a container carrying bits);
  - bit synchronization;
  - bandwidth partitions (TDM);
  - several speeds: OC3 = 155.52 Mbps; OC12 = 622.08 Mbps; OC48 = 2.45 Gbps, OC192 = 9.6 Gbps
- **TI/T3**: transmission frame structure (old telephone hierarchy): 1.5 Mbps/ 45 Mbps
- **unstructured**: just cells (busy/idle)

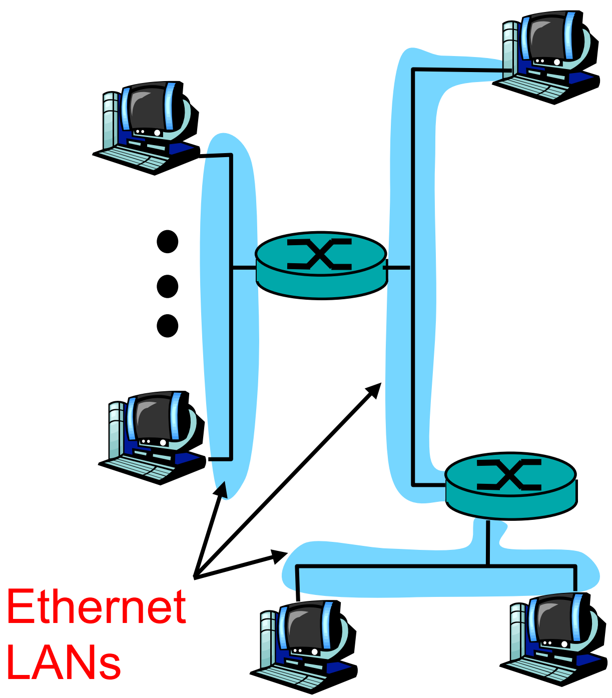
IP-Over-ATM

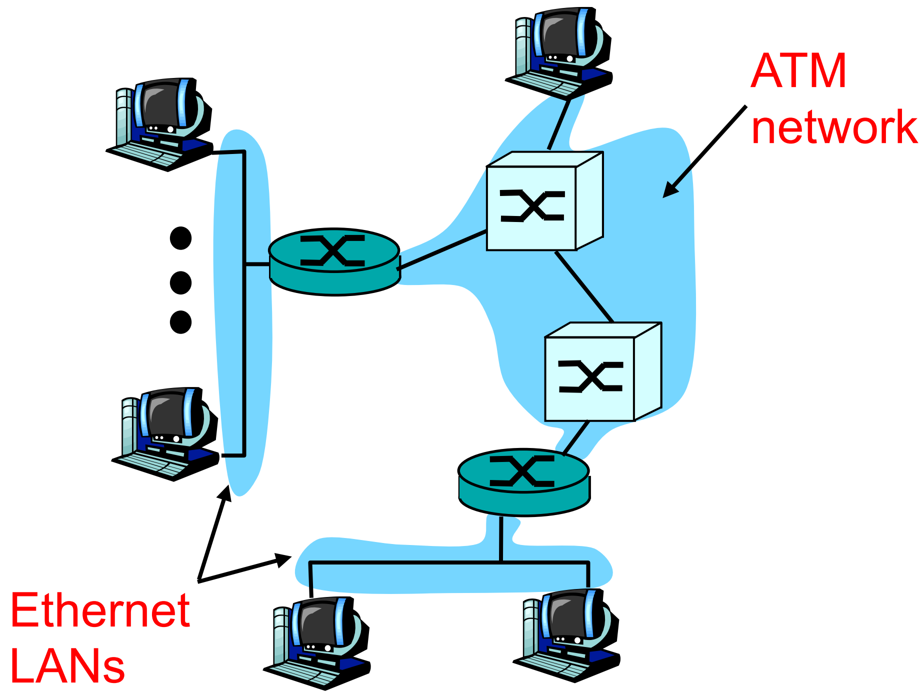
Classic IP only

- 3 "networks" (e.g., LAN segments)
- MAC (802.3) and IP addresses

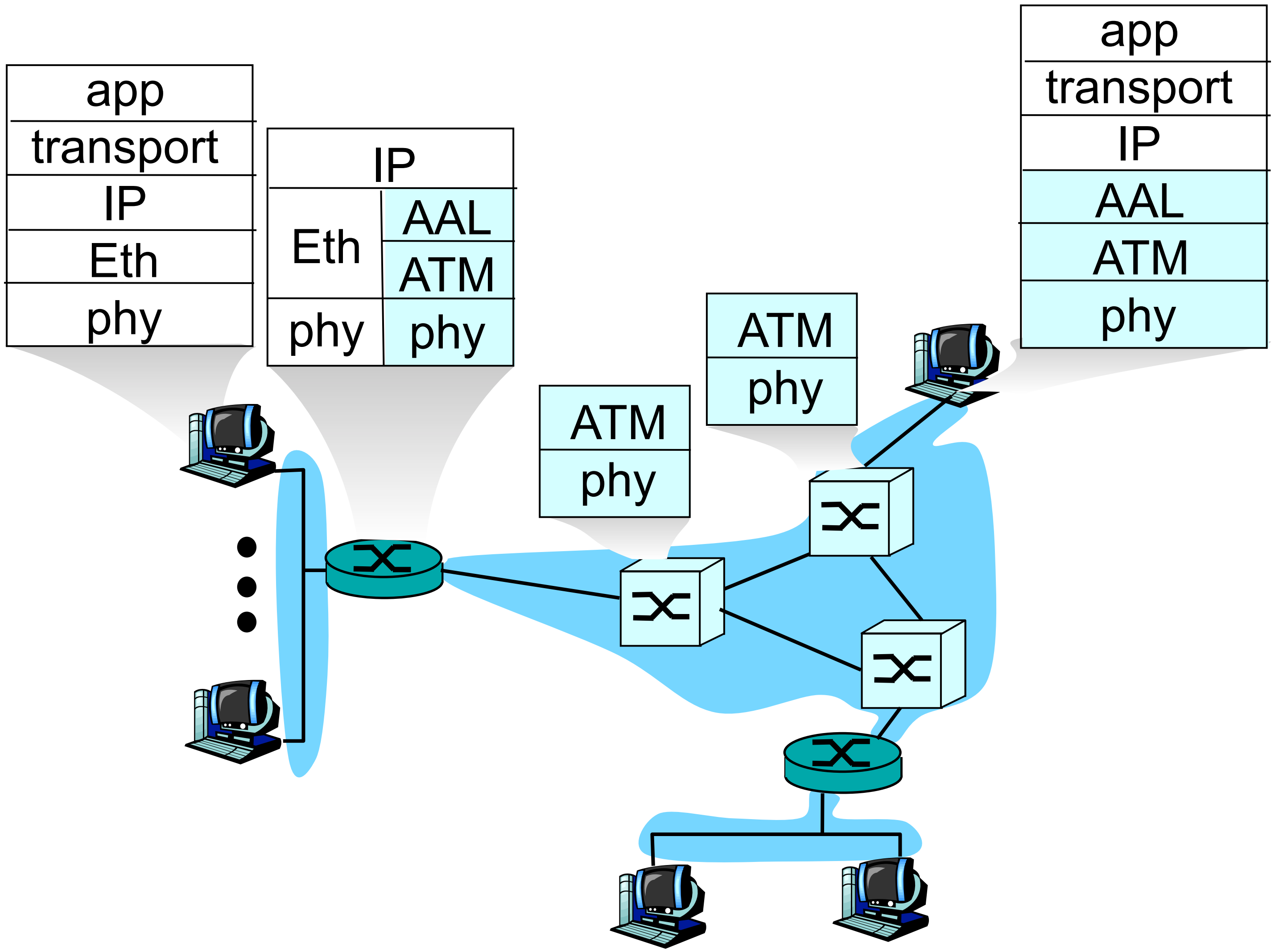
IP over ATM

- replace "network" (e.g., LAN segment) with ATM network
- ATM addresses, IP addresses





IP-Over-ATM


app
transport
IP
Eth
phy
IP
Eth
AAL
ATM
phy
phy
ATM
phy
ATM
phy
app
transport
IP
AAL
ATM
phy

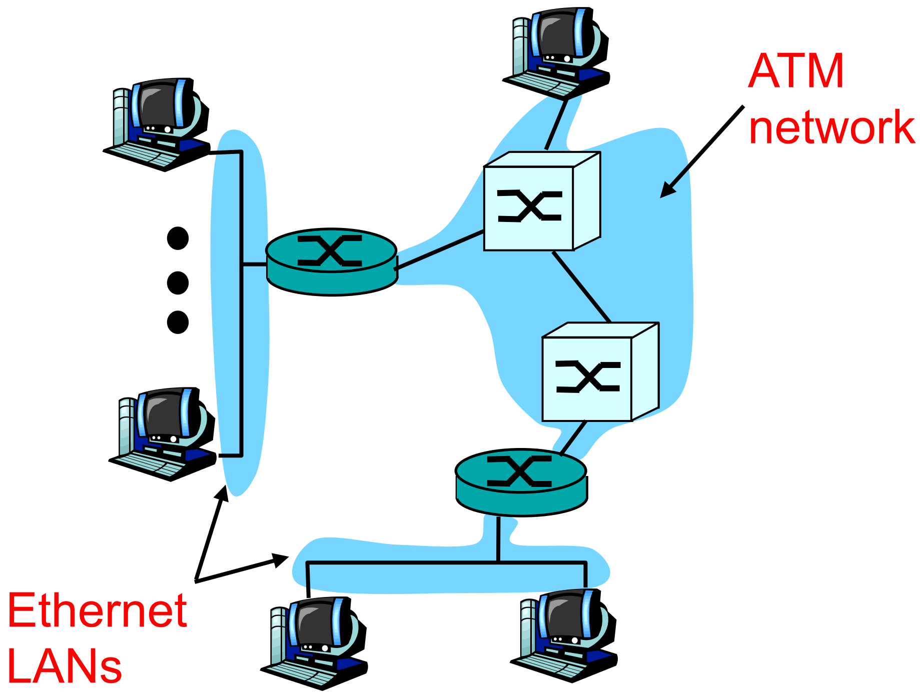
Datagram Journey in IP-over-ATM Network

- at Source Host:
  - IP layer maps between IP, ATM dest address (using ARP)
  - passes datagram to AAL5
  - AAL5 encapsulates data, segments cells, passes to ATM layer
- ATM network: moves cell along VC to destination
- at Destination Host:
  - AAL5 reassembles cells into original datagram
  - if CRC OK, datagram is passed to IP

IP-Over-ATM

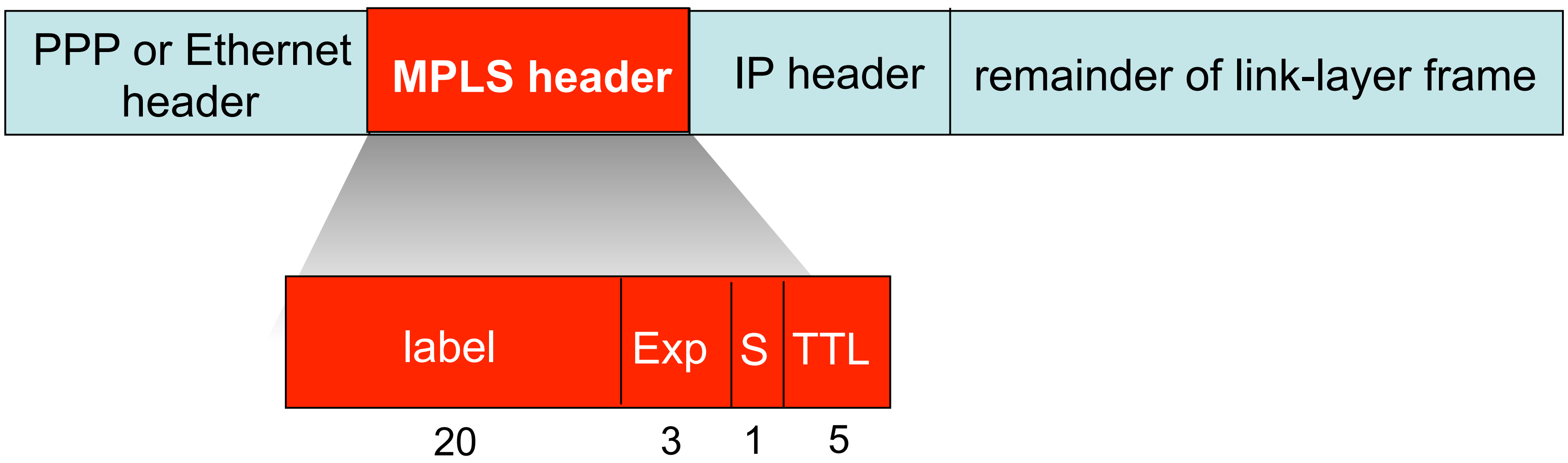
Issues:

- IP datagrams into ATM AAL5 PDUs
- from IP addresses to ATM addresses
  - just like IP addresses to 802.3 MAC addresses!



Multiprotocol label switching (MPLS)

- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



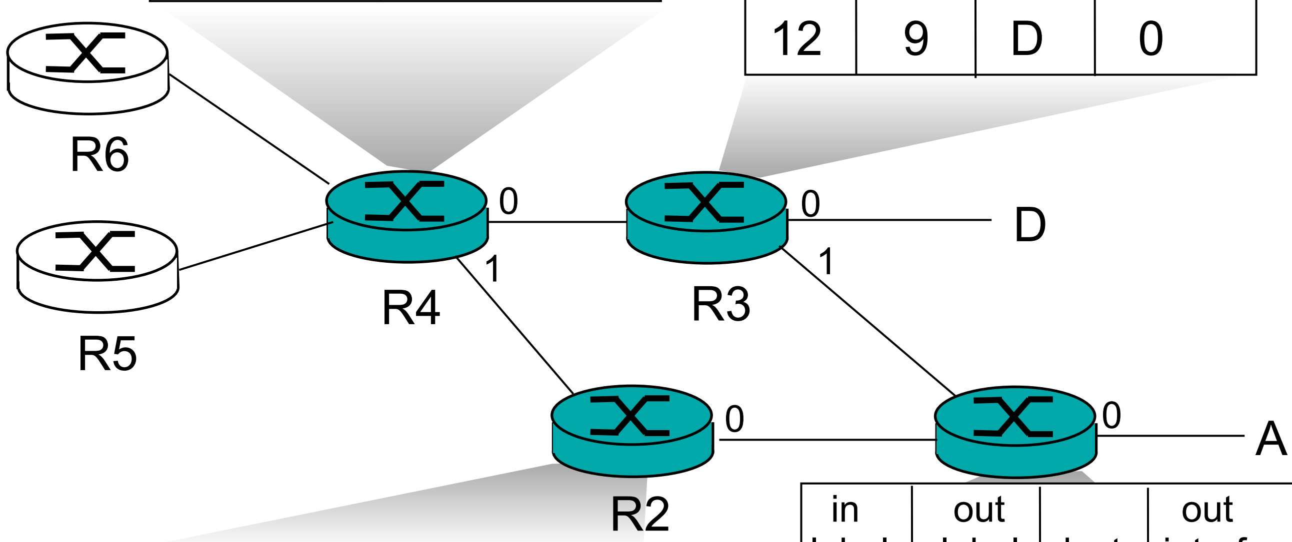
MPLS capable routers

- a.k.a. label-switched router
- forwards packets to outgoing interface based only on label value (don't inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
  - RSVP-TE
  - forwarding possible along paths that IP alone would not allow (e.g., source-specific routing) !!
  - use MPLS for traffic engineering
- must co-exist with IP-only routers

MPLS forwarding tables

in label	out label	dest	out interface
	10	A	0
	12	D	0
	8	A	1

in label	out label	dest	out interface
10	6	A	1
12	9	D	0



in label	out label	dest	out interface
8	6	A	0

in label	out label	dest	out interface
6	-	A	0

Thank you

for your attention!

Your Questions?

Questions

- Why is circuit switching expensive?
- Why is packet switching cheap?
- Is best effort packet switching able to carry voice communication?
- What happens if we introduce "better than best effort" service?
- How can we charge fairly for Internet services: by time, by volume, or flat?
- Who owns the Internet?
- You've invented a new protocol. What do you do?
- How does the Internet grow? Exponentially? What is the growth perspective?

Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Packet Switch Architectures

An overview of router architectures

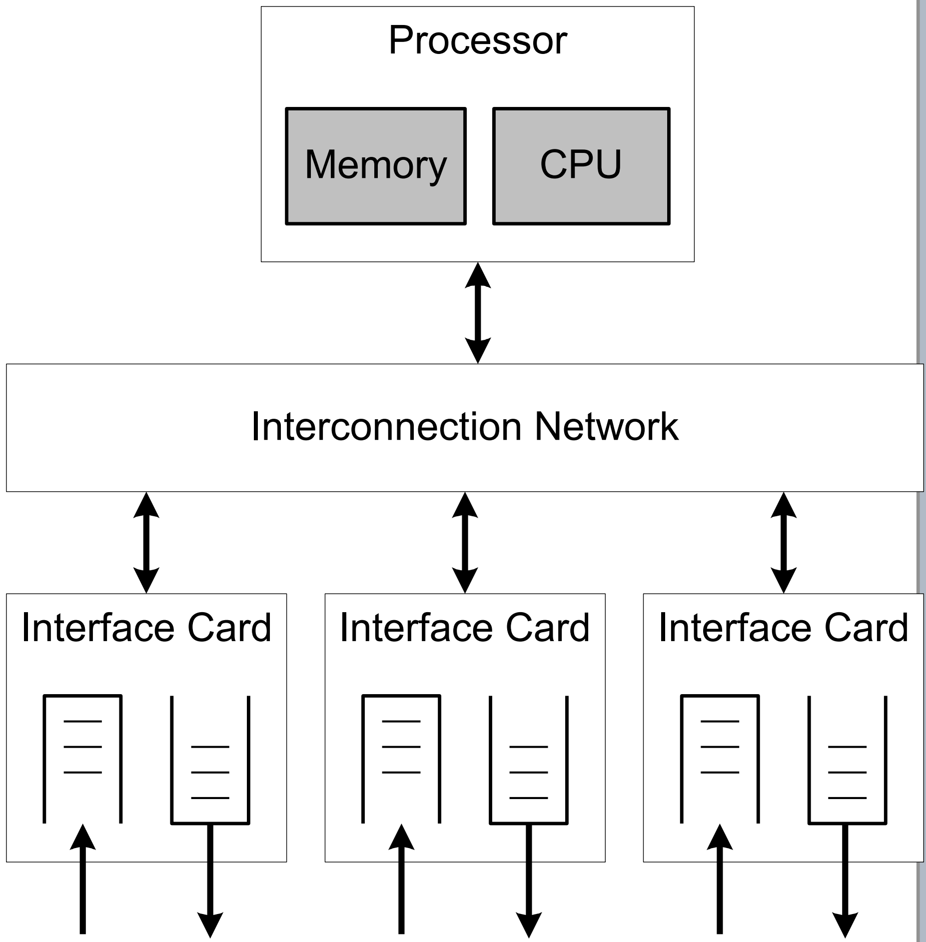
Introduction

What is a Packet Switch?

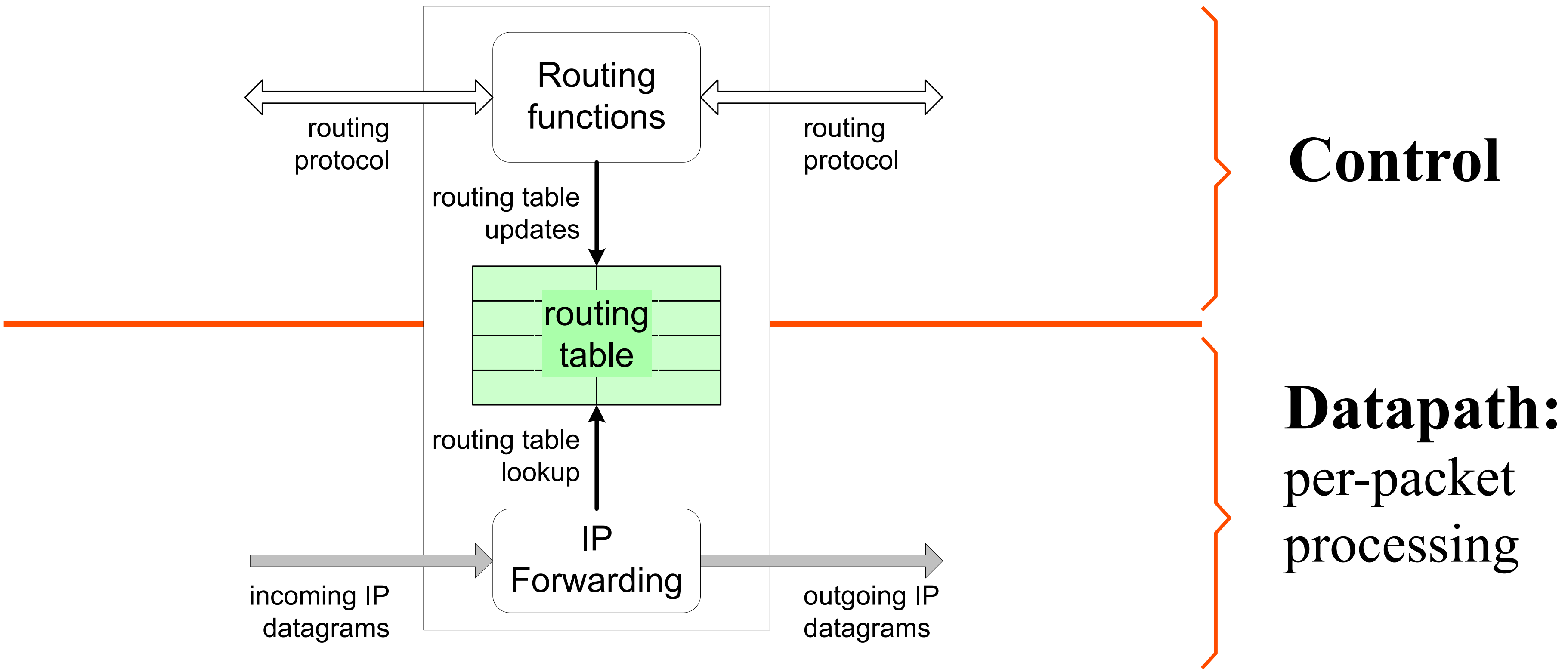
- Basic Architectural Components of an IP Router
- Example Packet Switches

Router Components

- Hardware components of a router:
  - Network interfaces
  - Interconnection network
  - Processor with a memory and CPU
- **PC router:**
  - interconnection network is the (PCI) bus and interface cards are NICs
  - All forwarding and routing is done on central processor
- **Commercial routers:**
  - Interconnection network and interface cards are sophisticated
  - Processor is only responsible for control functions **(route processor)**
  - Almost all forwarding is done on interface cards



Functional Components



Routing and Forwarding

Routing functions include:

- route calculation
- maintenance of the routing table
- execution of routing protocols

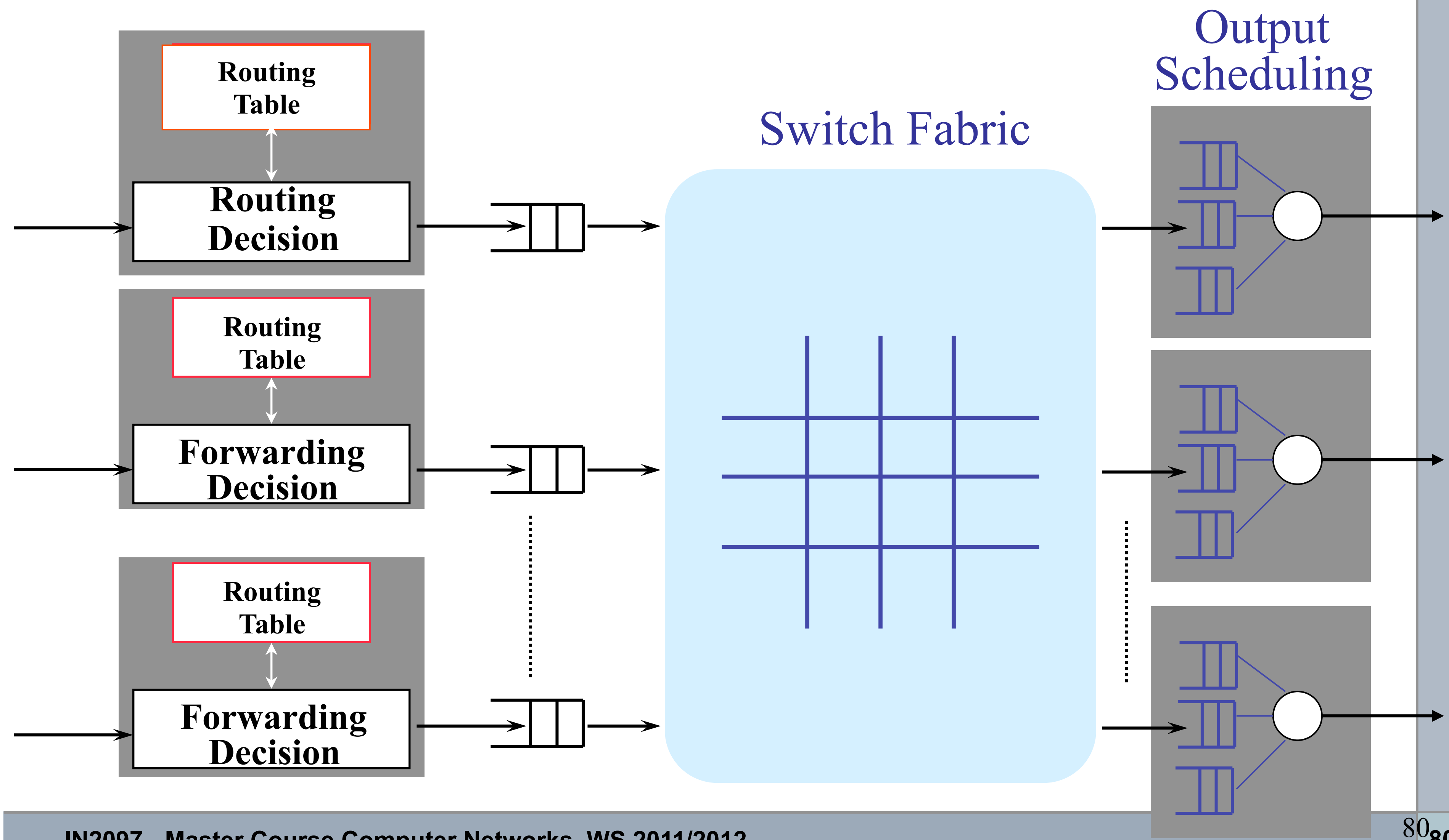
- On commercial routers handled by a single general purpose processor, called *route processor*

IP forwarding is per-packet processing

- On high-end commercial routers, IP forwarding is distributed
- Most work is done on the interface cards

Basic Architectural Components

Per-packet processing



IP Router

- Lookup packet destination address in forwarding table.
  - If known, forward to correct port.
  - If unknown, drop packet.
- Decrement TTL, update header checksum.
- Forward packet to outgoing interface.
- Transmit packet onto link.

ATM Switch

- Look up VCI/VPI of cell in VC table.
- Replace old VCI/VPI with new.
- Forward cell to outgoing interface.
- Transmit cell onto link.

Ethernet Switch

- Lookup frame destination address in forwarding table.
  - If known, forward to correct port.
  - If unknown, broadcast to all ports.
- Learn source address of incoming frame.
- Forward frame to outgoing interface.
- Transmit frame onto link.

Thank you

for your attention!

Your Questions?