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

- ❑ Network virtualisation:
  Link virtualization: MPLS

Project

- Grades
  - are given in svn

    grades.txt

- Forum
  - you find the forum in https://www.moodle.tum.de/
  - 2 Forums online
    - *Announcements*

      administered by teachers and used for announcements.
    - *Projects*

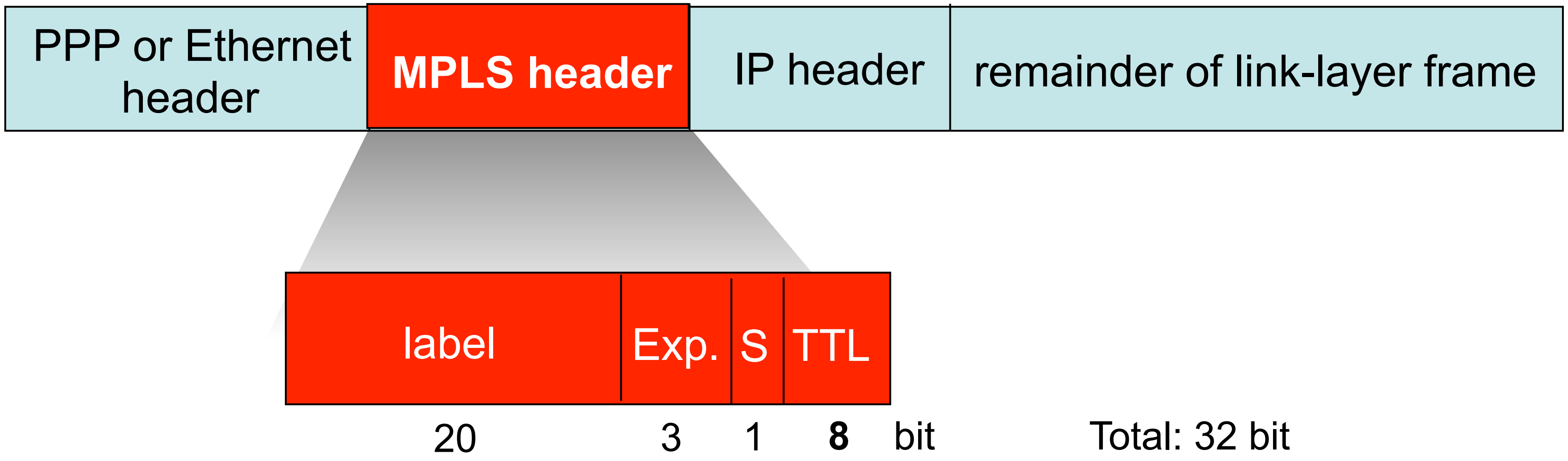
      for use to exchange project related information by students

Network Architectures

Link virtualization:
MPLS - Multi-Protocol Label Switching

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
  - IP datagram still keeps IP address
  - RFC 3032 defines MPLS header
    - Label: has role of Virtual Circuit Identifier
    - Exp: experimental usage, may specify Class of Service (CoS)
    - S: Bottom of Stack - end of series of stacked headers
    - TTL: time to live



PPP or Ethernet header | MPLS header | IP header | remainder of link-layer frame

label | Exp. | S | TTL

20 | 3 | 1 | **8** bit

Total: 32 bit

Multiprotocol label switching (MPLS)

- RFC 3270: Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P. and J. Heinanen, “Multi-Protocol Label Switching (MPLS) Support of Differentiated Services”, May 2002.
  - EXP: 3 bits - this field contains the value of the EXP field for the EXP<->PHB (Per-Hop-Behaviour) mapping
  - Mapping transported via signaling protocol
- RFC 3140: Black, D., Brim, S., Carpenter, B. and F. Le Faucheur, "Per Hop Behavior Identification Codes", June 2001.
    - Case 1: PHBs defined by standards action, as per [RFC 2474]. PHB is recommended 6-bit DSCP value for that PHB,
      left-justified in a 16 bit field, with bits 6 through 15 set to zero.
    - Case 2: PHBs not defined by standards action, i.e., experimental or local use PHBs In this case an arbitrary 12 bit PHB-ID is placed left-justified in the a bit field.
      Bit 15 is set to 1, Bits 12 and 13 are zero.

MPLS TTL Processing

c.f. RFC 3032 - MPLS Label Stack Encoding

- Protocol-independent rules
  - "outgoing TTL" of a labeled packet is either
    a) one less than the incoming TTL, or b) zero.
  - Packets with TTL=0 are discarded
- IP-dependent rules
  - When an IP packet is first labeled, the TTL field of the label stack is set to the value of the IP TTL field.
  - If the IP TTL field needs to be decremented, as part of the IP processing, it is assumed that this has already been done.
  - When a label is popped, and the resulting label stack is empty, then the value of the IP TTL field SHOULD BE replaced with the outgoing MPLS TTL value.
  - A network administration may prefer to decrement the IPv4 TTL by one as it traverses an MPLS domain.

ICMP

- When a router receives an IP datagram that it can't forward, it sends an ICMP message to the datagram's originator
- The ICMP message indicates why the datagram couldn't be delivered
  - E.g., Time Expired, Destination Unreachable
- The ICMP message also contains the IP header and at least leading 8 octets of the original datagram
  - RFC 1812 - Requirements for IP Version 4 Routers extends this to "as many bytes as possible"
  - Historically, every ICMP error message has included the Internet header and at least
  - Including only the first 8 data bytes of the datagram that triggered the error is no longer adequate, due to use e.g. of IP-in-IP tunneling

ICMP in presence of MPLS

- When an LSR receives an MPLS encapsulated datagram that it can't deliver
  - It removes entire MPLS labels stack
  - It sends an ICMP message to datagram's originator
- The ICMP message indicates why the datagram couldn't be delivered (e.g., time expired, destination unreachable)
- The ICMP message also contains the IP header and leading 8 octets of the original datagram
  - RFC 1812 extends this to "as many bytes as possible"

ICMP in Presence of MPLS

Issue

- The ICMP message contains no information regarding the MPLS stack that encapsulated the datagram when it arrived at the LSR
- This is a significant omission because:
  - The LSR tried to forward the datagram based upon that label stack
  - Resulting ICMP message may be confusing

Why?

ICMP in Presence of MPLS

Issue

- ❑ ICMP Destination Unreachable
  - ▪ Message contains IP header of original datagram
  - ▪ Router sending ICMP message has an IP route to the original datagram’s destination
  - ▪ Original datagram couldn’t be delivered because MPLS forwarding path was broken
- ❑ ICMP Time Expired
  - ▪ Message contains IP header of original datagram
  - ▪ TTL value in IP header is greater than 1
  - ▪ TTL expired on MPLS header. ICMP Message contains IP header of original datagram

ICMP with MPLS

c.f. RFC 4950 - ICMP Extensions for Multiprotocol Label Switching

- defines an ICMP extension object that permits an LSR to append MPLS information to ICMP messages.
- ICMP messages include the MPLS label stack, as it arrived at the router that is sending the ICMP message.
- equally applicable to ICMPv4 [RFC792] and ICMPv6 [RFC4443]
- sample output from an enhanced TRACEROUTE:

```
> traceroute 192.0.2.1
traceroute to 192.0.2.1 (192.0.2.1), 30 hops max, 40 byte packets
1 192.0.2.13 (192.0.2.13) 0.661 ms 0.618 ms 0.579 ms
2 192.0.2.9 (192.0.2.9) 0.861 ms 0.718 ms 0.679 ms
   MPLS Label=100048 Exp=0 TTL=1 S=1
3 192.0.2.5 (192.0.2.5) 0.822 ms 0.731 ms 0.708 ms
   MPLS Label=100016 Exp=0 TTL=1 S=1
4 192.0.2.1 (192.0.2.1) 0.961 ms 8.676 ms 0.875 ms
```

- MPLS Label Stack Object: can be appended to ICMP Time Exceeded and Destination Unreachable messages.

```
       0             1              2            3
+-------------+-------------+-------------+-------------+
|                    Label         |EXP |S|     TTL     |
+-------------+-------------+-------------+-------------+
|                                                       |
|         // Remaining MPLS Label Stack Entries //      |
|                                                       |
+-------------+-------------+-------------+-------------+
```

- Must be preceded by an ICMP Extension Structure Header and an ICMP Object Header, defined in [RFC4884].

Multi-Part ICMP Messages - RFC 4884

- ICMP Extension Structure may be appended to ICMP v4 / v6 Destination Unreachable and Time Exceeded messages
- ICMP Extension Structure Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|      (Reserved)       |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

  ICMP extension version number: 2

- ICMP Object Header and Object Payload

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length            |   Class-Num   |   C-Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   // (Object Payload) //                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

# MPLS for Linux

# The work of James Leu:

https://sourceforge.net/projects/mpls-linux/

Discussions:

http://sourceforge.net/mailarchive/forum.php?forum_name=mpls-linux-devel

# Bug fixes of Jorge Boncompte:

http://mpls-linux.git.sourceforge.net/git/gitweb.cgi?p=mpls-linux/net-next;a=shortlog;h=refs/heads/net-next-mpls

# Additional bug fixes by Igor Maravić:

https://github.com/i-maravic/MPLS-Linux

https://github.com/i-maravic/iproute2

# MPLS for Linux Labs

by Irina Dumitrascu and Adrian Popa: graduation project with purpose of teaching MPLS to university students, at Limburg Catholic University College

http://ontwerpen1.khlim.be/~lrutten/cursussen/comm2/mpls-linux-docs/

inlcudes e.g. Layer 2 VPN with MPLS, Layer 3 VPN with MPLS

Virtual Private Networks

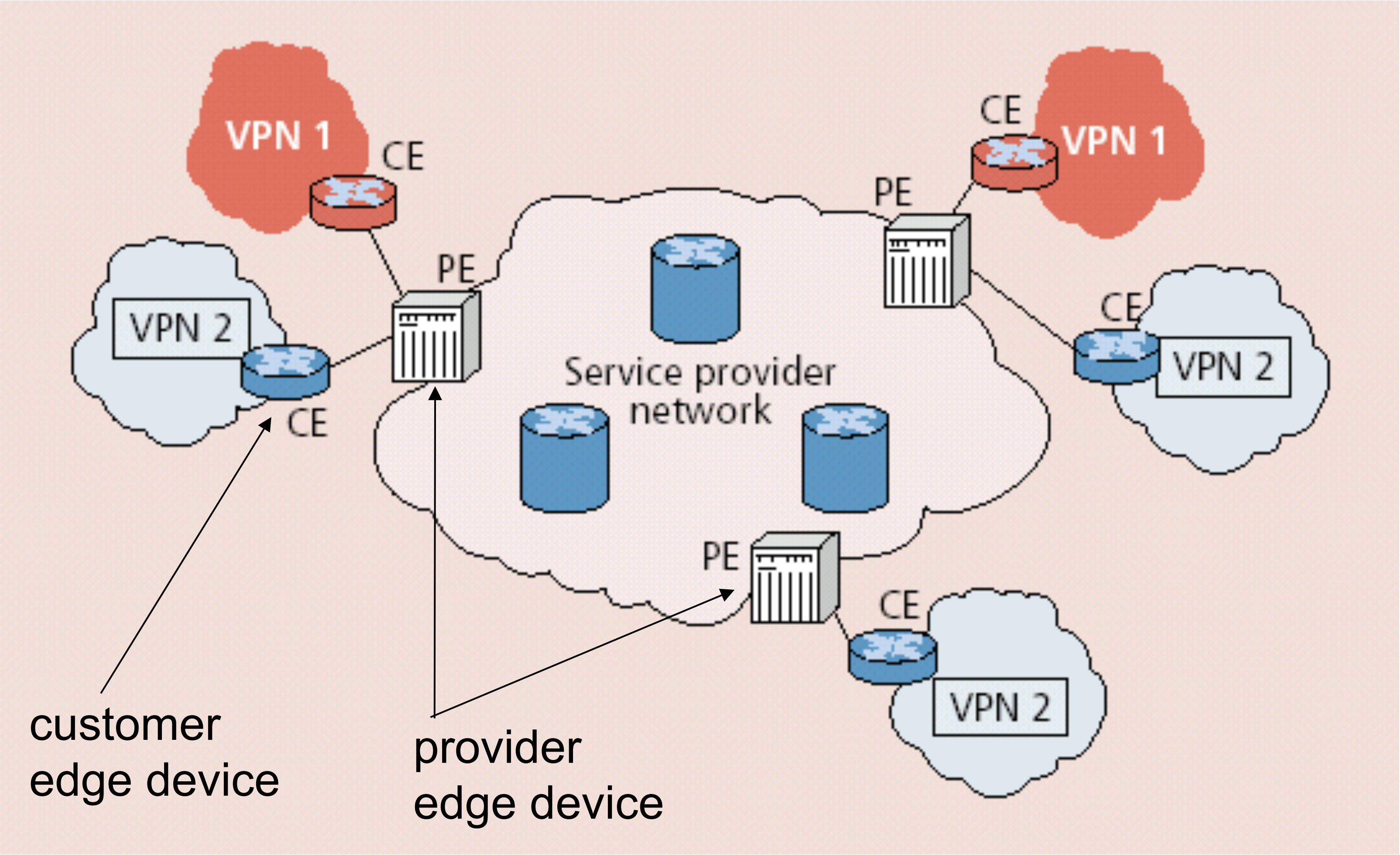
Virtual Private Networks (VPN)

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



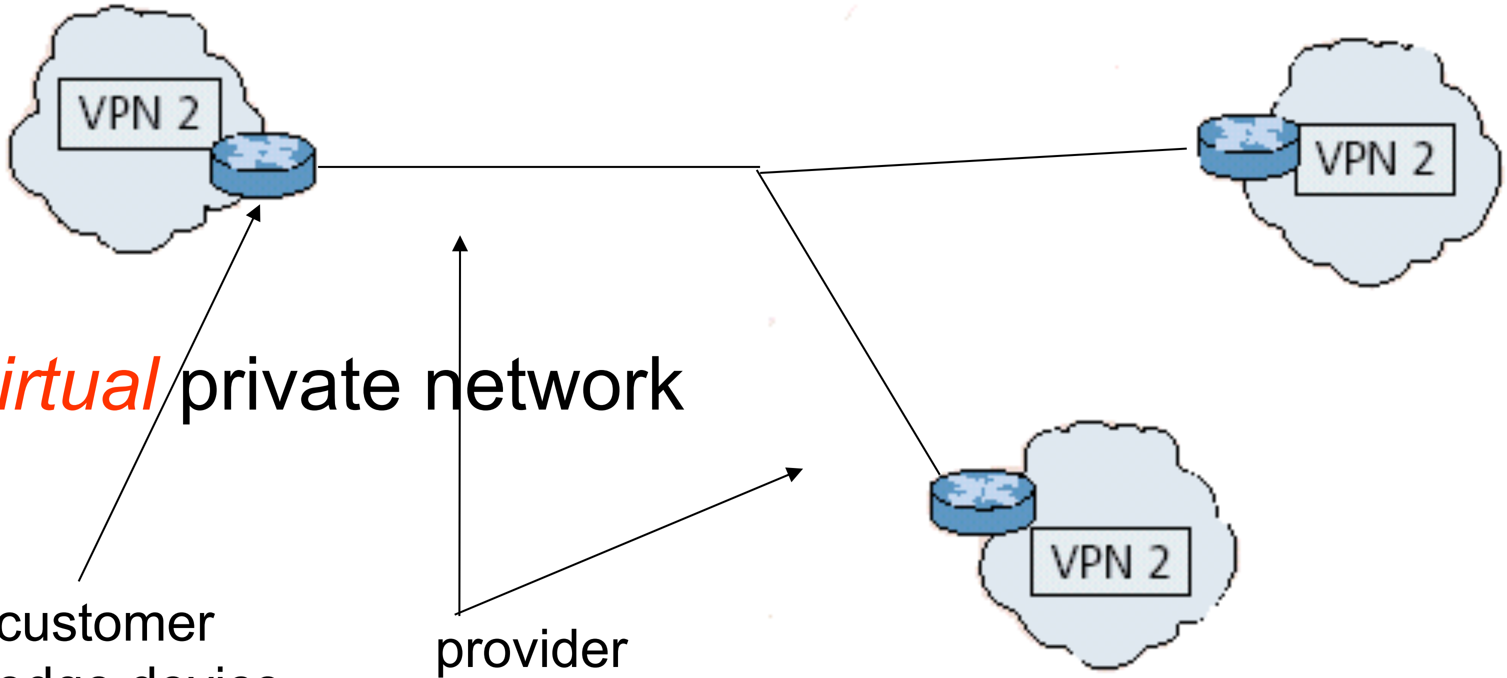
customer
edge device

provider
edge device

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, redundancy, 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

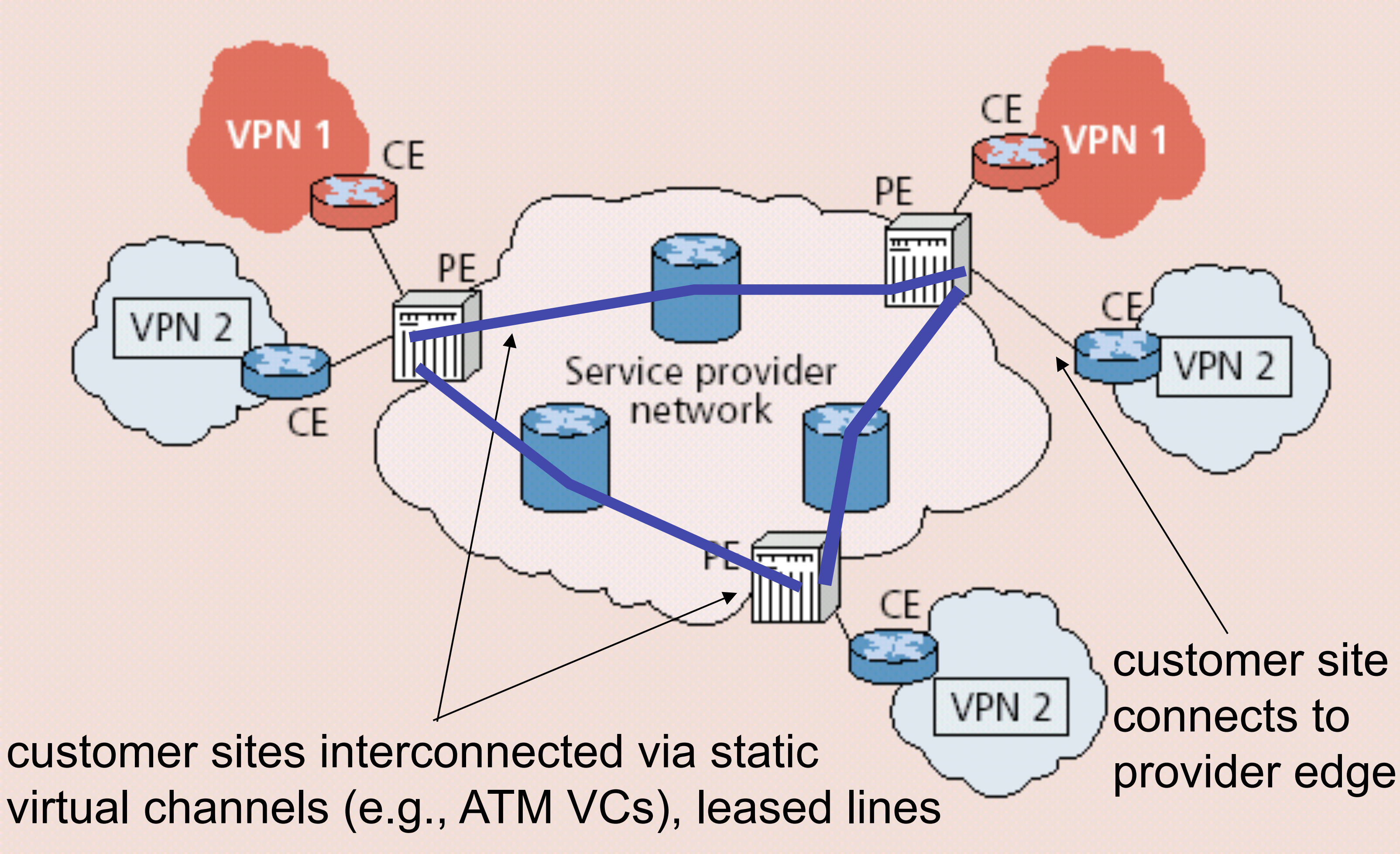


virtual private network

customer
edge device

provider
edge device

Leased-Line VPN

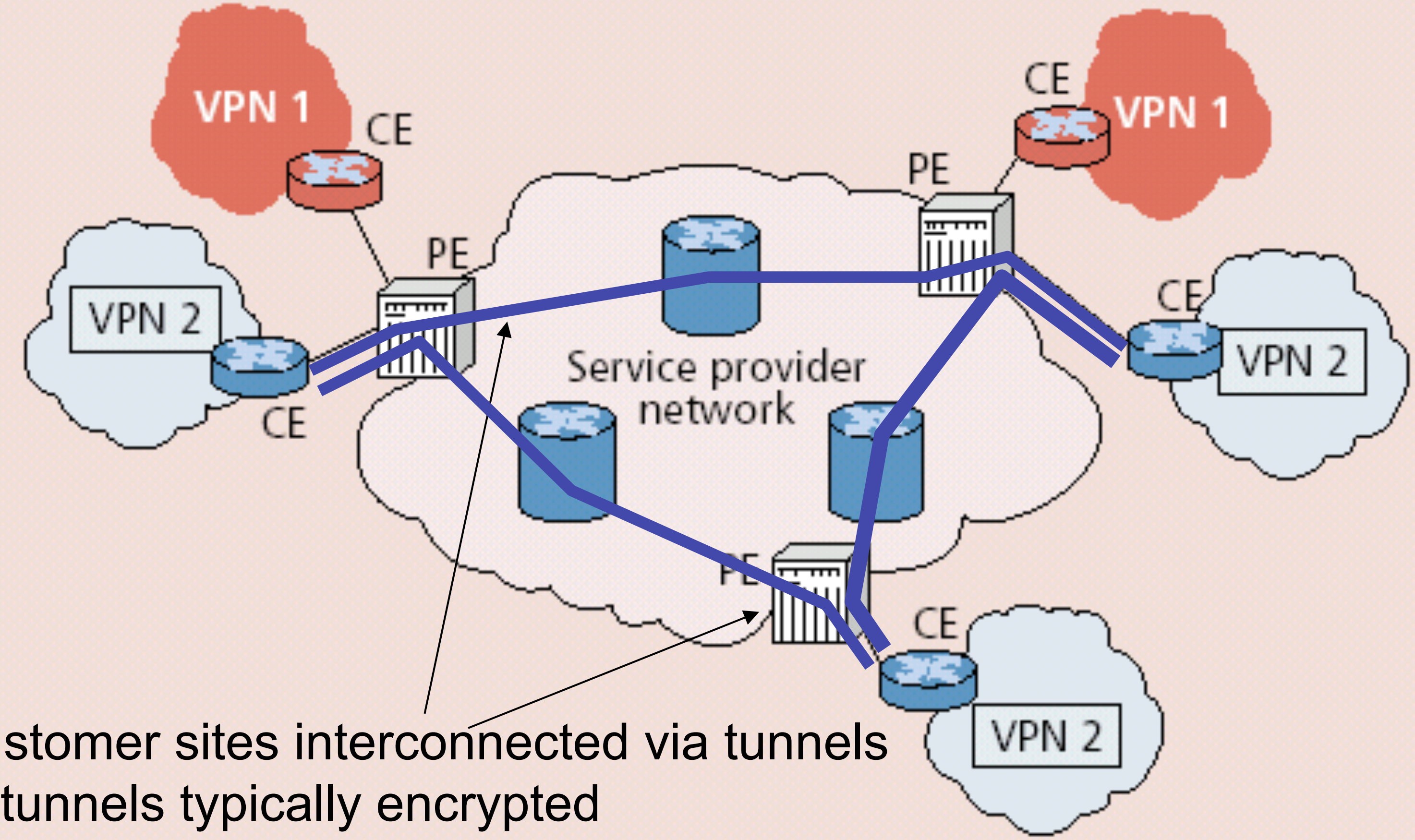


customer sites interconnected via static virtual channels (e.g., ATM VCs), leased lines

customer site connects to provider edge

Customer Premise VPN

- all VPN functions implemented by customer



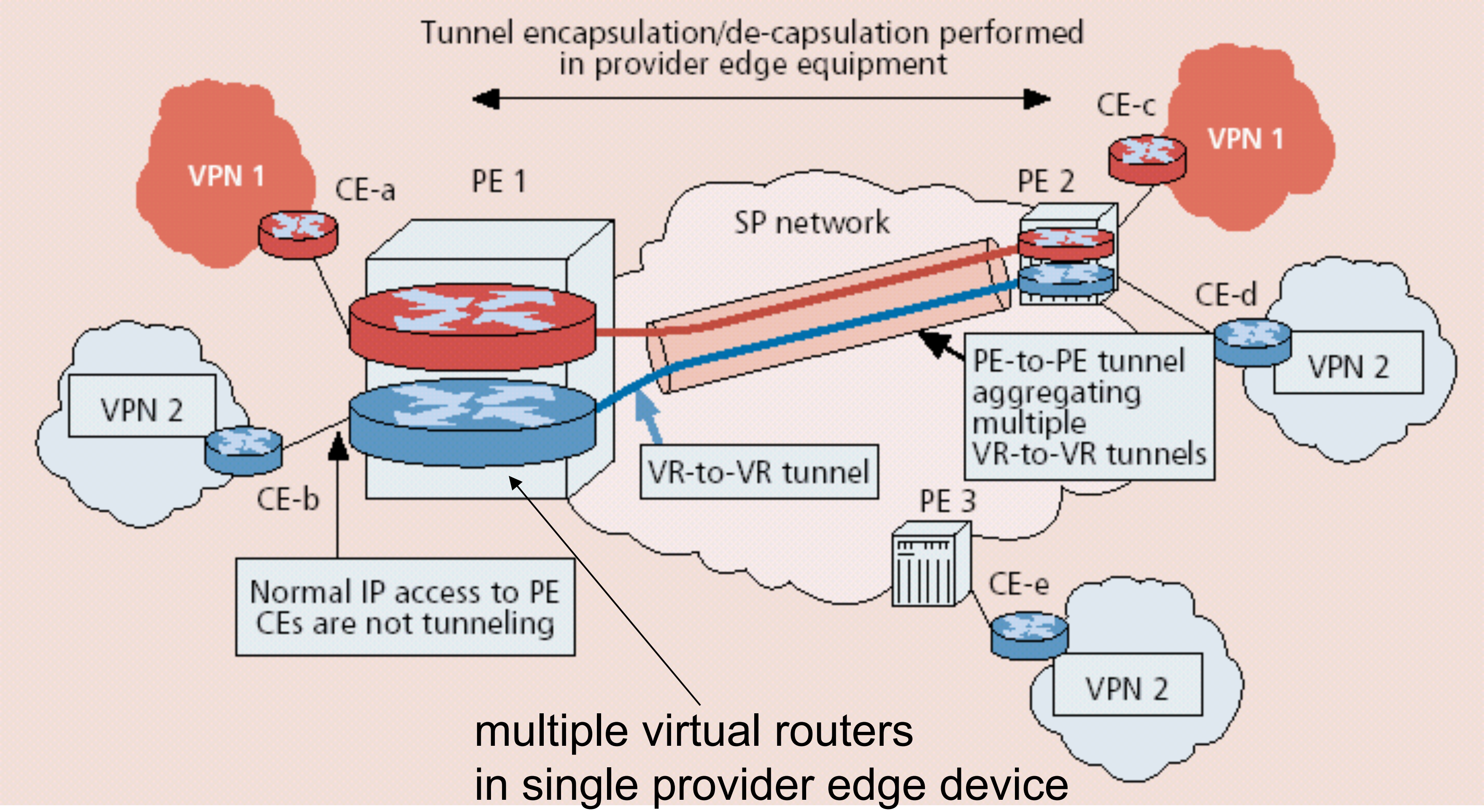
customer sites interconnected via tunnels

- tunnels typically encrypted
- Service provider treats VPN packets like all other packets

Variants of VPNs

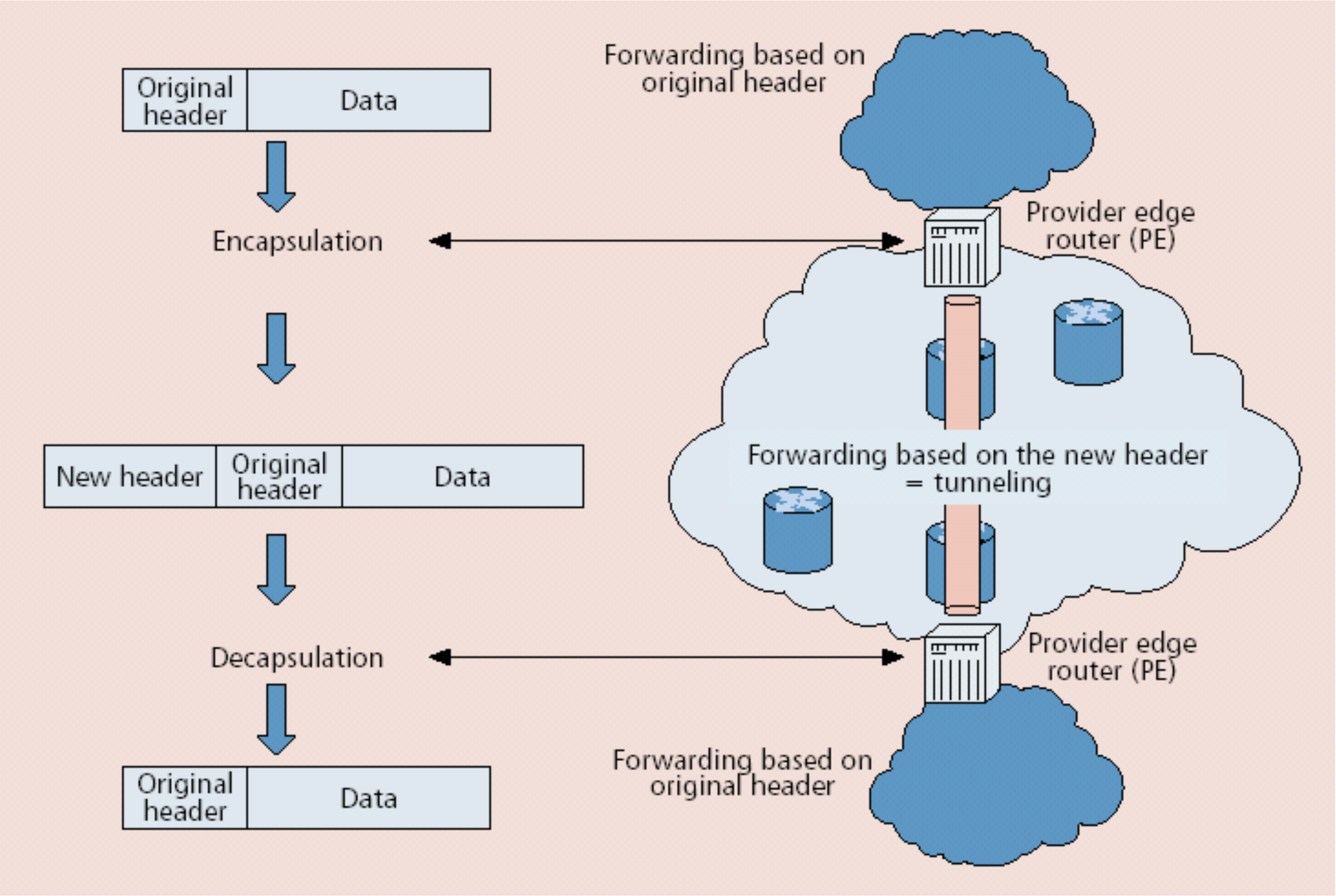
- Leased-line VPN
  - configuration costs and maintenance by service provider: long time to set up, manpower
- CPE-based VPN
  - expertise by customer to acquire, configure, manage VPN
- Network-based VPN
  - Customer routers connect to service provider routers
  - Service provider routers maintain separate (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

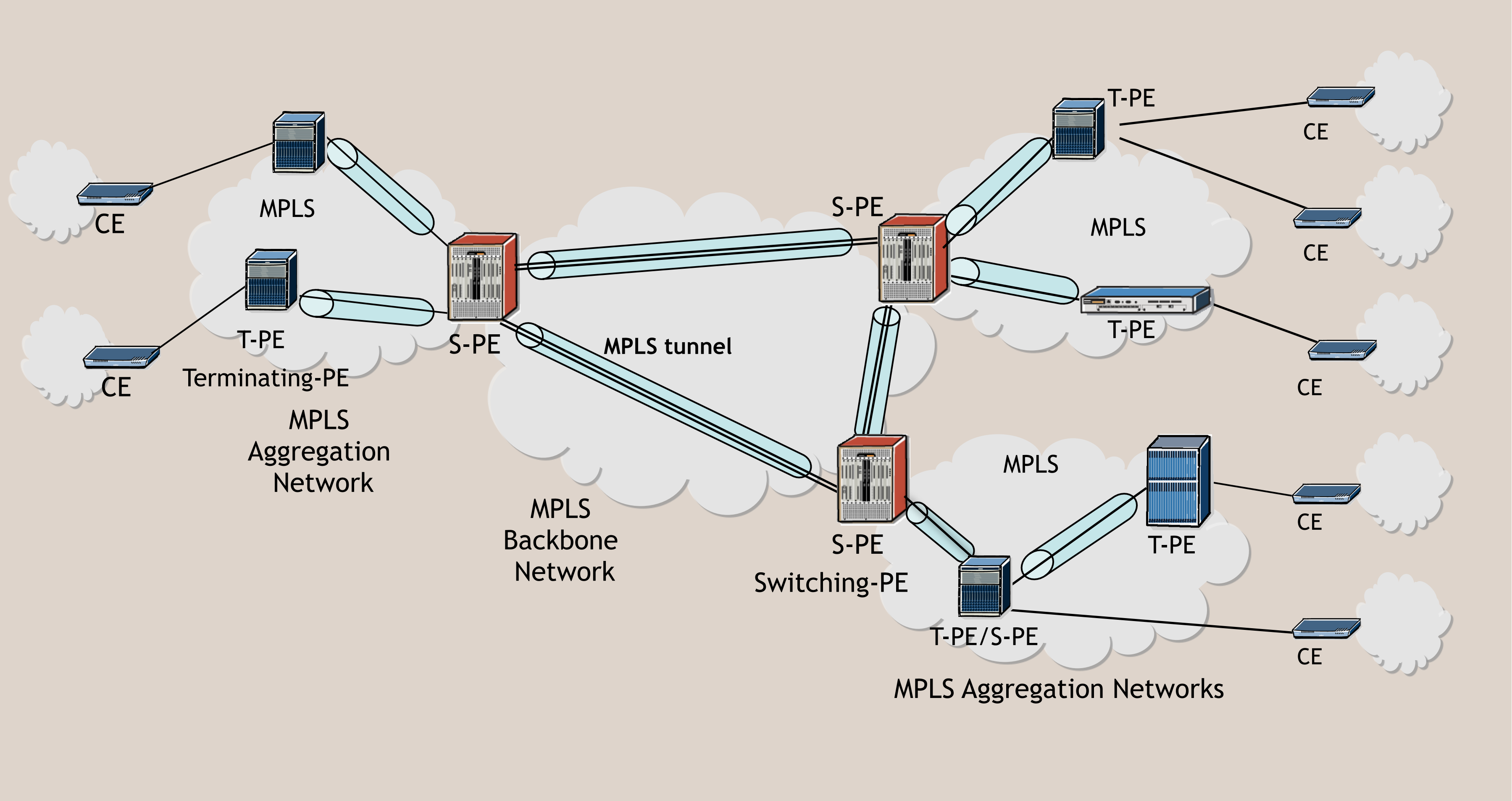

Tunnel encapsulation/de-capsulation performed in provider edge equipment
VPN 1
CE-a
PE 1
SP network
PE 2
CE-c
VPN 1
CE-d
VPN 2
VPN 2
CE-b
PE-to-PE tunnel aggregating multiple VR-to-VR tunnels
VR-to-VR tunnel
PE 3
CE-e
VPN 2
Normal IP access to PE CEs are not tunneling

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

MPLS-based VPN



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?

Tunneling and Overlay

Benefits

- Allows bootstrapping and incremental deployment of innovative protocols and mechanisms
- Many new networks have begun as overlay networks
- Innovations do not have to be deployed at every node

Costs

- Overhead
  - Additional layer: additional header + processing
- Complexity
  - possible unintended interactions between layers

Packet Switch Architectures

An overview of router architectures

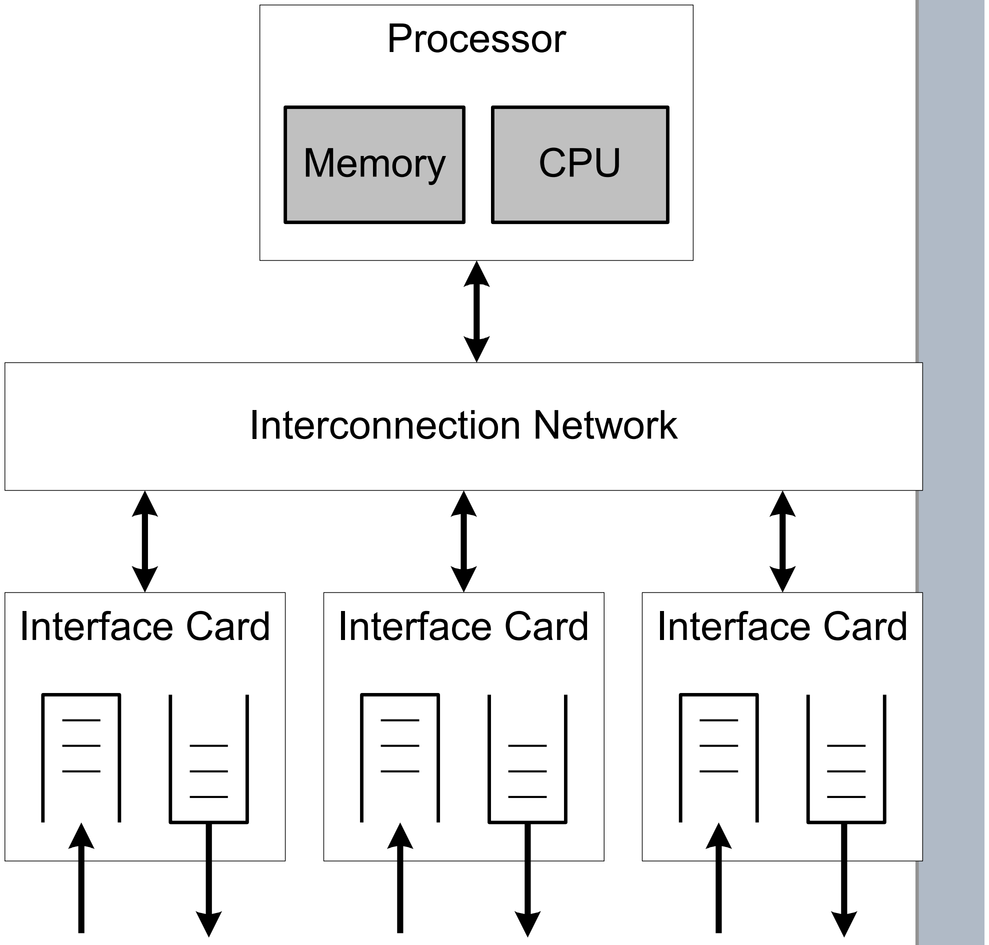
Introduction

What is a Packet Switch?

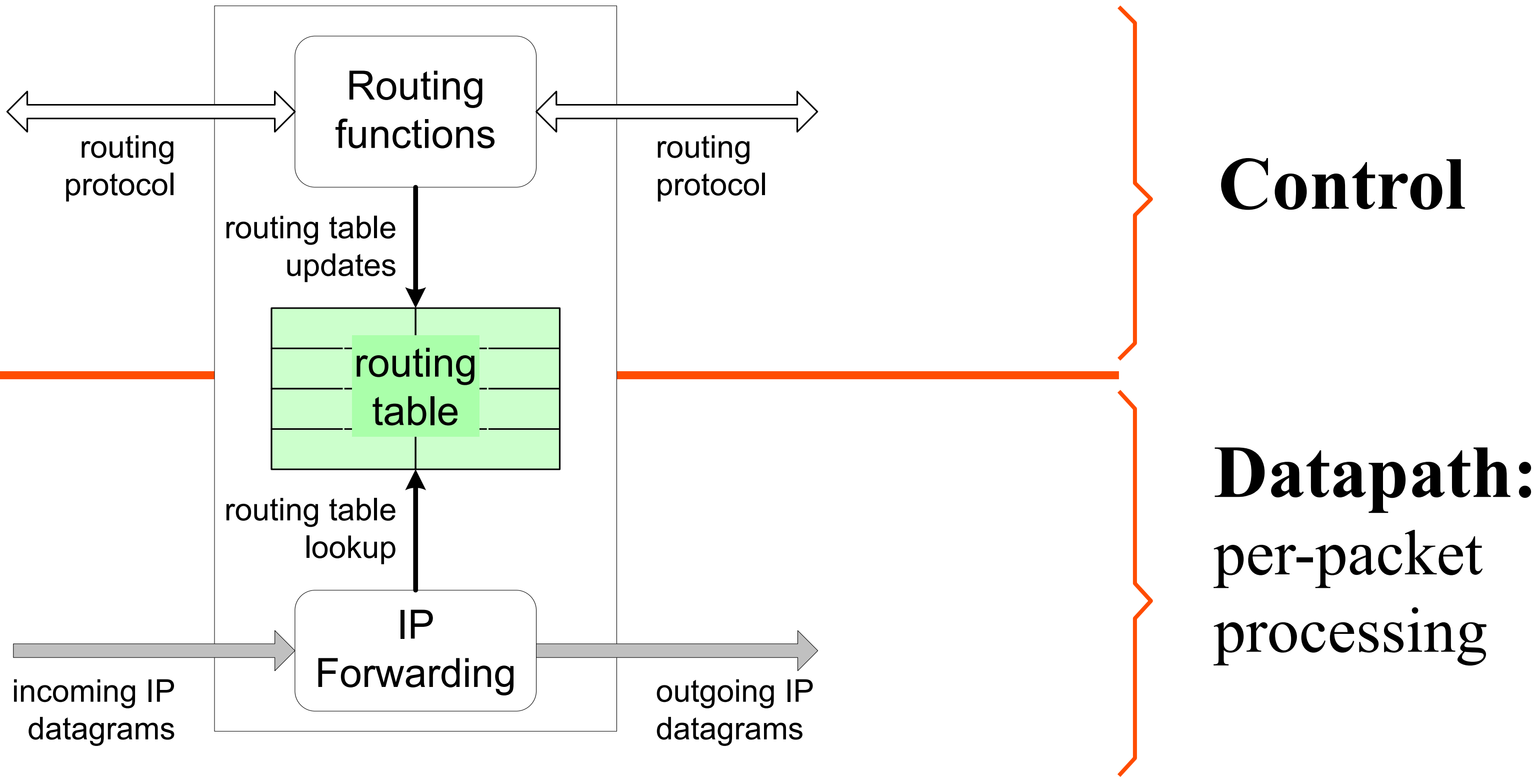
- ❑ Basic Architectural Components of an IP Router
- ❑ Example Packet Switches

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*

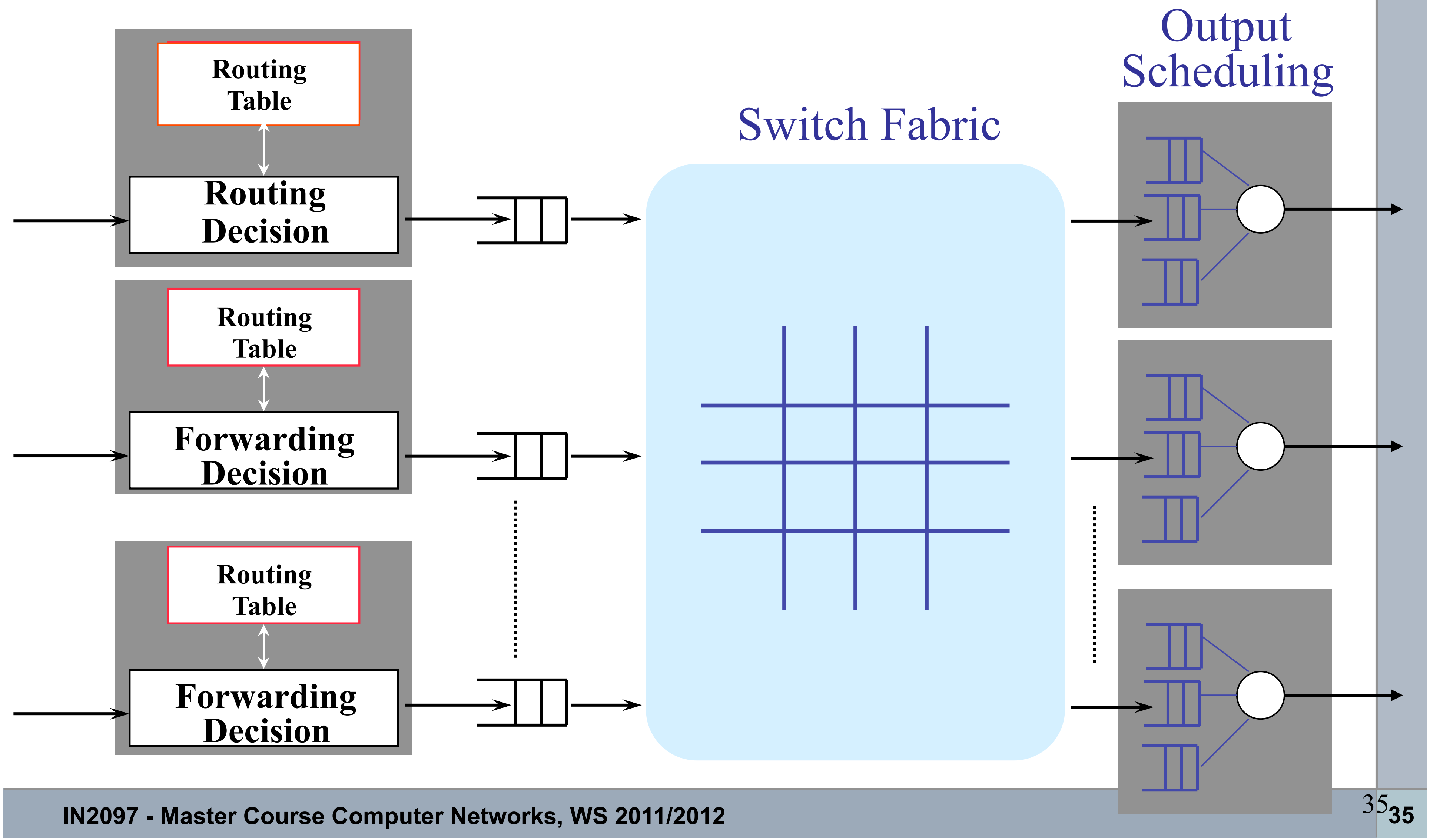
IP forwarding is per-packet processing

❑ 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?