CS 540
Computer Networks II

Sandy Wang
chwang_98@yahoo.com

5. TUNNELS

Topics

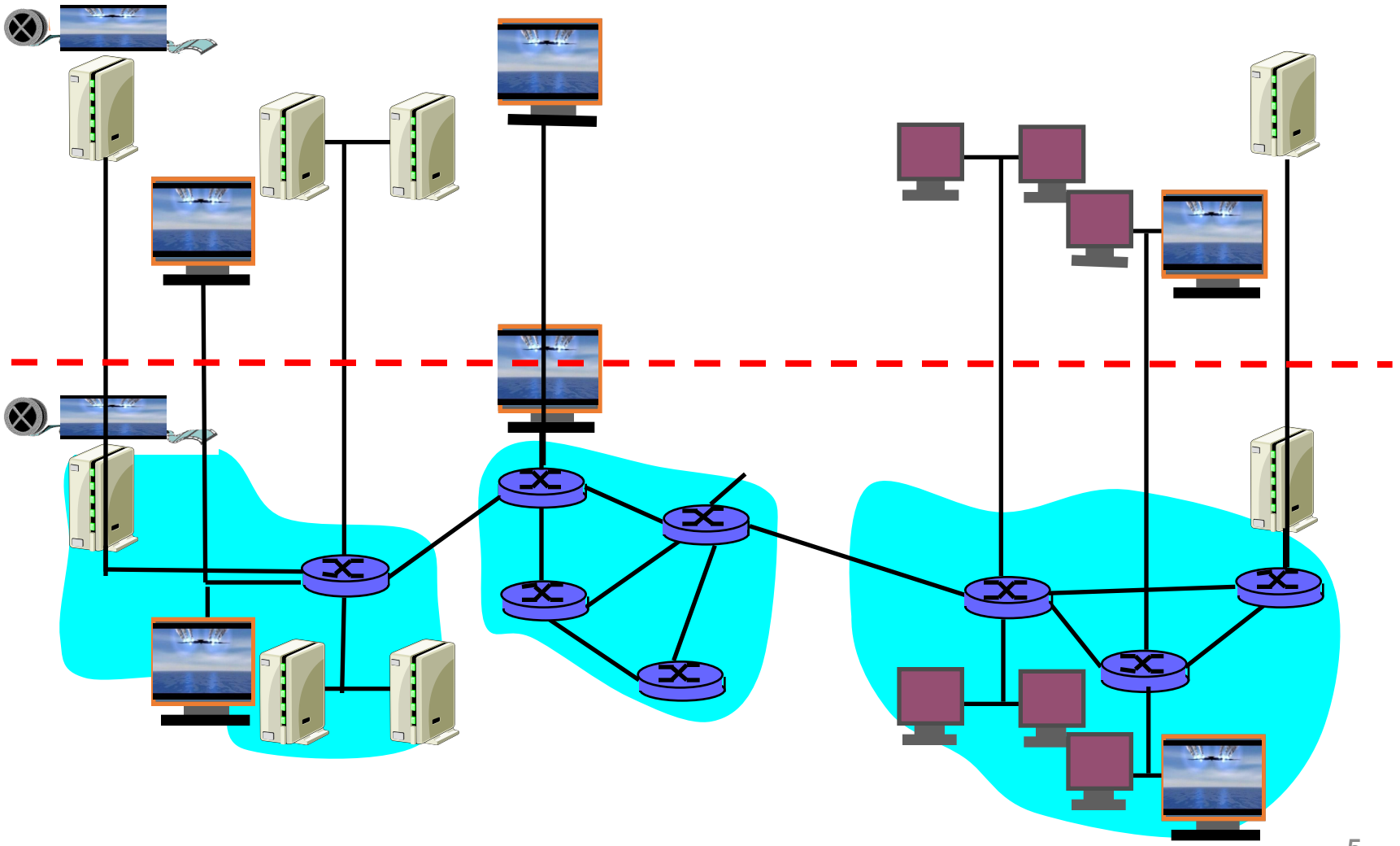
1. Overview
2. LAN Switching
3. IPv4
4. IPv6
5. **Tunnels**
6. Routing Protocols -- RIP, RIPng
7. Routing Protocols -- OSPF
8. IS-IS
9. Midterm Exam
10. BGP
11. MPLS
12. Transport Layer -- TCP/UDP
13. Congestion Control & Quality of Service (QoS)
14. Access Control List (ACL)
15. Final Exam

Reference Books

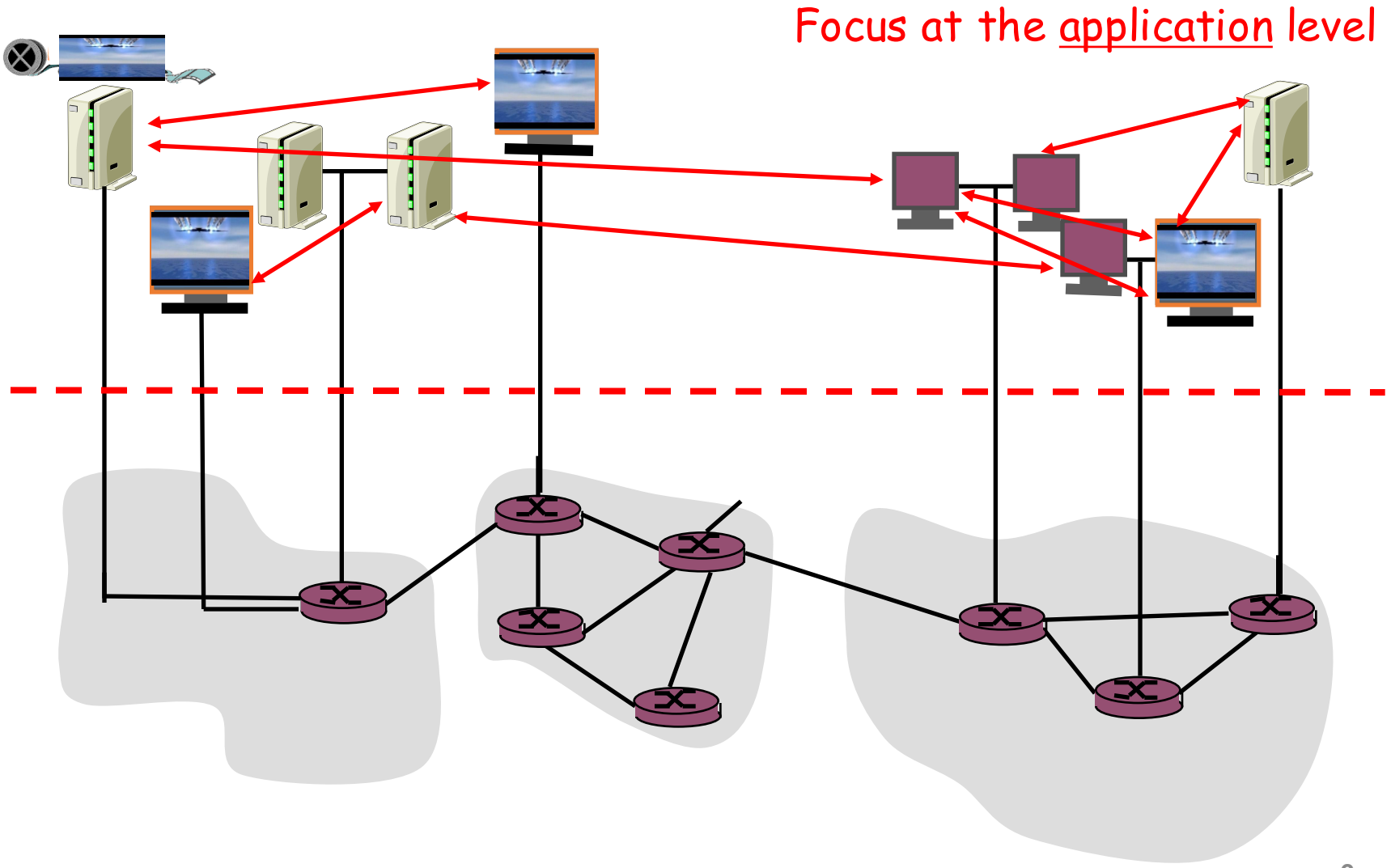
- **Cisco CCNA Routing and Switching ICND2 200-101 Official Cert Guide, Academic Edition** by Wendel Odom -- July 10, 2013**.** ISBN-13: 978-1587144882
- **The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference** by Charles M. Kozierok – October 1, 2005. ISBN-13: 978-1593270476
- **Data and Computer Communications (10th Edition) (William Stallings Books on Computer and Data Communications)** by Williams Stallings – September 23, 2013. ISBN-13: 978-0133506488

http://class.svuca.edu/~sandy/class/CS540/

Overlay Networks

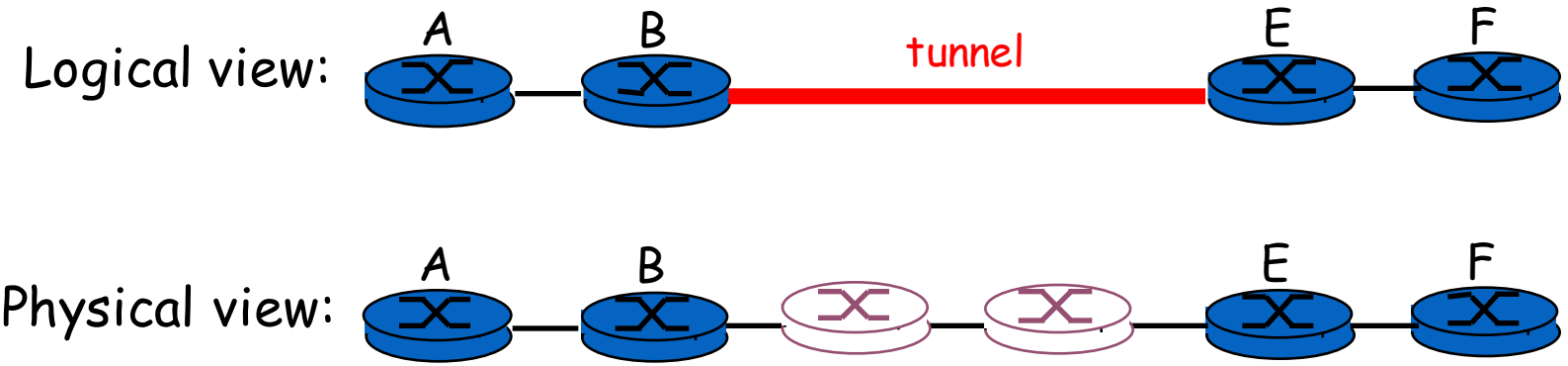


Overlay Networks



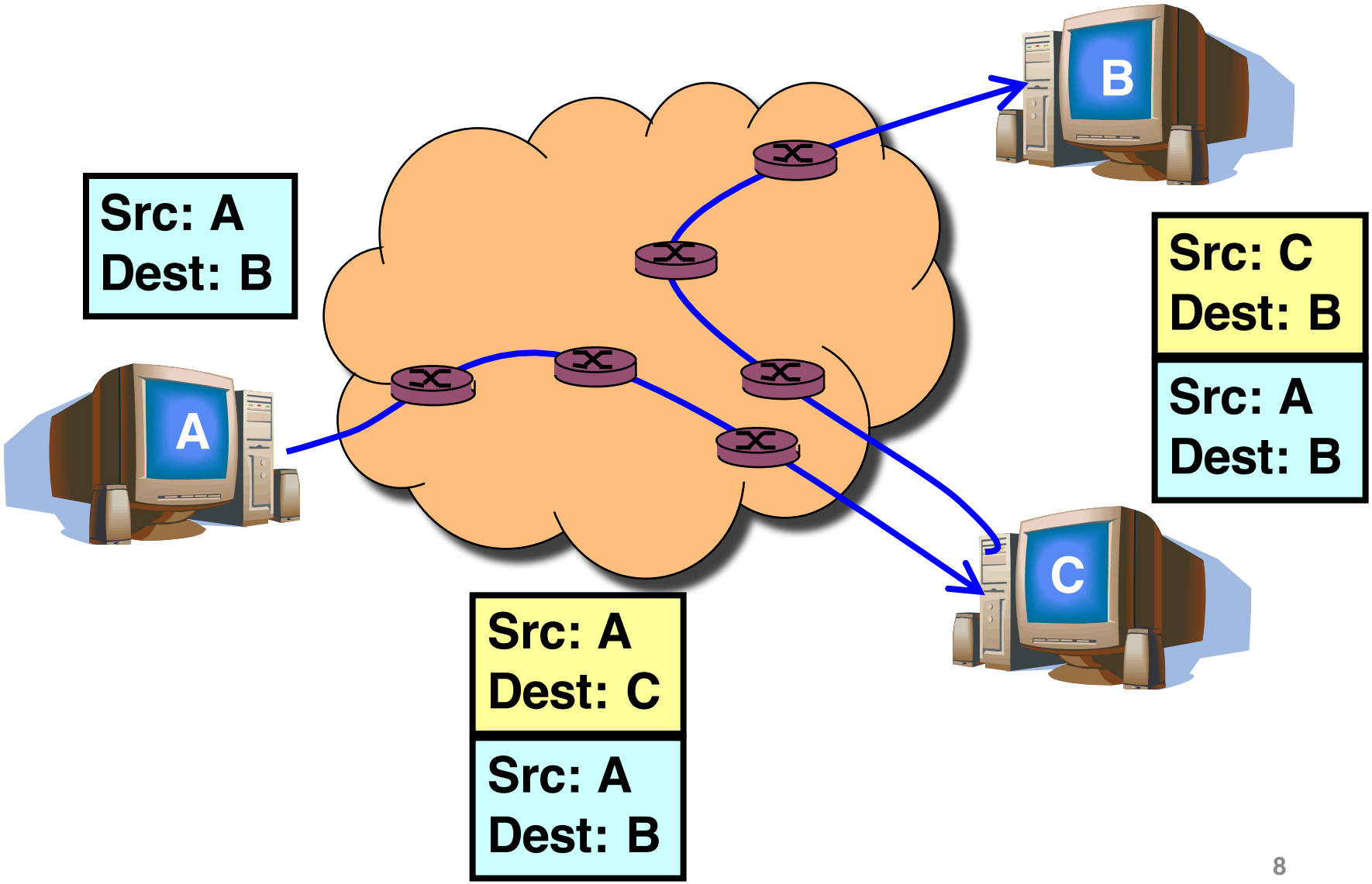
IP Tunneling to Build Overlay Links

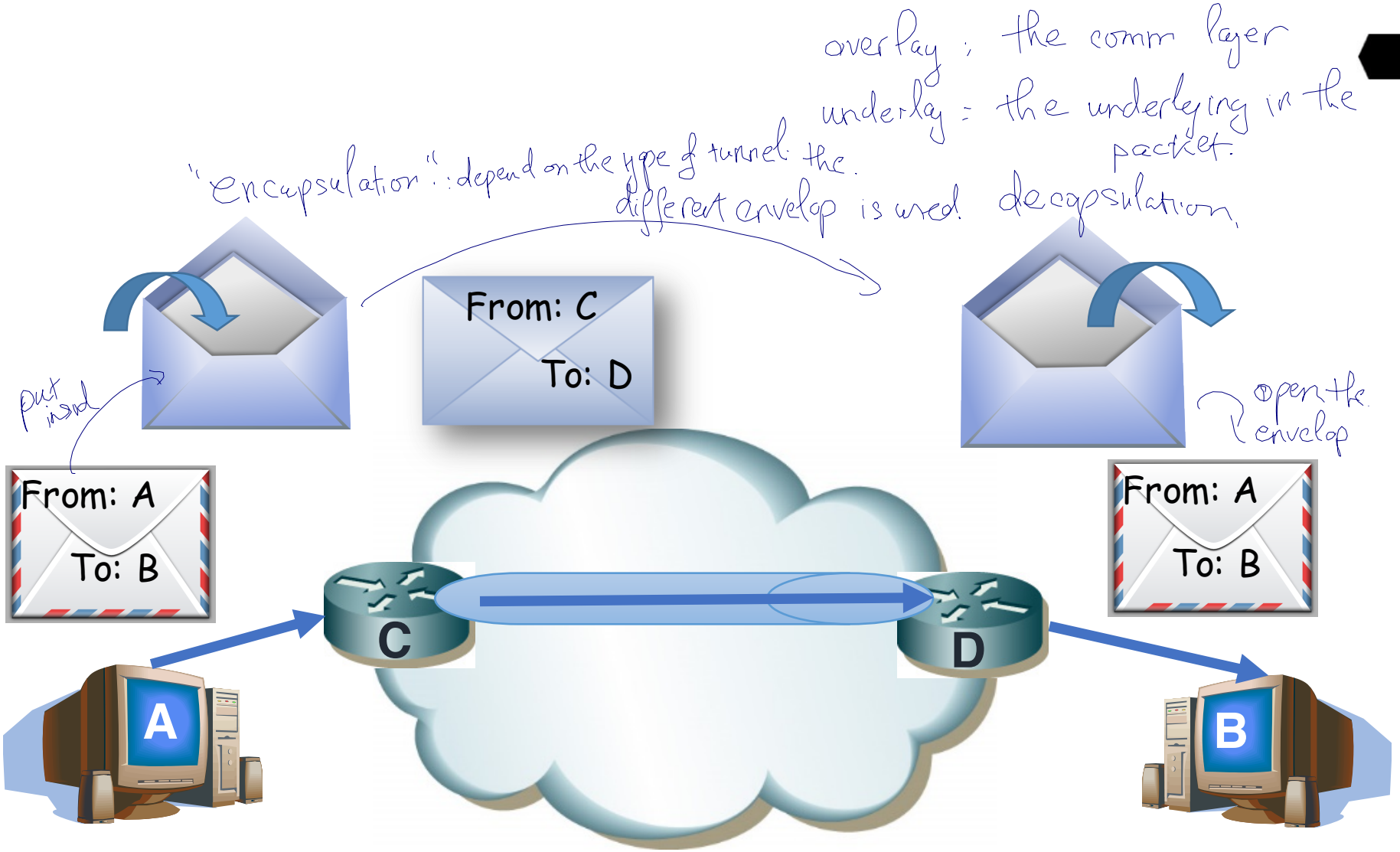
- IP tunnel is a virtual point-to-point link
  - Illusion of a direct link between two separated nodes



- Encapsulation of the packet inside an IP datagram
  - Node B sends a packet to node E
  - … containing another packet as the payload

Tunnels Between End Hosts




overlay : the comm layer
underlay : the underlying in the packet
"encapsulation" : depend on the type of tunnel the different envelop is used
decapsulation
put inside
From: C
To: D
From: A
To: B
From: A
To: B
open the envelop
C
D
A
B

Overlay Networks

- A logical network built on top of a physical network
  - Overlay links are tunnels through the underlying network
- Many logical networks may coexist at once
  - Over the same underlying network
  - And providing its own particular service

Common Uses

- Carry data over incompatible delivery-networks
- Provide a (encrypted) path through a public network
- Allowing “some kind” of traffic may lead to “any kind”

Misuse of Network Tunneling

- Pre-existing network-based security tools (firewalls, IDS) may not be able to apply the controls to the tunneled traffic
  - Evading traffic regulation
- Lack of host-based security controls
  - Defense in depth
- Inability for ingress and egress filtering
- ‘Open-ended’ tunnel may forward traffic to other internal hosts

IPSec Tunnel Mode

- The original IP packet is encrypted
- The ESP header indicates that the entire packet is the payload (IP-in-IP)
- Inserts a new IP header (next header is ESP)

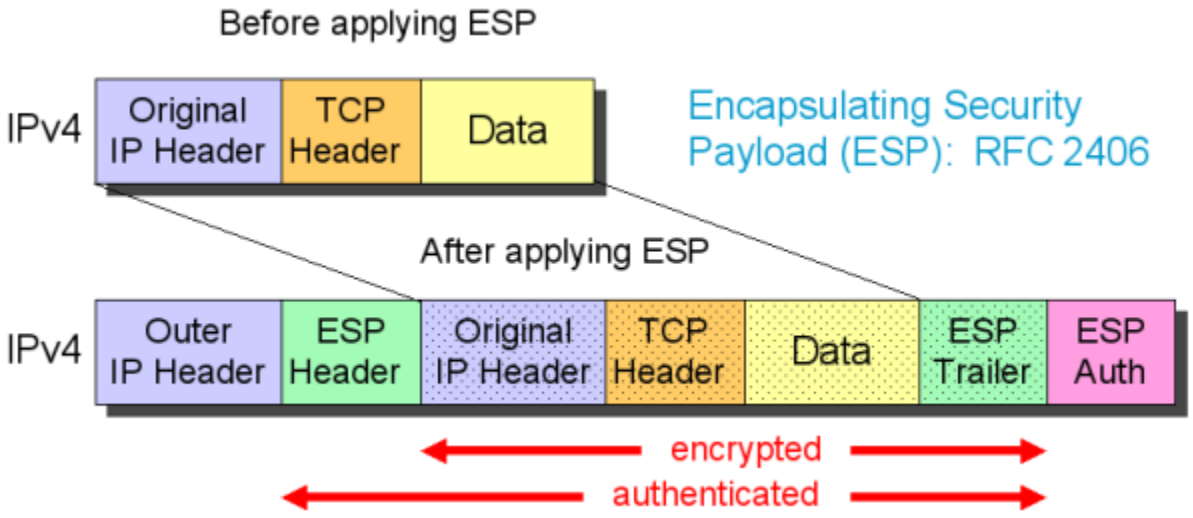


Image taken from http://www.free-it.de/archiv/talks_2005/paper-11156/paper-11156.html

IPSec Tunnel Mode

- Security services from gateway to gateway or from host to gateway over an insecure network
- The entire original packet is encrypted
  - Internal traffic behind the gateways is not protected
- Often used to implement Virtual Private Networks (IPsec VPNs)
  - Site-to-site
  - Client-to-site

GRE – Generic Routing Encapsulation

- "GRE (Generic Routing Encapsulation) specifies a protocol for encapsulation of an arbitrary protocol over another arbitrary network layer protocol" – RFC 2784 and 2890
- Point-to-point links

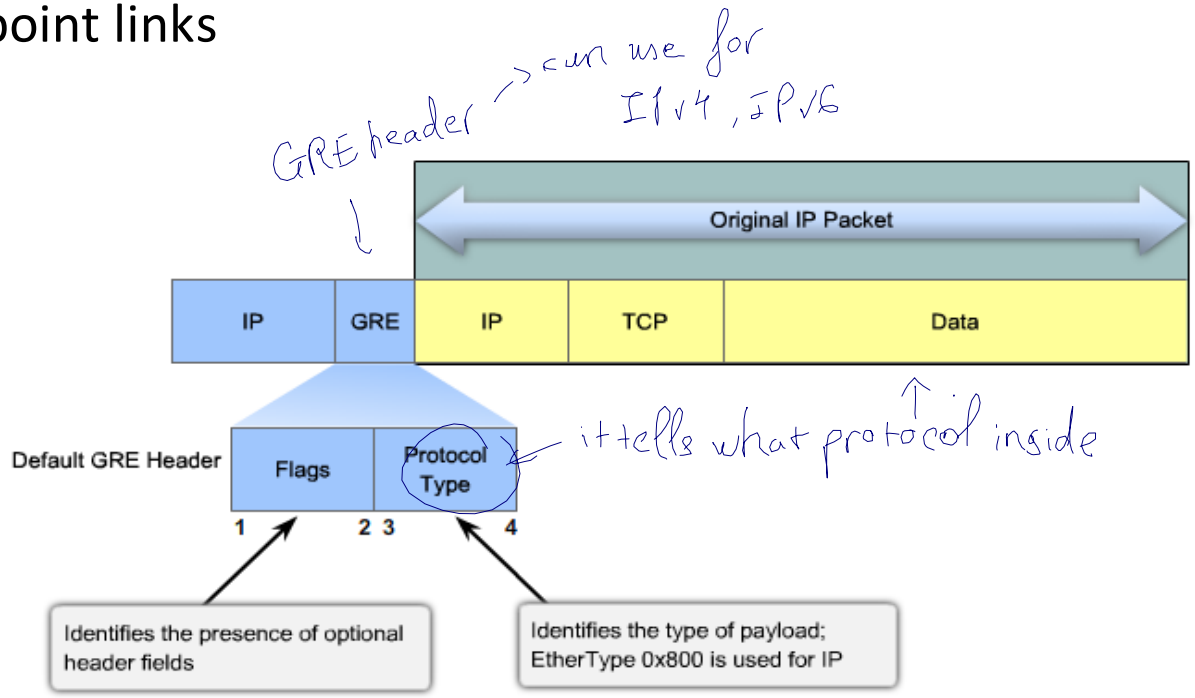
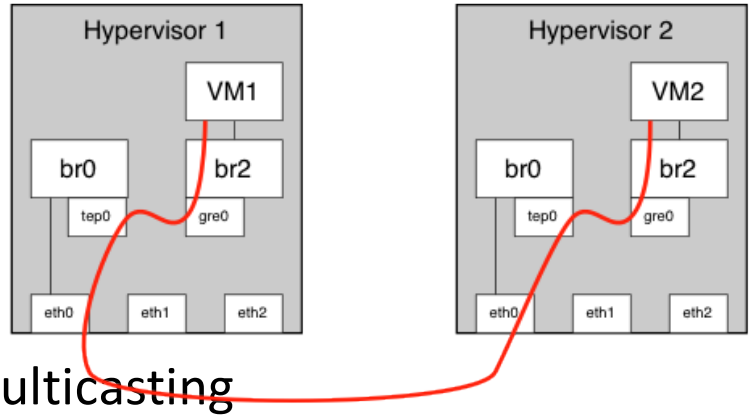
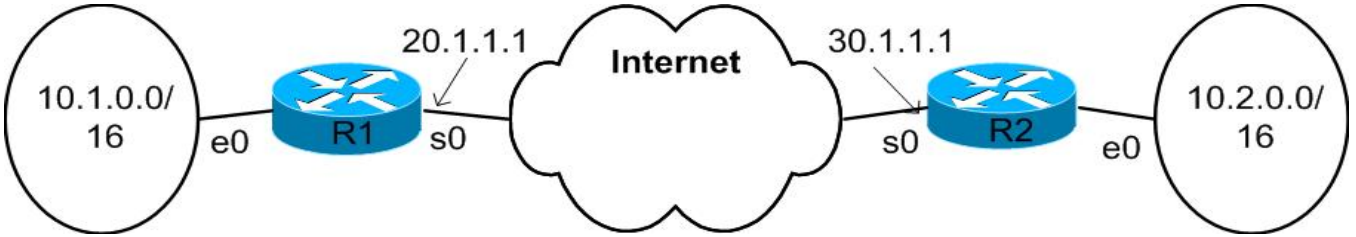


Image taken from http://netwild.ru/pptp/

GRE and IP



- Ethernet over IPv4/IPv6
  (e.g. Openstack Neutron)
- Support for tunneling broadcasting/multicasting
  - e.g. Delivering routing updates to multiple sites
- IPv4/IPv6 over IPv4/IPv6
- No default encryption/security services
  - IPSec Tunnel/Transport over GRE



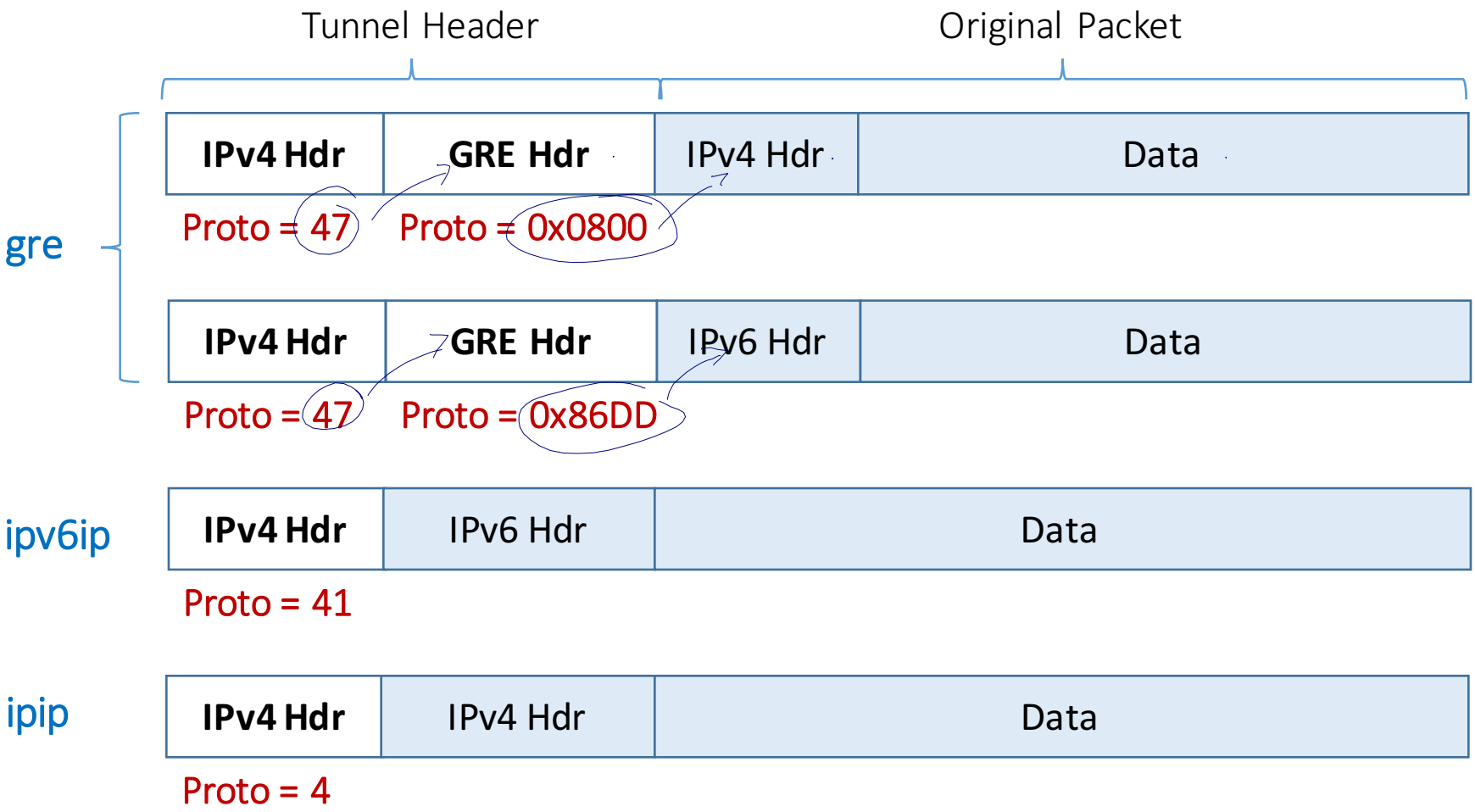
missing Tunnel Mode & Encapsulation

IP Protocol Numbers

if outside is IPv4

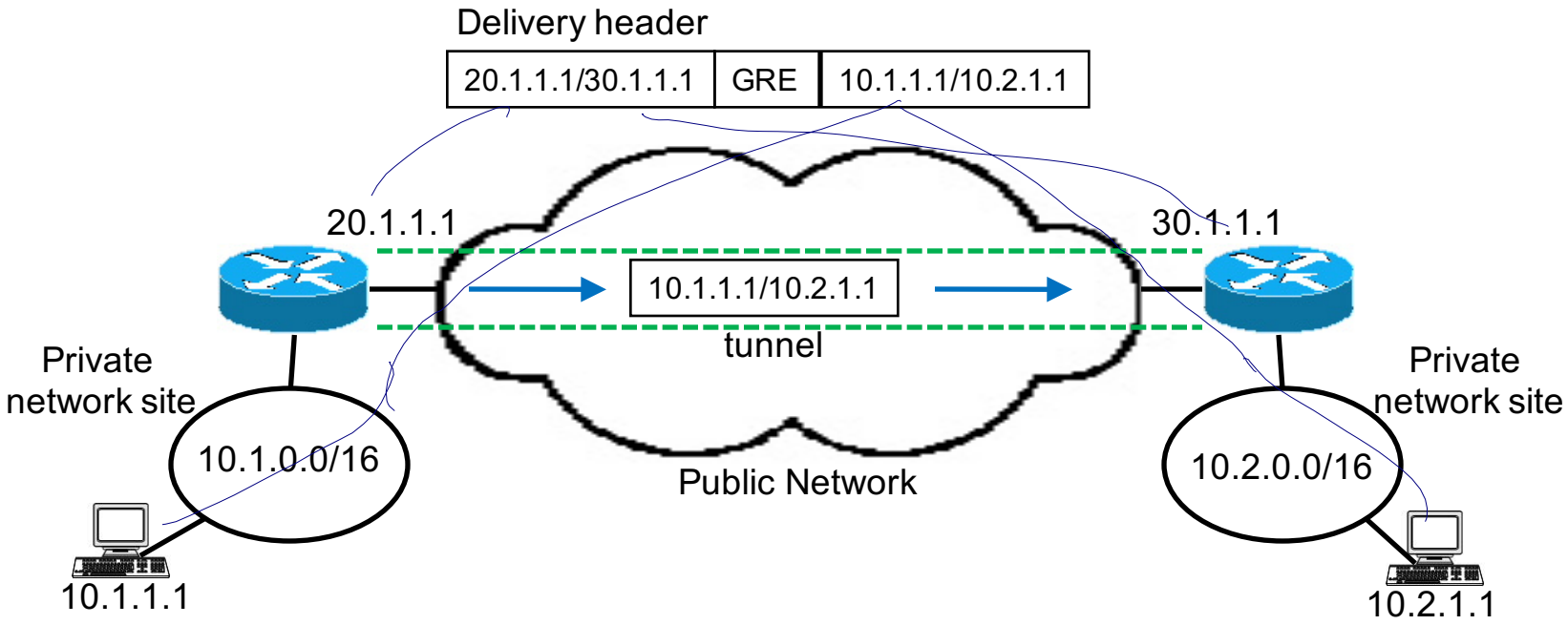
Decimal	Hex	Keyword	Protocol
4 (inside is IPv4)	0x04	IP-in-IP	IP_in_IP (encapsulation)
41 (inside is IPv6)	0x29	IPv6	IPv6 Encapsulation
47	0x2F	GRE	Generic Routing Encapsulation
50	0x32	ESP	Encapsulating Security Payload
51	0x33	AH	Authentication Header

Tunnel Mode and Encapsulation



Generic Routing Encapsulation (GRE)

- Tunneling
  - Encapsulation with delivery header
  - The addresses in the delivery header are the addresses of the head-end and the tail-end of the tunnel



Generic Routing Encapsulation (GRE)

4 bytes header.

- RFC 2784 – Generic Routing Encapsulation

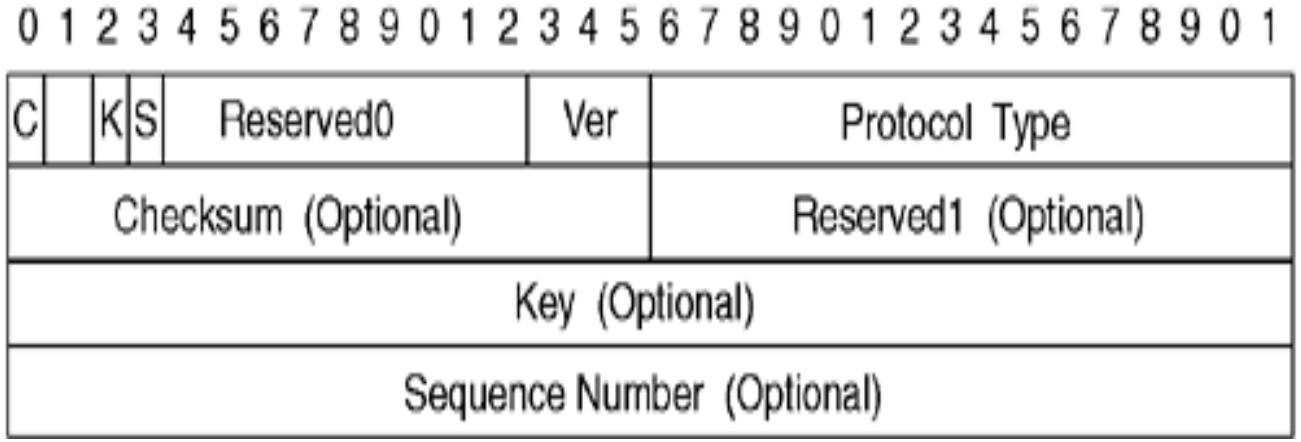
```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|       Reserved0          | Ver |         Protocol Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Checksum (optional)      |      Reserved1 (Optional)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

must have
others are optional.

Protocol Type – EtherType in RFC 1700. Examples below:

EtherType	Protocol
0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x86DD	Internet Protocol Version 6 (IPv6)

RFC 2890 – Key and Sequence Number Extensions to GRE

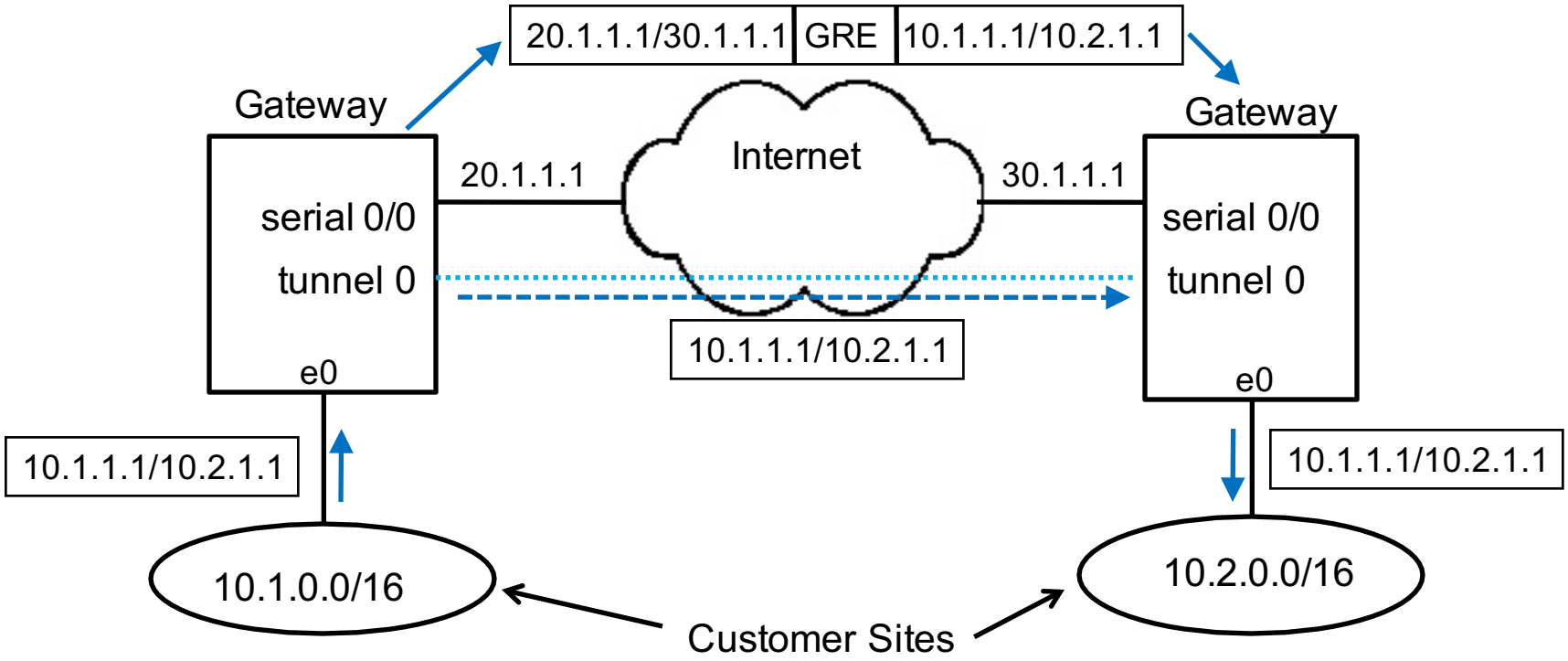


Key – Identify an individual traffic flow within a tunnel. Packets belonging to a traffic flow are encapsulated using the same Key value and the decapsulating tunnel endpoint identifies packets belonging to a traffic flow based on the Key Field value.

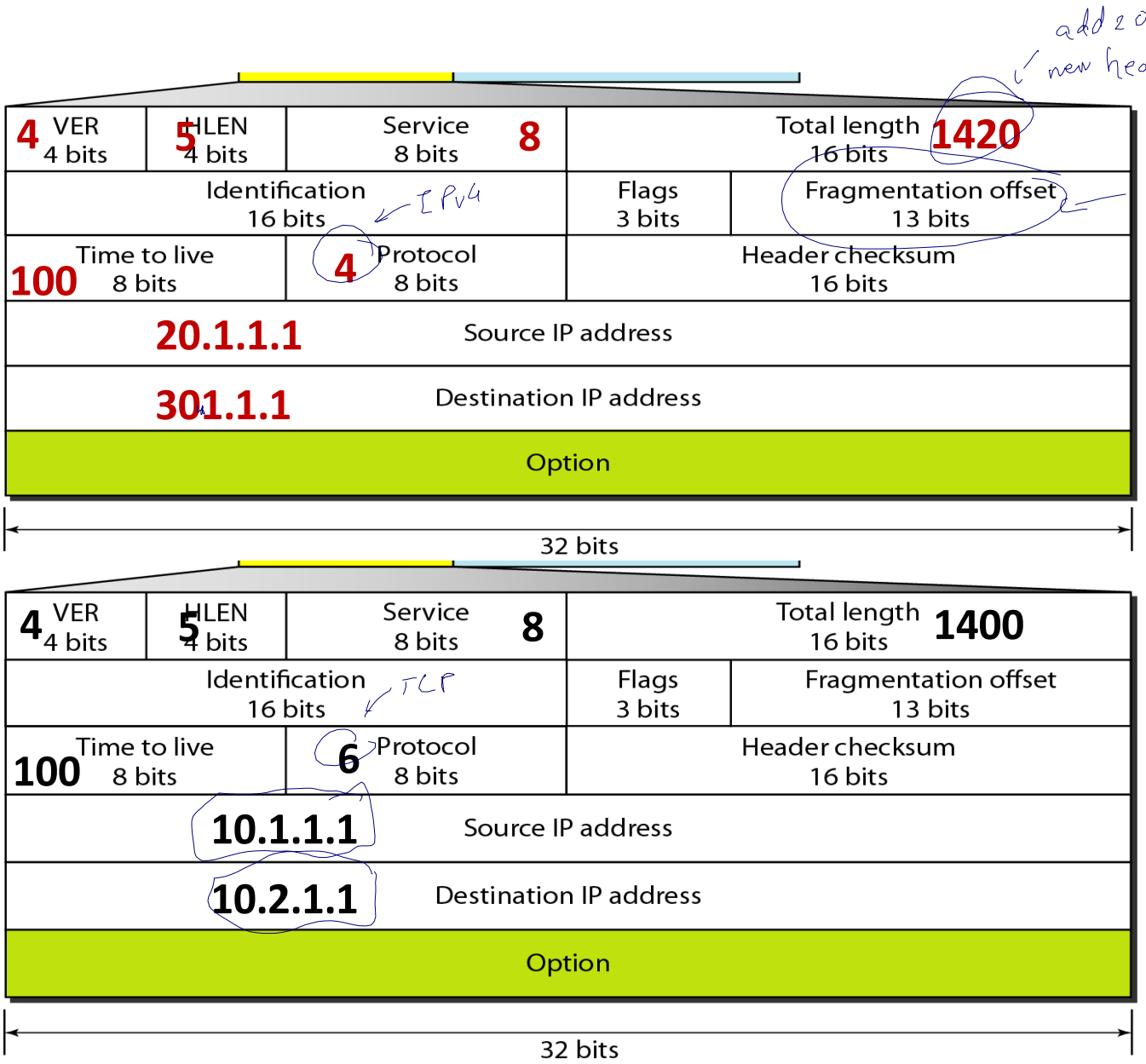
Sequence Number -- inserted by the encapsulator when Sequence Number Present Bit is set. The Sequence Number MUST be used by the receiver to establish the order

Generic Routing Encapsulation (GRE)

- IP access of the tunnel through the tunnel interface



Encapsulation



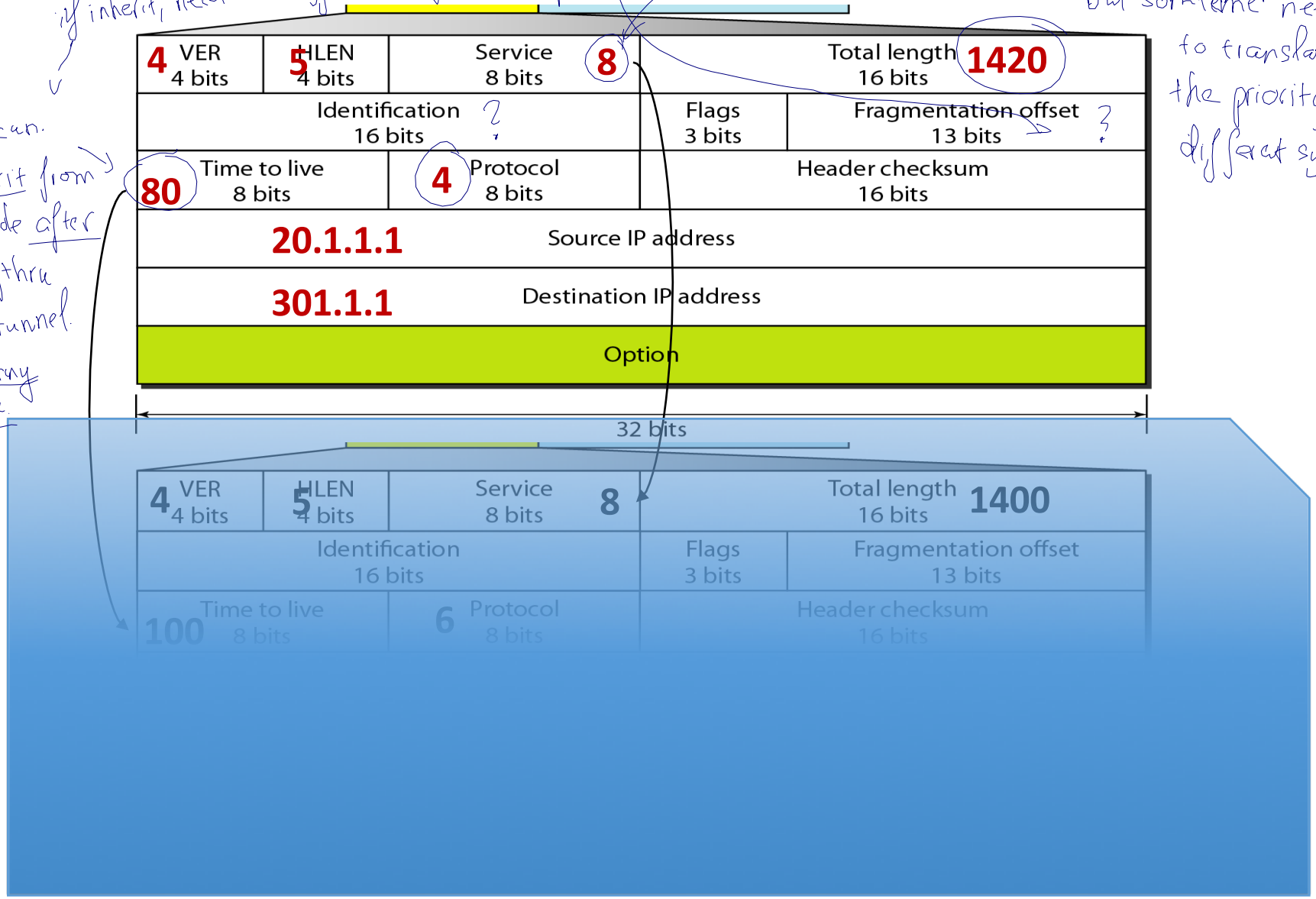
Decapsulation

because of additional header, there might need fragmentation.

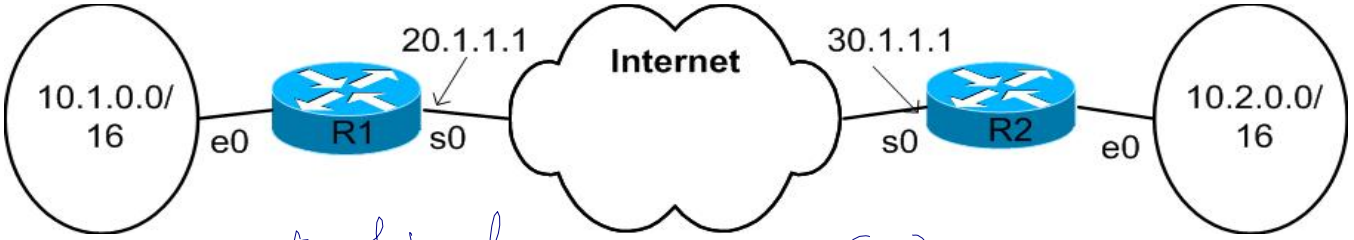
if inherit, need to modify the TTL of the inner packet

it could be the same or different. from original priority but sometime needs to translate the priority of different systems

it can inherit from outside after going thru the tunnel OR any value



Generic Routing Encapsulation (GRE)



type of tunnel.

Symmetric conn.

```
interface tunnel0
  mode GRE
  tunnel source 20.1.1.1
  tunnel destination 30.1.1.1
!
ip route 10.2.0.0 255.255.0.0 tunnel0
```

```
interface tunnel0
  mode GRE
  tunnel source 30.1.1.1
  tunnel destination 20.1.1.1
!
ip route 10.1.0.0 255.255.0.0 tunnel0
```

if want to reach this IP use this tunnel0

Routing table of R1

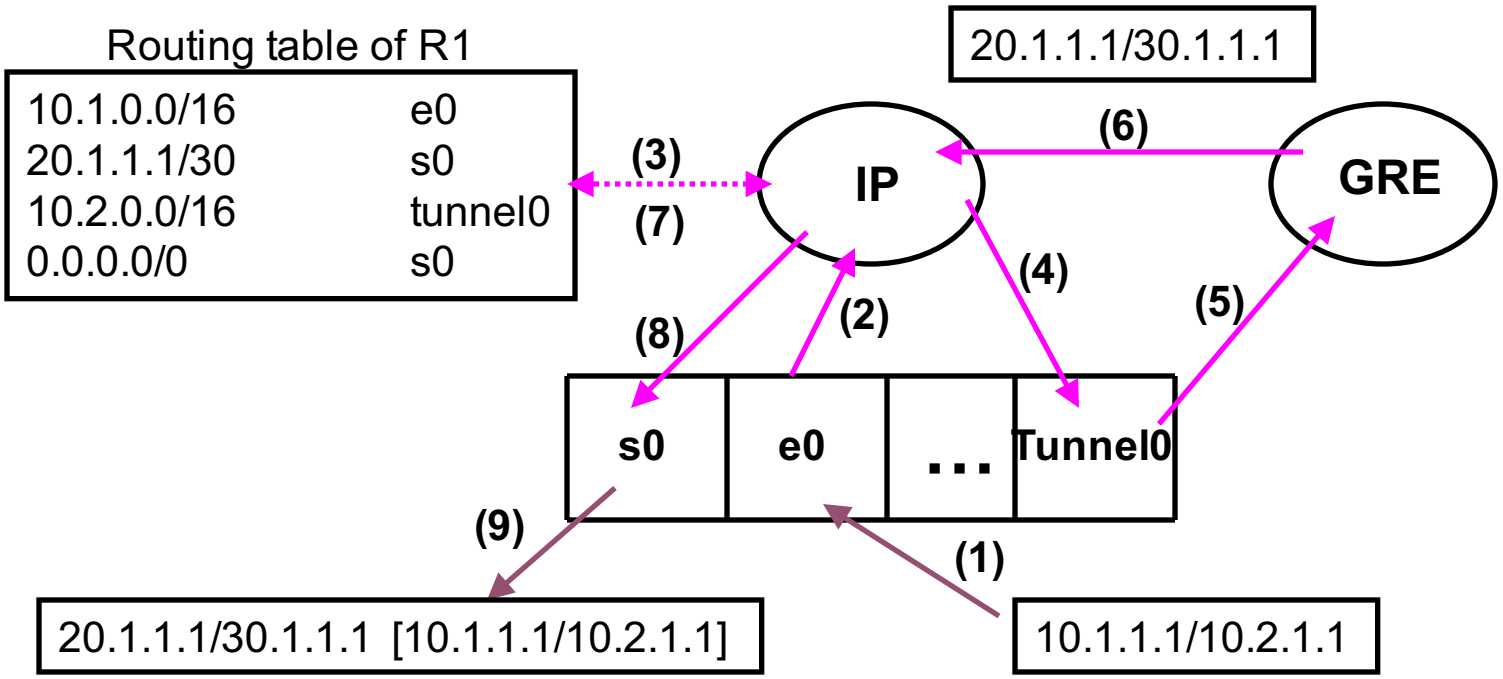
10.1.0.0/16	e0
20.1.1.1/30	s0
10.2.0.0/16	tunnel0
0.0.0.0/0	s0

Routing table of R2

10.2.0.0/16	e0
30.1.1.1/30	s0
10.1.0.0/16	tunnel0
0.0.0.0/0	s0

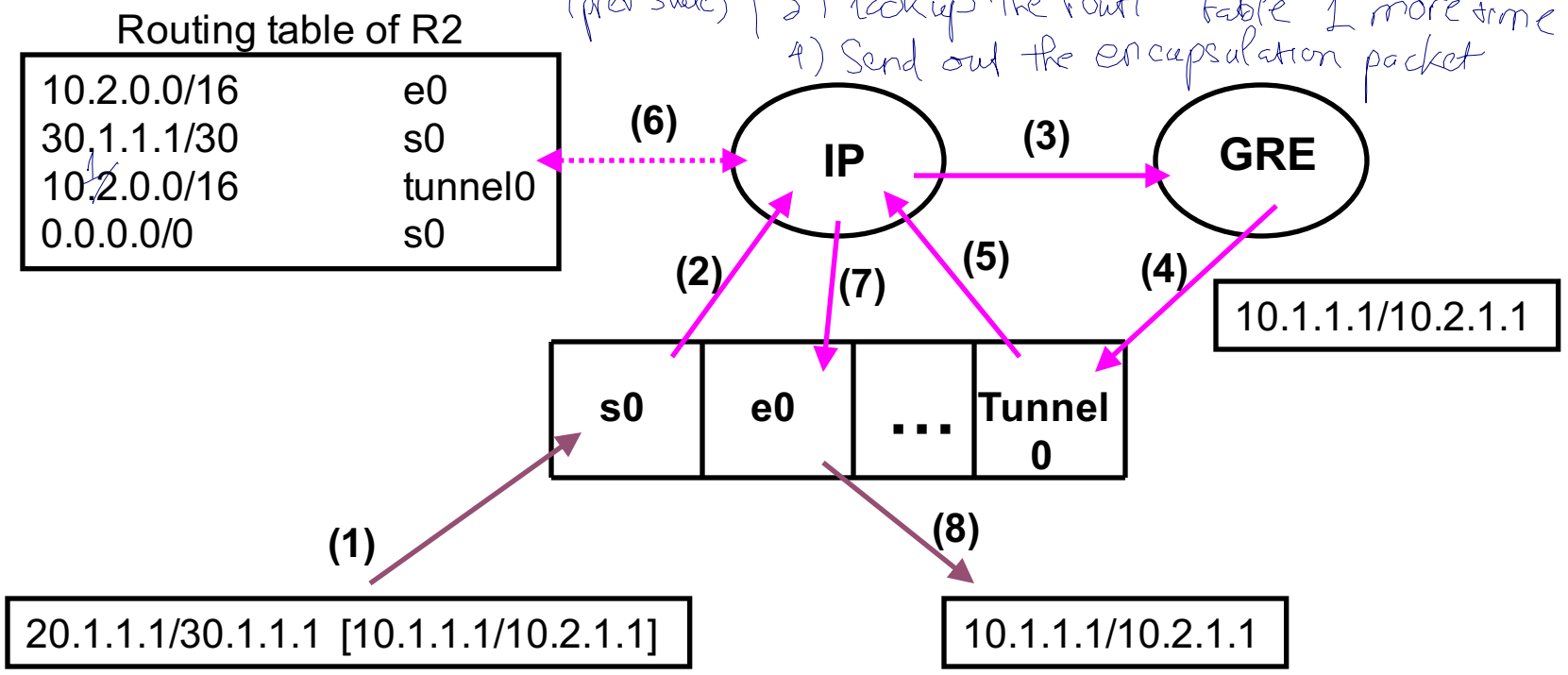
Generic Routing Encapsulation (GRE)

- Tunneling mechanism at IP
  - Outbound traffic



Generic Routing Encapsulation (GRE)

- Inbound traffic



IP Tunnel Example -- Linux

ip [OPTIONS] tunnel { COMMAND | help }

ip tunnel { add | change | del | show | prl } [NAME]

[mode MODE] [remote ADDR] [local ADDR]

[[i|o]seq] [[i|o]key KEY] [[i|o]csum]]

[encaplimit ELIM] [ttl TTL]

[tos TOS] [flowlabel FLOWLABEL]

[prl-default ADDR] [prl-nodefault ADDR] [prl-delete ADDR]

[[no]pmtudisc] [dev PHYS_DEV]

IPv6 over IPv4 Transition Mechanisms

- Tunnel Brokers provide a network tunneling service
- 6in4 – IPv6 over IPv4

- 4in6 – IPv4 over IPv6
- ISATAP
- Teredo – IPv6 over UDP over IPv4
- …and others

IP Tunnel Example -- Linux

MODE := { ipip | gre | sit | isatap | ip6ip6 | ipip6 | ip6gre | any }

ADDR := { IP_ADDRESS | any }

TOS := { STRING | 00..ff | inherit | inherit/STRING | inherit/00..ff}

ELIM := { none | 0..255 }

TTL := { 1..255 | inherit }

KEY := { DOTTED_QUAD | NUMBER }

TIME := NUMBER[s|ms]

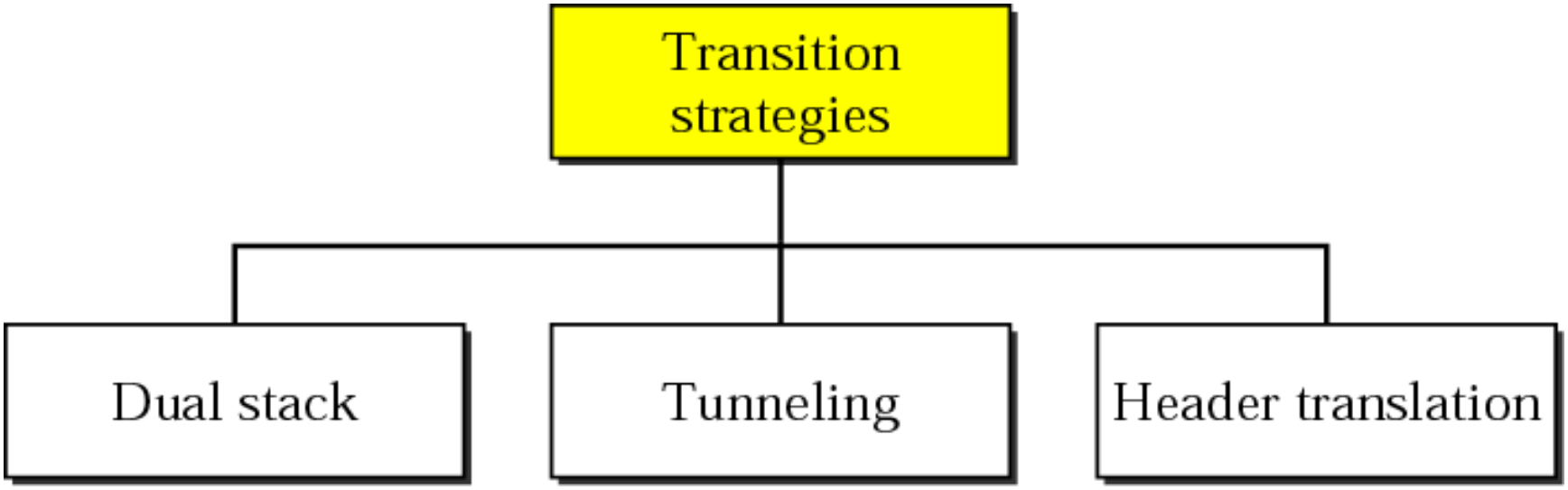
Transformation From IPv4 to IPv6

Three strategies have been devised by the IETF to provide for a smooth transition from IPv4 to IPv6.

The topics discussed in this section include:

- *Dual Stack*
- *Tunneling*
- *Header Translation*

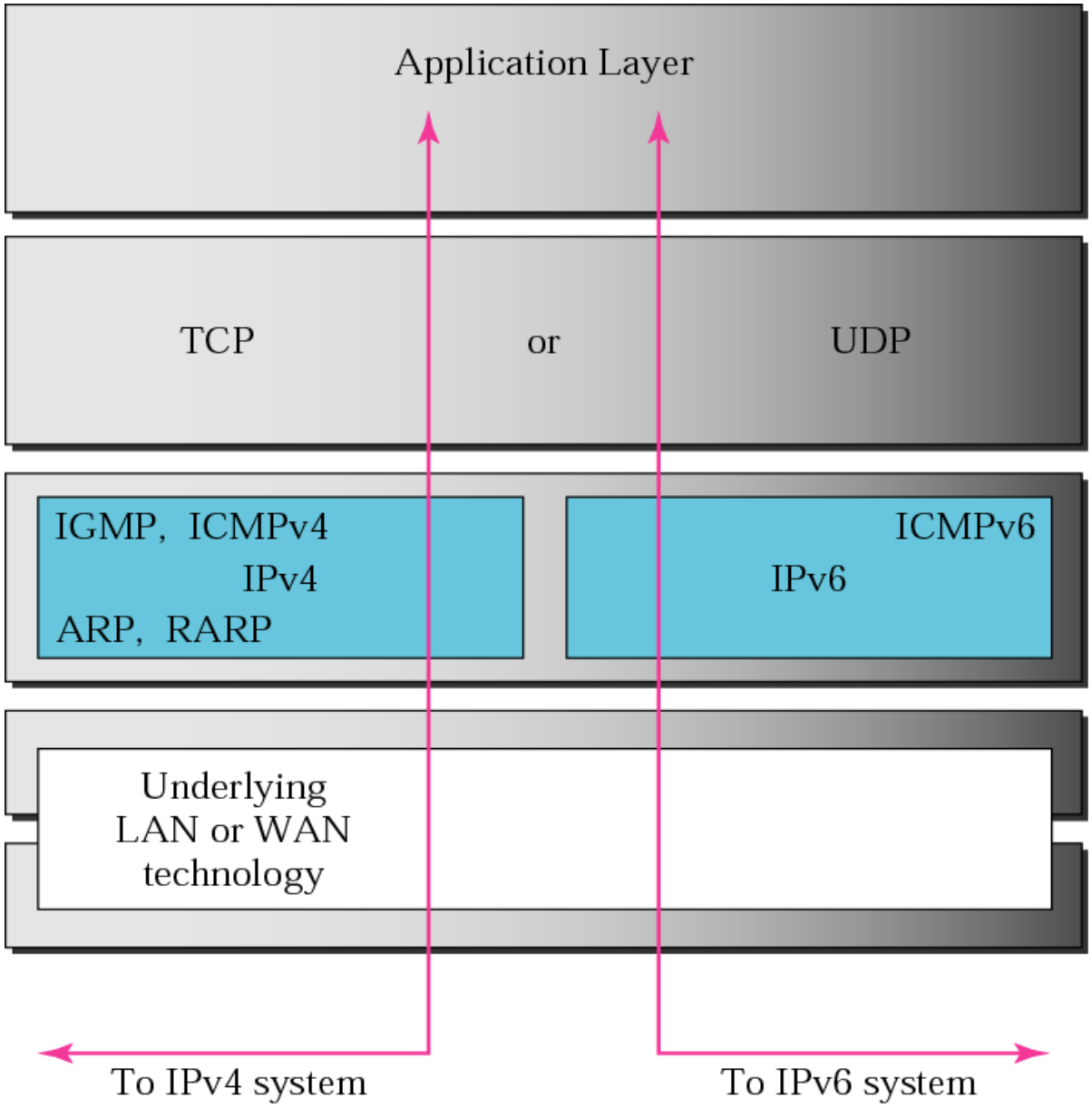
Three transition strategies



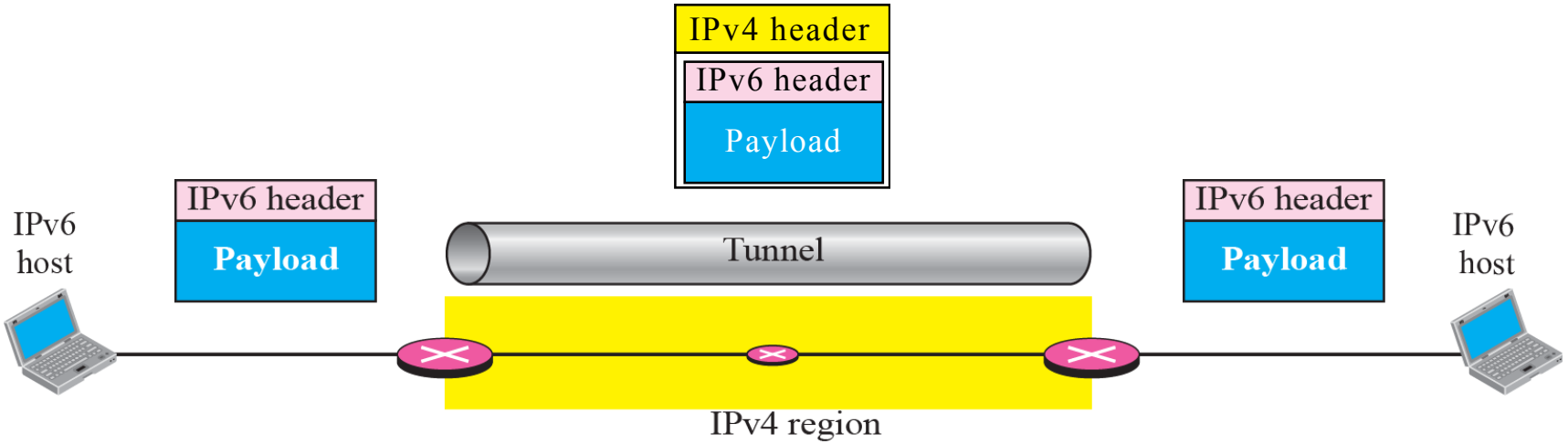
speak different langs

have on different people

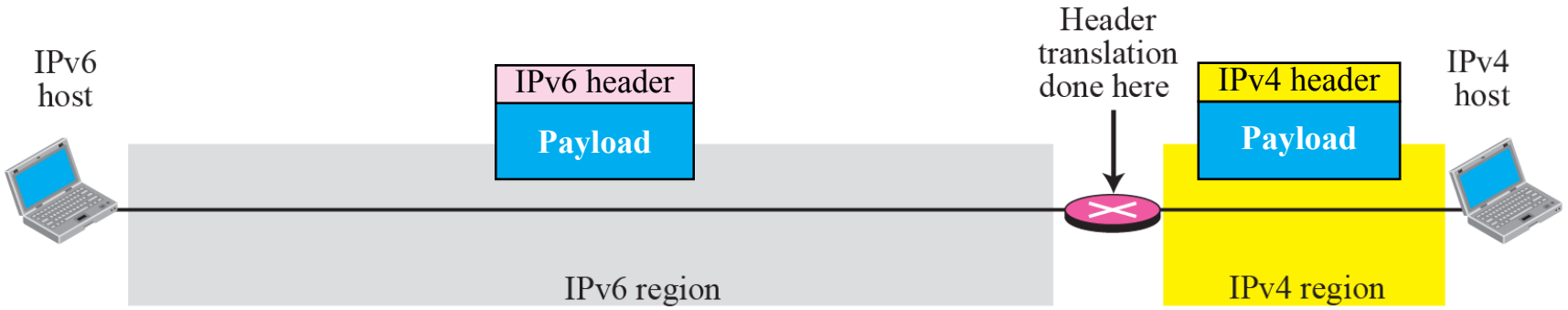
Dual stack


Application Layer
TCP
or
UDP
IGMP, ICMPv4
IPv4
ARP, RARP
ICMPv6
IPv6
Underlying
LAN or WAN
technology
To IPv4 system
To IPv6 system

Tunneling strategy



Header translation strategy



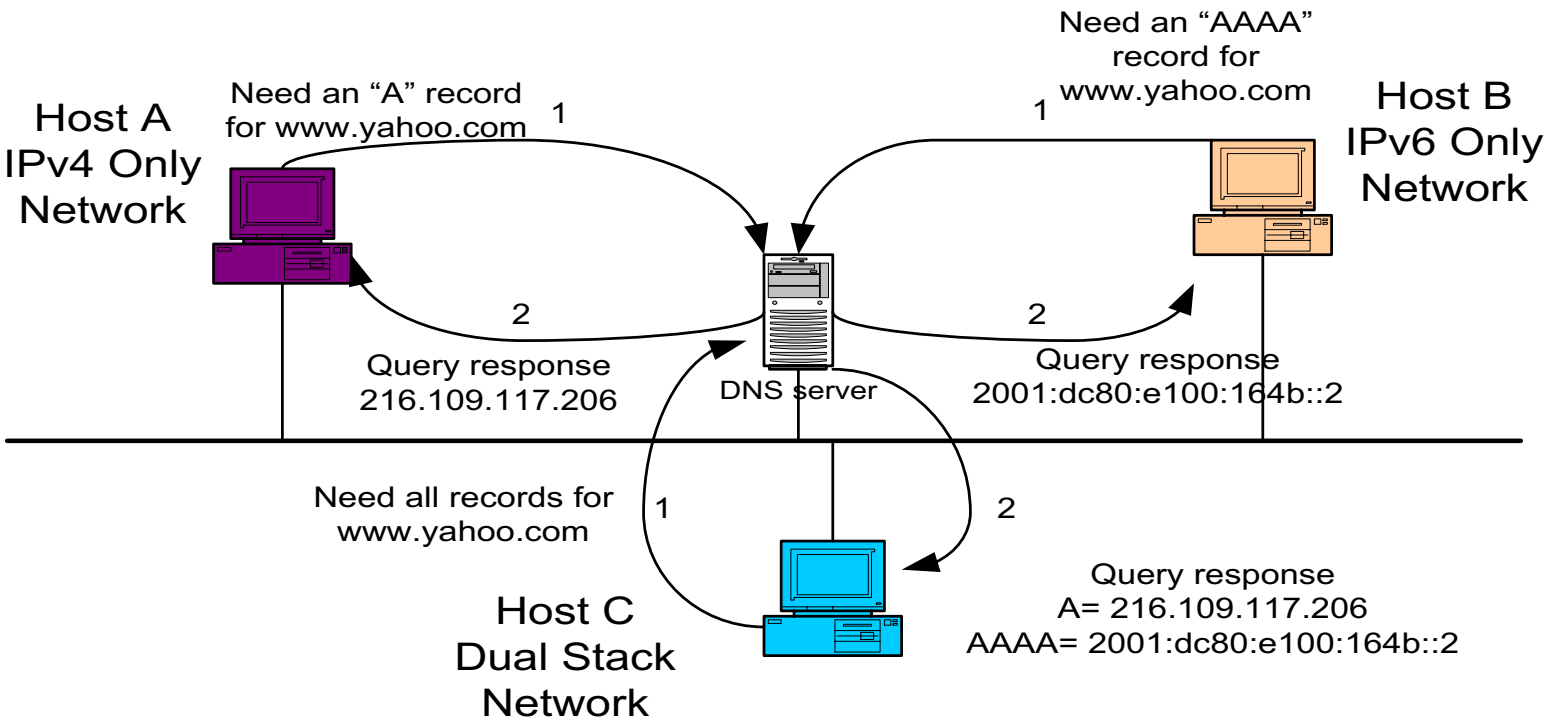
Naming Services

- DNS must be included in transition strategy
- Resolving Names:
  - IPv4 specifies “A” records
  - IPv6 specifies “AAAA” records
- Applications should be aware of both records
- Will require development update and thorough testing
- Tools like “Scrubber” by Sun make it easy

Naming Services

for dual stack usage (know who you speak to)

Querying DNS server



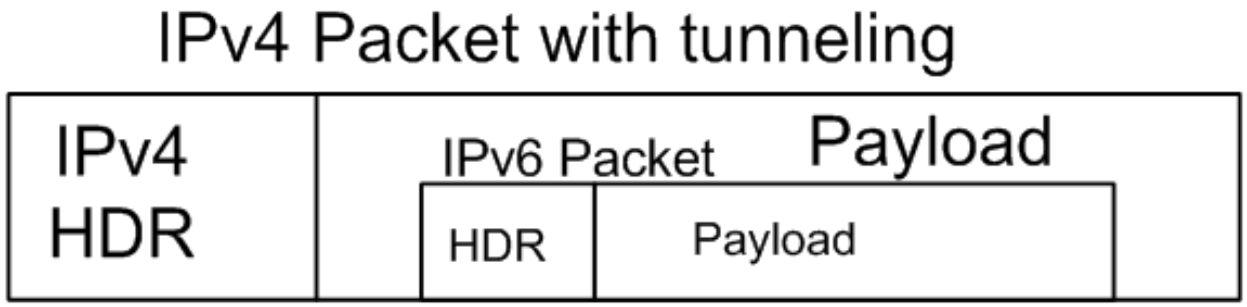
IPv6 over IPv4 Transition Mechanisms

- Tunnel Brokers provide a network tunneling service
- 6in4 – IPv6 over IPv4

- 4in6 – IPv4 over IPv6
- ISATAP
- Teredo – IPv6 over UDP over IPv4
- …and others

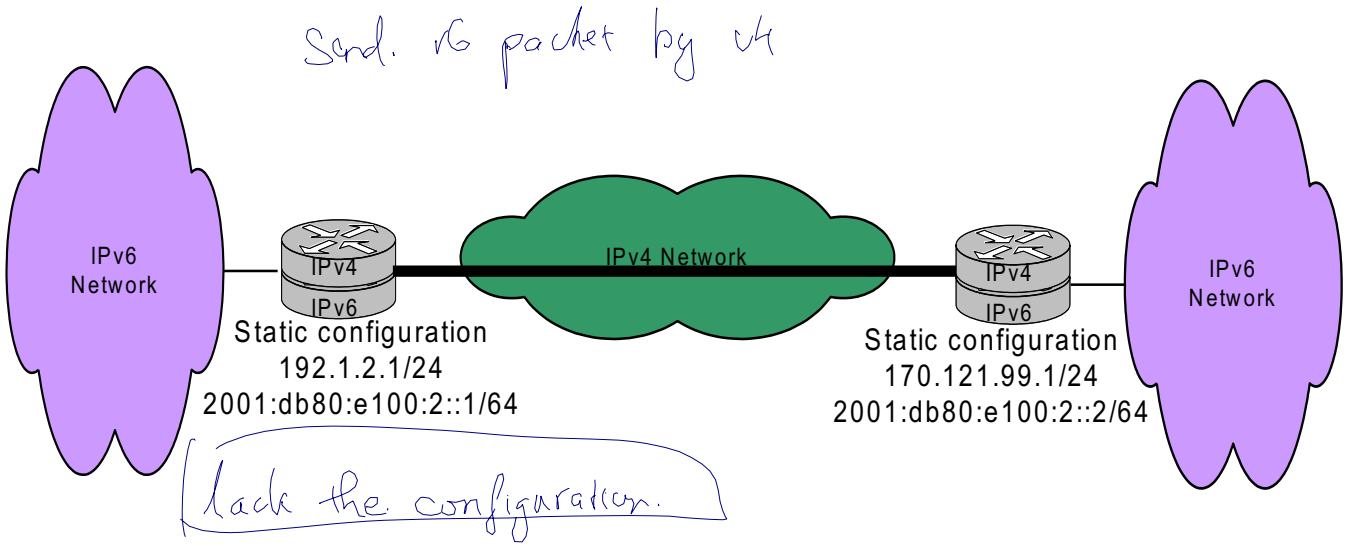
Manually Configured Tunnels

- Manually configured tunnels are logical tunnels formed when one protocol version packet is encapsulated in the payload of another version packet
- e.g. IPv4 encapsulated in IPv6 or IPv6 encapsulated in IPv4

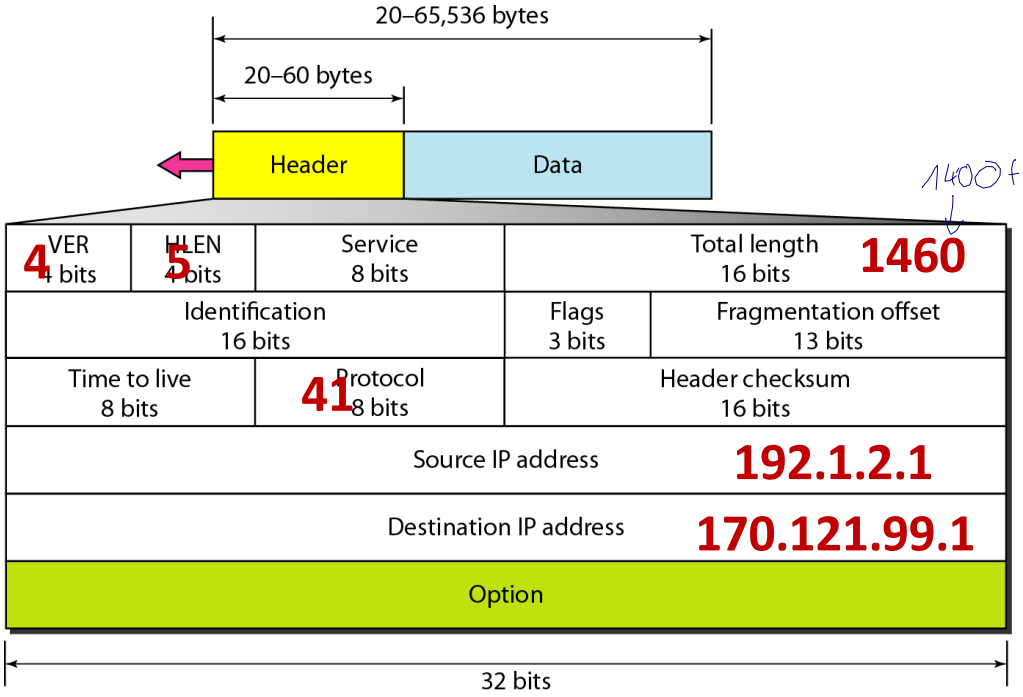


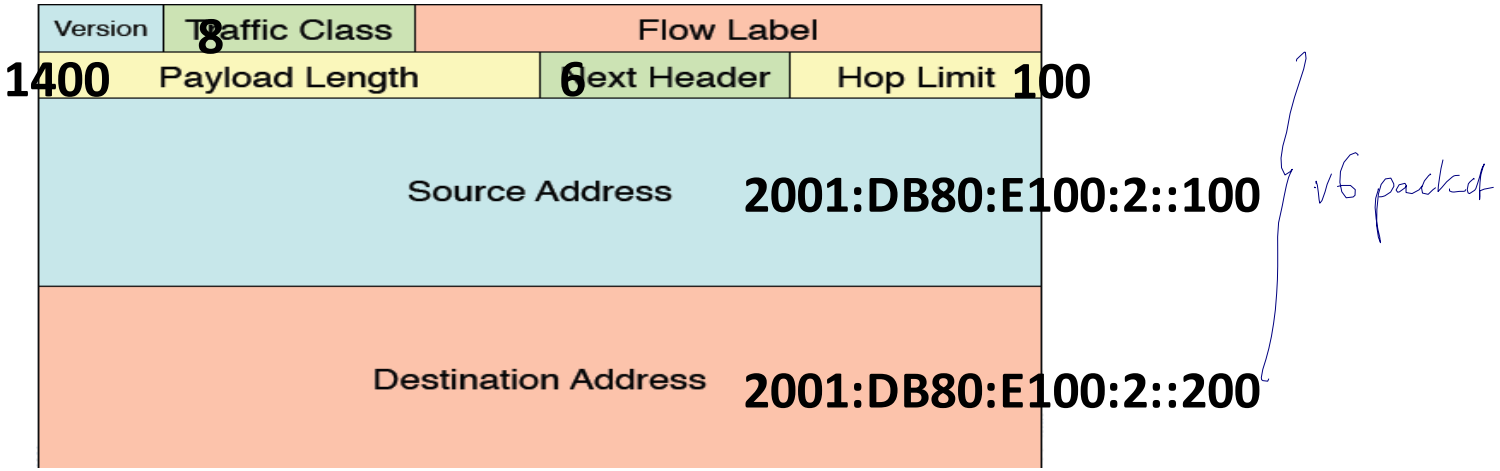
Configured Tunnel-building

- Configured tunnels require static IPv4 addresses
- Configured tunnels are generally setup and maintained by a network administrator
- Configured tunnels are a proven IPv6 deployment technique and provide stable links



Encapsulation





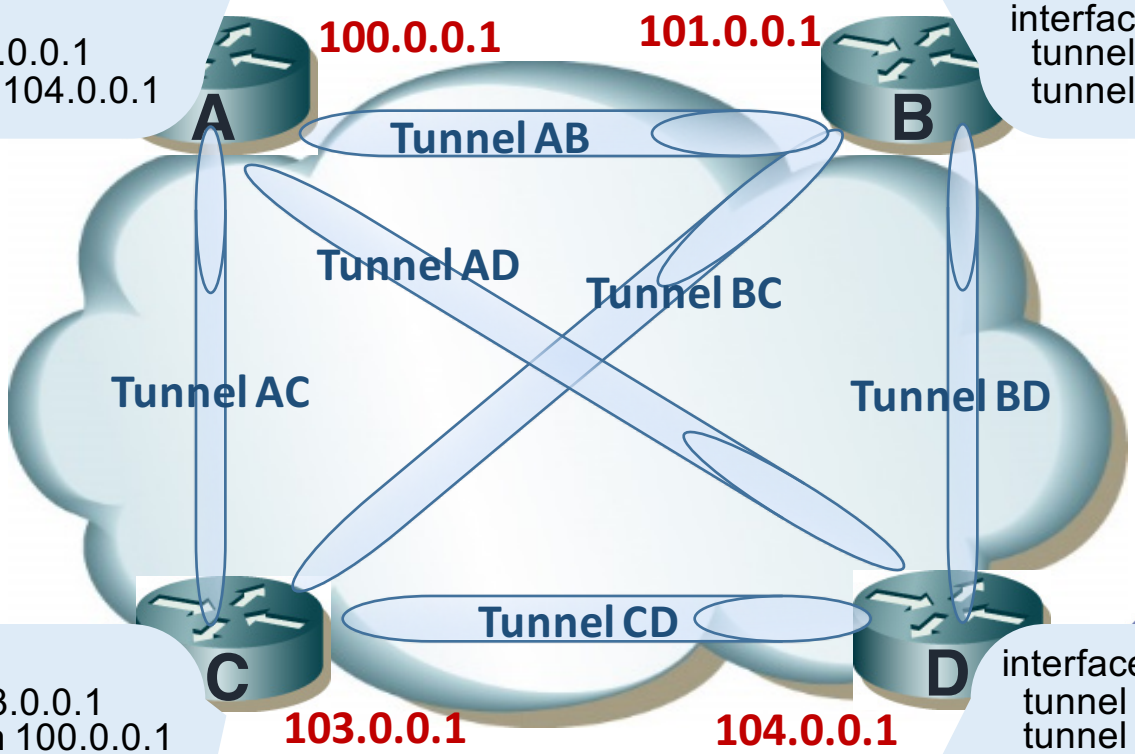
Potential Tunnel Issues

- MTU fragmentation
- ICMPv4 error handling → only send ICMP back to the tunnel src, The origin does not know!
- Filtering protocol 41 ← does not know what inside, so drop the tunnel packet
- NAT (Network Address Translation)

```
interface tunnelAB
  tunnel source 100.0.0.1
  tunnel destination 101.0.0.1
!
interface tunnelAC
  tunnel source 100.0.0.1
  tunnel destination 103.0.0.1
!
interface tunnelAD
  tunnel source 100.0.0.1
  tunnel destination 104.0.0.1
```

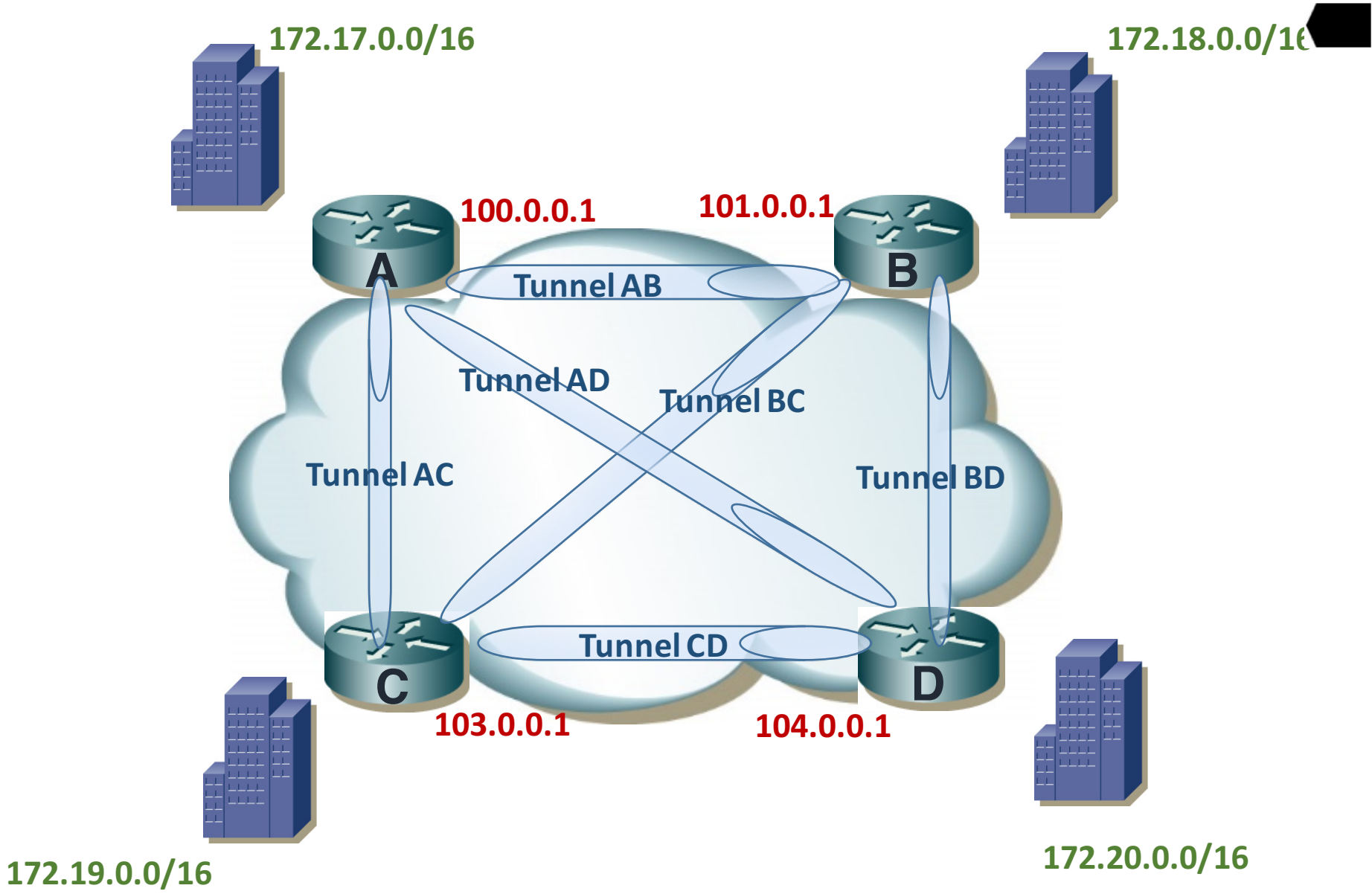
0.0/16

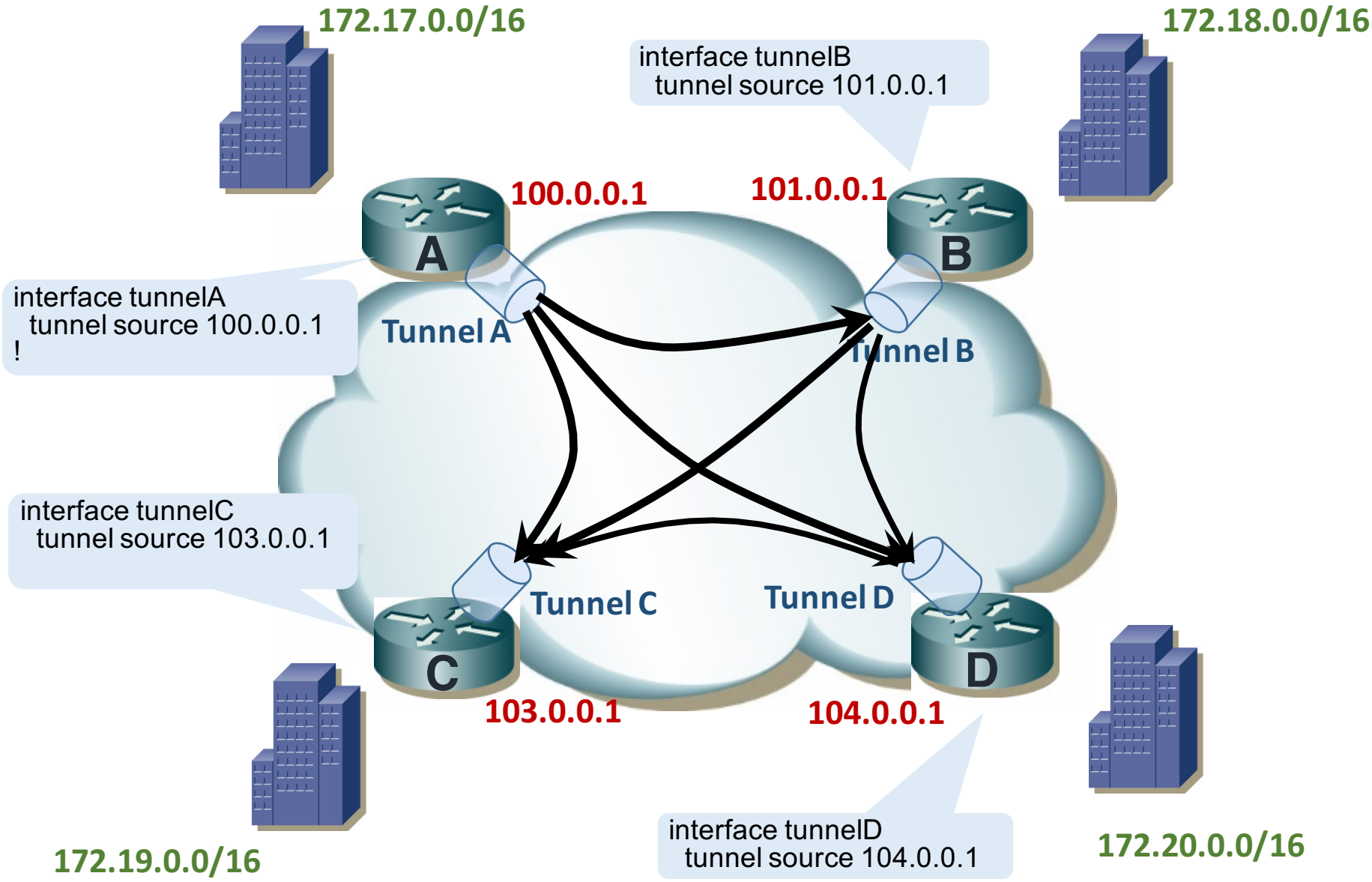
```
interface tunnelBA
  tunnel source 101.0.0.1
  tunnel destination 100.0.0.1
!
interface tunnelBC
  tunnel source 101.0.0.1
  tunnel destination 103.0.0.1
!
interface tunnelBD
  tunnel source 101.0.0.1
  tunnel destination 104.0.0.1
```

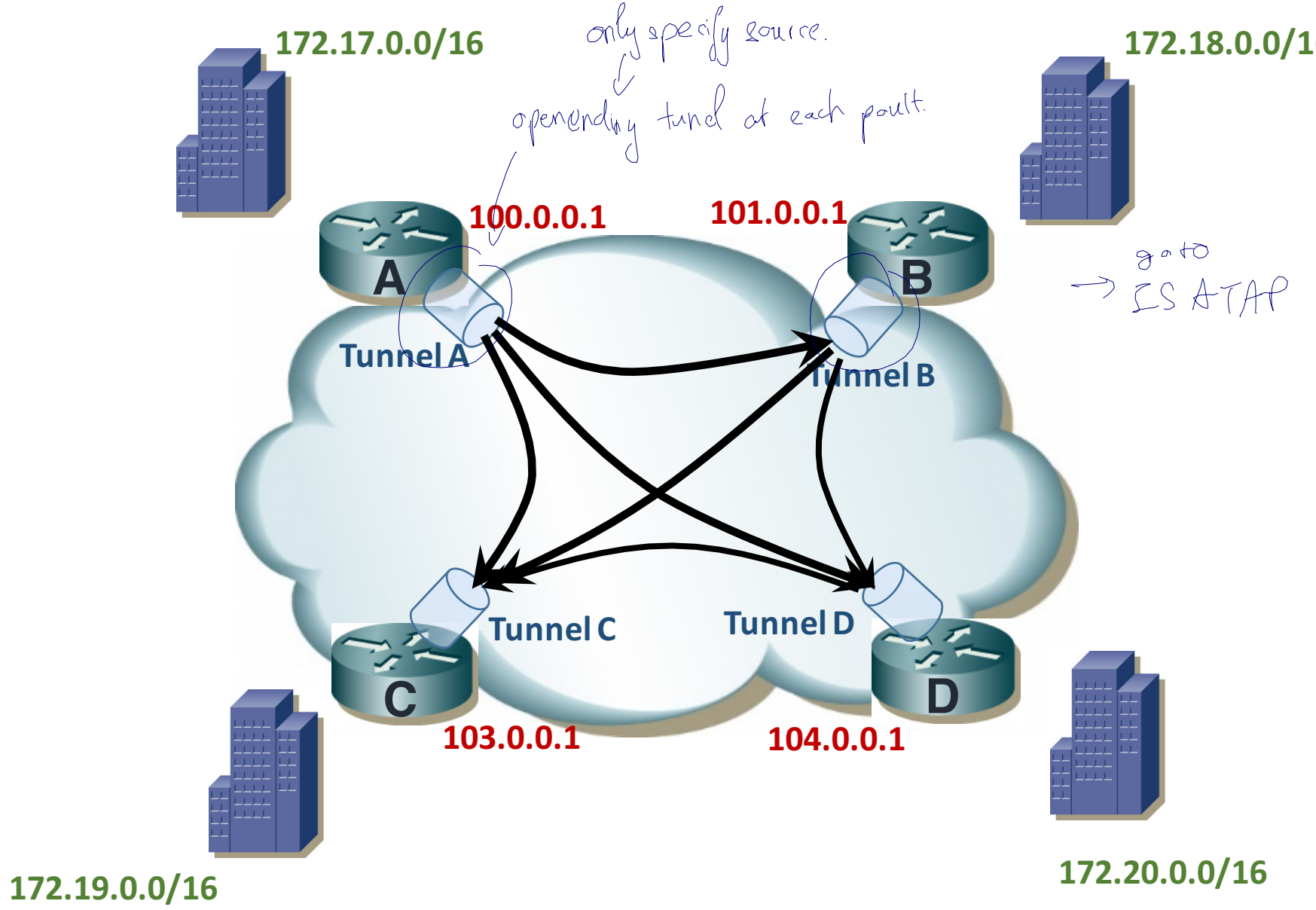



```
interface tunnelCA
  tunnel source 103.0.0.1
  tunnel destination 100.0.0.1
!
interface tunnelCB
  tunnel source 103.0.0.1
  tunnel destination 101.0.0.1
!
interface tunnelCD
  tunnel source 103.0.0.1
  tunnel destination 104.0.0.1
```

```
interface tunnelDA
  tunnel source 104.0.0.1
  tunnel destination 100.0.0.1
!
interface tunnelDC
  tunnel source 104.0.0.1
  tunnel destination 103.0.0.1
!
interface tunnelDB
  tunnel source 104.0.0.1
  tunnel destination 101.0.0.1
```


172.17.0.0/16
172.18.0.0/16
100.0.0.1
101.0.0.1
A
B
Tunnel AB
Tunnel AD
Tunnel BC
Tunnel AC
Tunnel BD
Tunnel CD
C
D
103.0.0.1
104.0.0.1
172.19.0.0/16
172.20.0.0/16


172.17.0.0/16
172.18.0.0/16
interface tunnelB
tunnel source 101.0.0.1
100.0.0.1
101.0.0.1
A
B
interface tunnelA
tunnel source 100.0.0.1
!
Tunnel A
Tunnel B
interface tunnelC
tunnel source 103.0.0.1
Tunnel C
Tunnel D
C
D
103.0.0.1
104.0.0.1
172.19.0.0/16
interface tunnelD
tunnel source 104.0.0.1
172.20.0.0/16


172.17.0.0/16
only specify source.
opening tunnel at each point.
172.18.0.0/16
100.0.0.1
101.0.0.1
A
B
go to
ISATAP
Tunnel A
Tunnel B
Tunnel C
Tunnel D
C
D
103.0.0.1
104.0.0.1
172.19.0.0/16
172.20.0.0/16

ISATAP

- **ISATAP** (Intra-Site Automatic Tunneling Addressing Protocol) an automatic tunneling mechanism used inside an organization that has an IPv4-dominant backbone, but has selected users that need IPv6 capability

ISATAP Functions

- ISATAP connects dual-stack nodes, isolated within an IPv4-only network
  - To exchange IPv6 traffic with each other (host ISATAP)
  - To exchange traffic with the global IPv6 Internet
- ISATAP is a mechanism with minimal configuration required
- ISATAP is ideal when there are relatively few, relatively scattered individual nodes that need service

Link-Local ISATAP

192.0.2.100 → FE80::0000:5EFE:C000:0264
network identifier | ISATAP 32bit link local suffix | IP in hex

192.0.2.100

IPv4 Address

convert

Is converted to hex form

C000:0264

0000:5EFE

And pre-pended with the ISATAP 32-bit link-local suffix

lower 64 bit

::0000:5EFE:C000:0264

FE80::/10

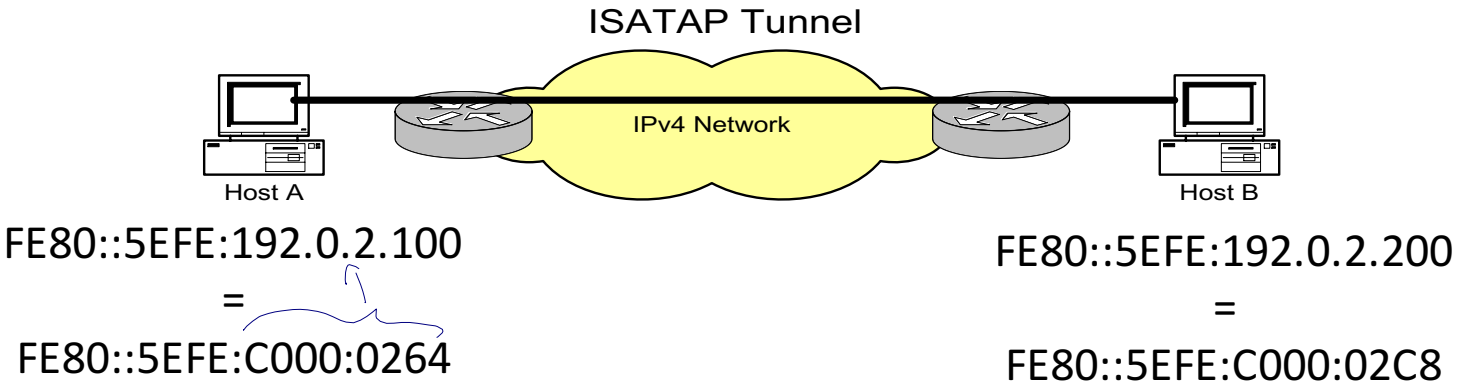
The link-local prefix merges with the network identifier to create the

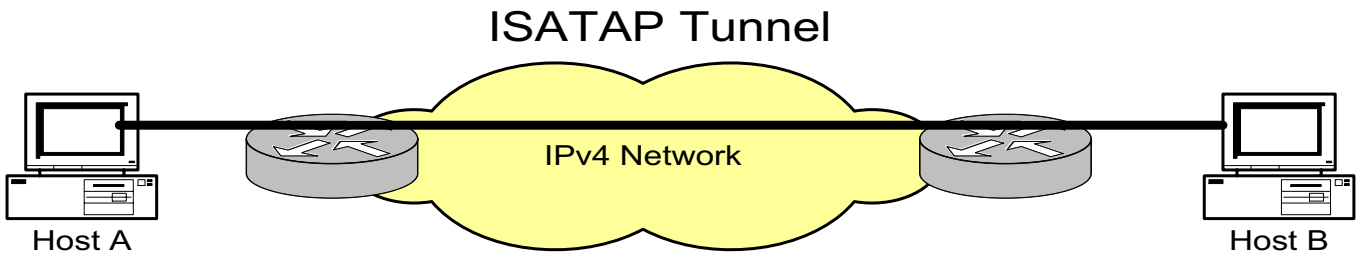
ISATAP IPv6 link-local address

FE80::0000:5EFE:C000:0264

Link-local ISATAP example

- Two ISATAP hosts exchanging packets using link-local addresses
- Only route on ISATAP hosts is “send all IPv6 traffic via ISATAP pseudo-IF”
- Hosts are many IPv4 hops away which appear link-local to IPv6





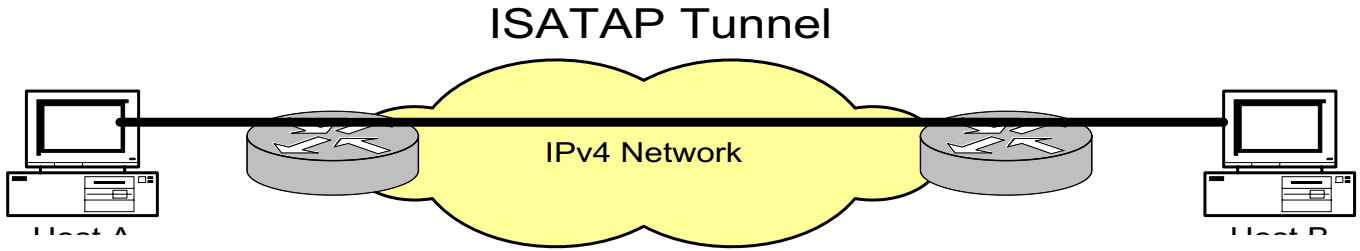
FE80::5EFE:192.0.2.100
=
FE80::5EFE:C000:0264

FE80::5EFE:192.0.2.200
=
FE80::5EFE:C000:02C8

src: 192.0.1.100
dst: 192.0.2.200

src: FE80::5EFE:C000:0264
dst: FE80::5EFE:C000:02C8

FE80::5EFE:192.0.2.10
=
FE80::5EFE:C000:020A

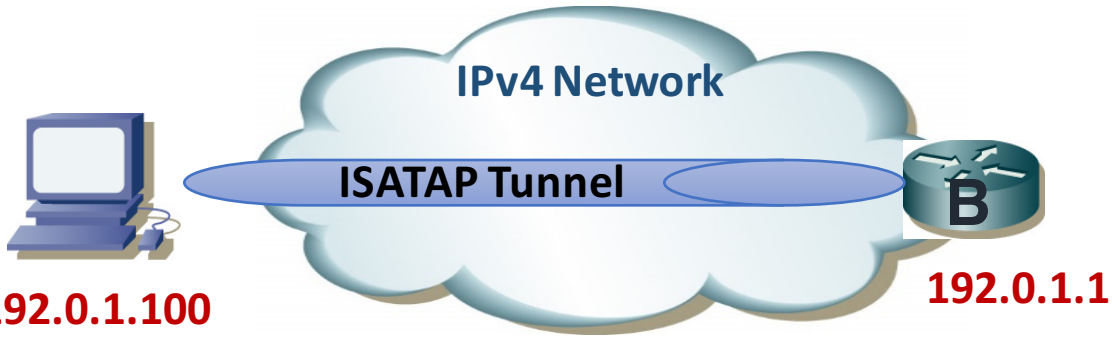


FE80::5EFE:192.0.2.100
=
FE80::5EFE:C000:0264

FE80::5EFE:192.0.2.200
=
FE80::5EFE:C000:02C8

src: 192.0.1.100
dst: 192.0.2.10

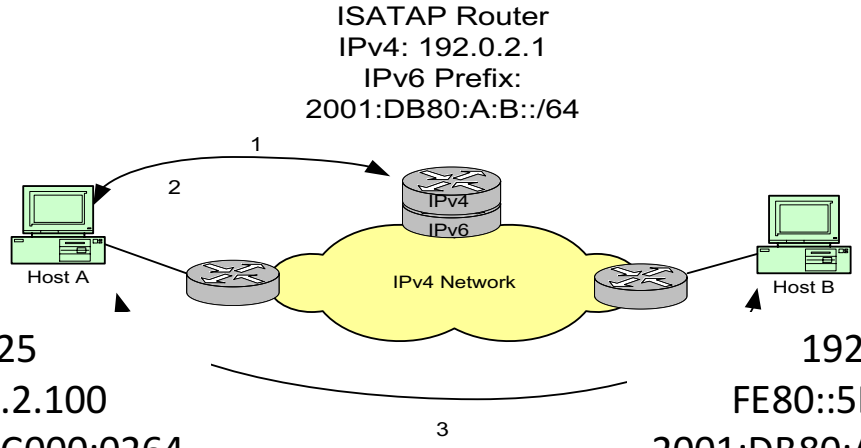
src: FE80::5EFE:C000:0264
dst: FE80::5EFE:C000:020A



Src: FE80::5EFE:C200:0164
Dst: FE80::5EFE:C200:0101

Globally-routable ISATAP

- ISATAP more flexible when using an ISATAP router
- ISATAP hosts are configured with ISATAP router IPv4 address
- Hosts sends router solicitation, inside tunnel, and ISATAP router responds



192.0.2.100/25
FE80::5EFE:192.0.2.100
2001:DB80:A:B::5EFE:C000:0264
ISATAP Router: 192.0.2.1

192.0.2.200/25
FE80::5EFE:192.0.2.200
2001:DB80:A:B::5EFE:C000:02C8
ISATAP router: 192.0.2.1

ISATAP Summary

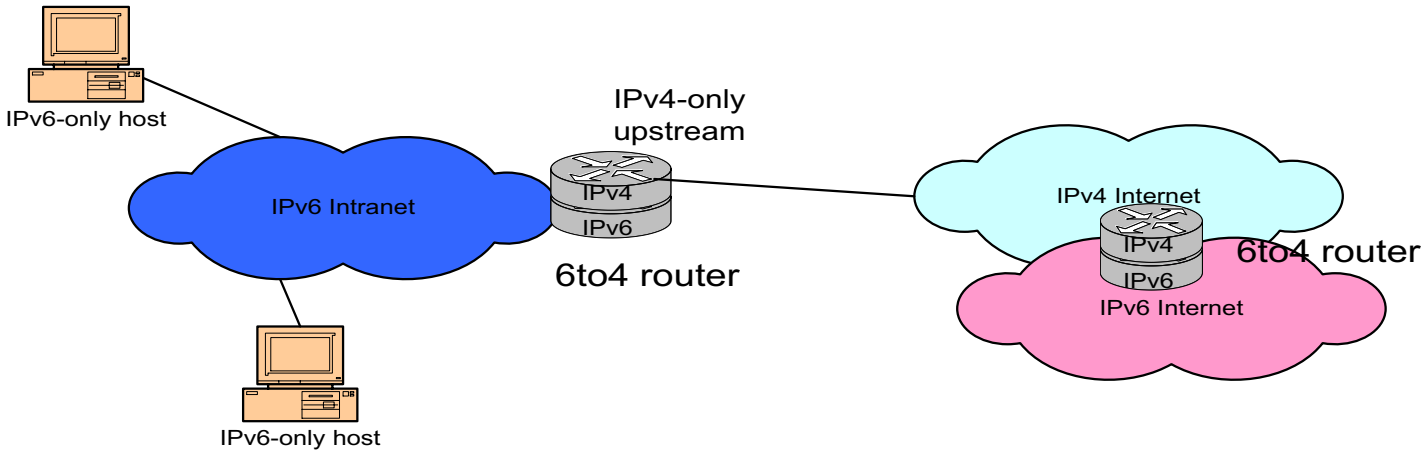
- ISATAP scales better than manually configured tunnels inside the enterprise
- Decapsulate-from-anywhere issues (like 6to4) mitigated by internal deployment
- No authentication provided – any dual stack node that knows ISATAP router address can obtain services
- May need to look at other alternatives if security is required

IPv6 6to4 Transition Mechanism

- 6to4 is an automatic tunneling mechanism that provides v6 capability to a dual-stack node or v6-capable site that has only IPv4 connectivity to the site

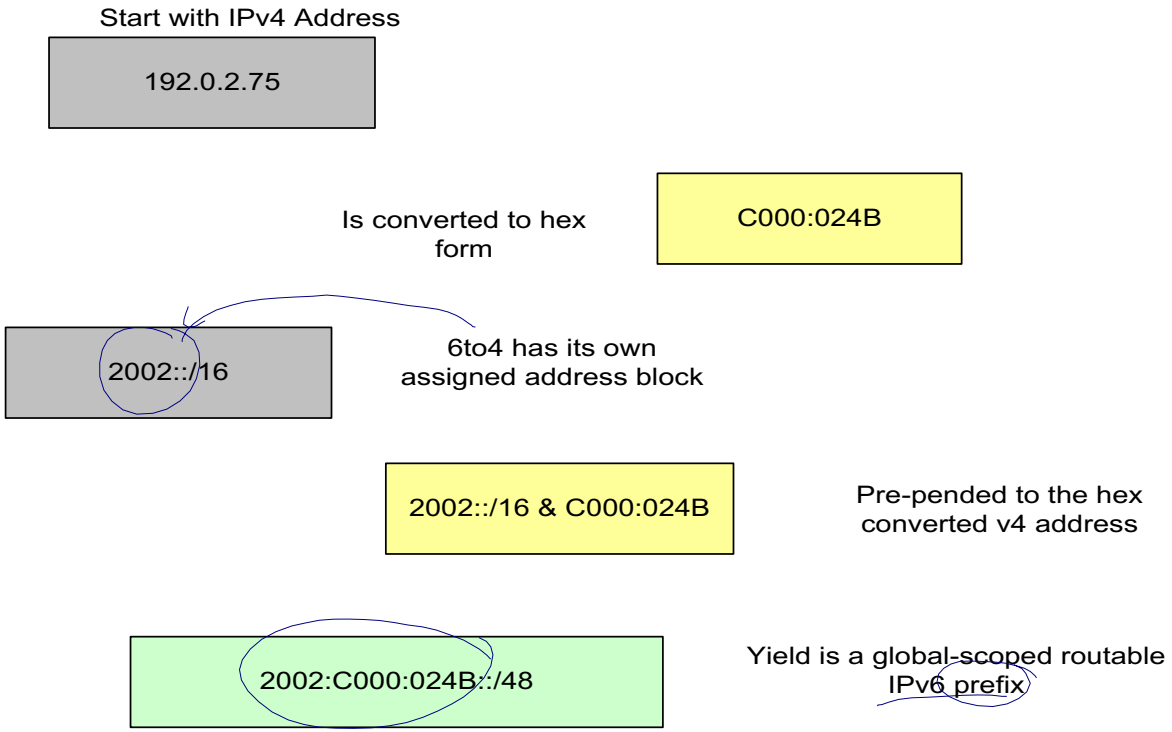
6to4 Basics

- 6to4 is an automatic tunnel mechanism
- Provides v6 upstream for v6-capable site over v4-only Internet connection
- Uses embedded addressing (v4addr embedded in v6addr) as do other automatic mechanisms



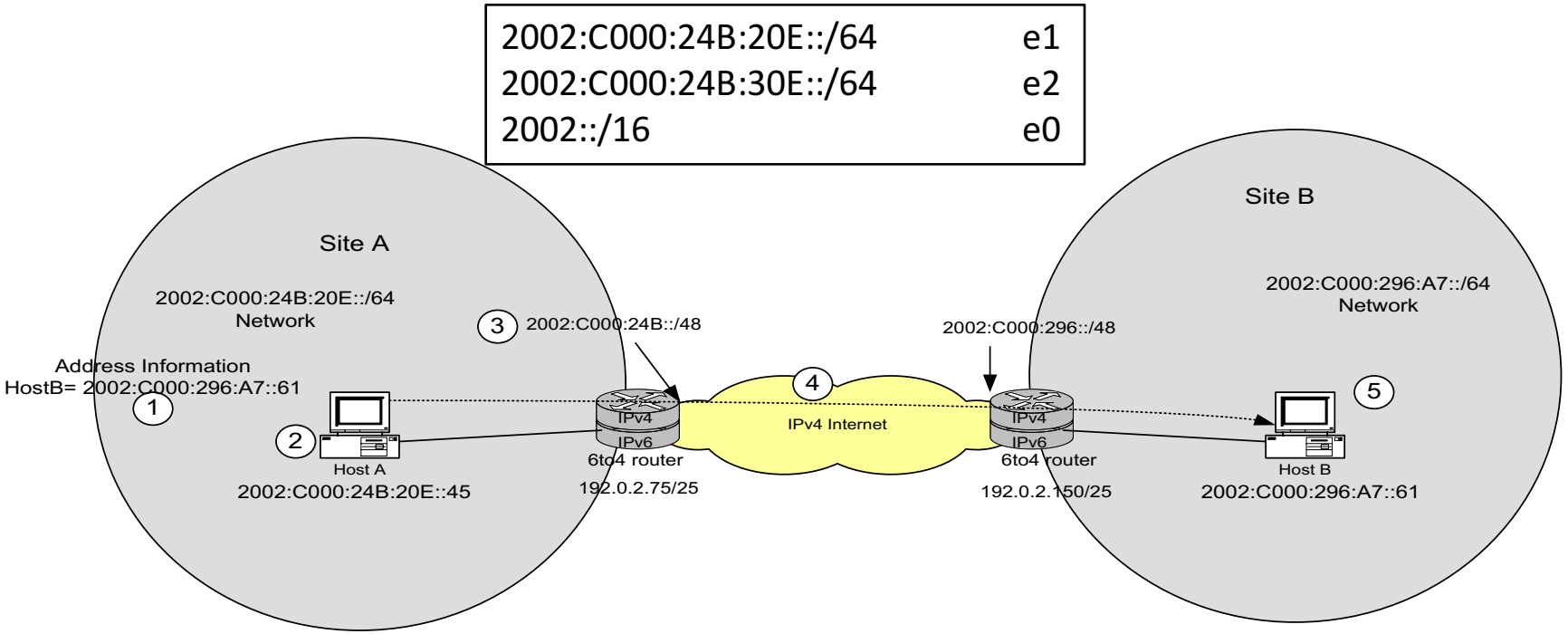
6to4 Address Construction

- 6to4 setups a valid, unique /48 IPv6 prefix from the outside IPv4 address of the site router

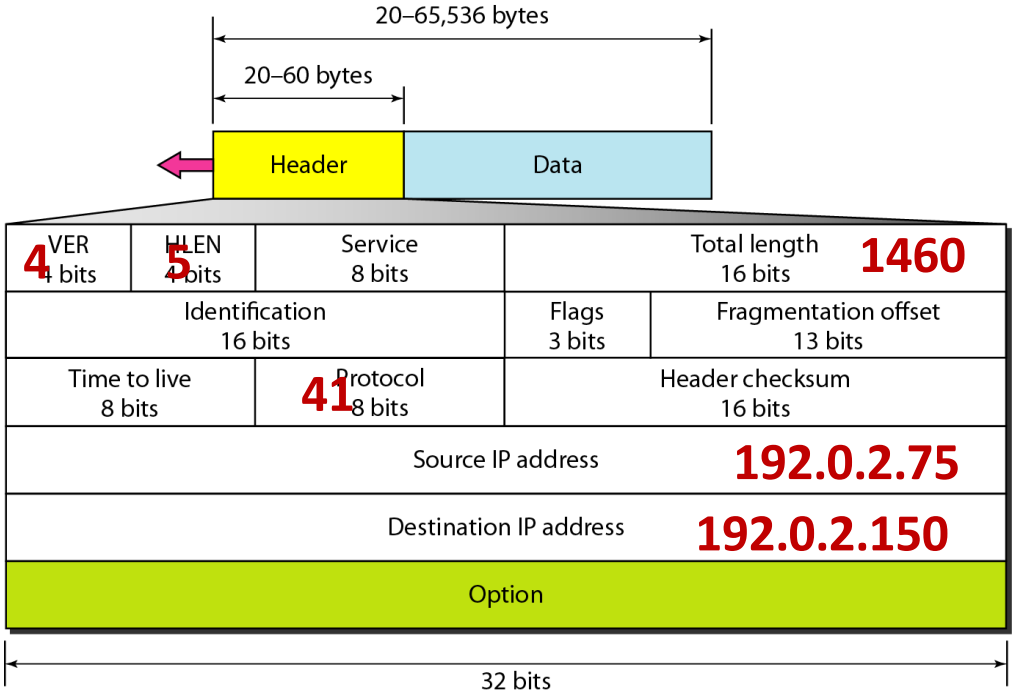


6to4 Site-to-Site Example

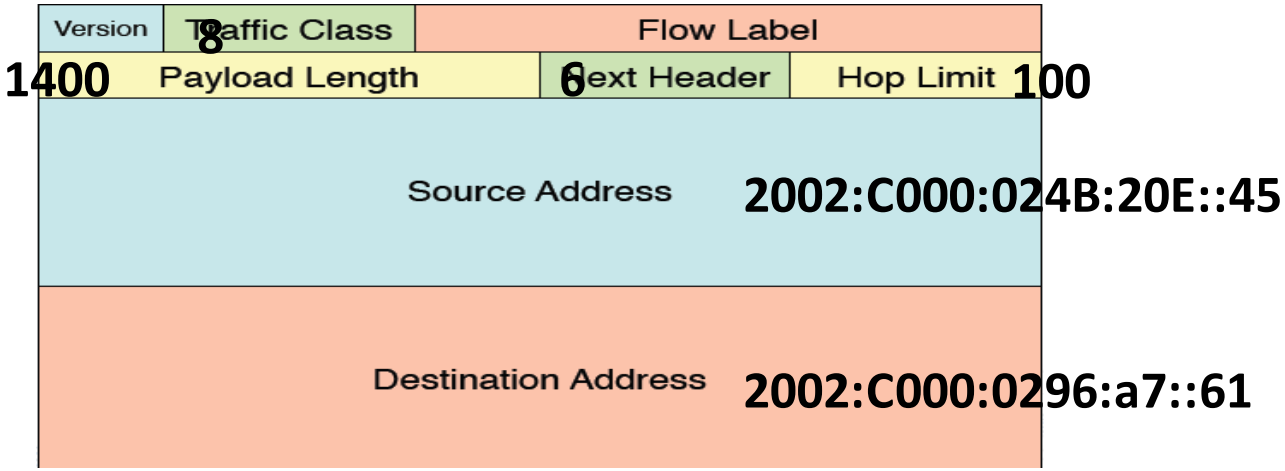
- 6to4 edge devices are called “6to4 site routers”
- IPv4-only between sites, full IPv6 within sites
- Host A packet tunneled through IPv4 network to destination 6to4 site

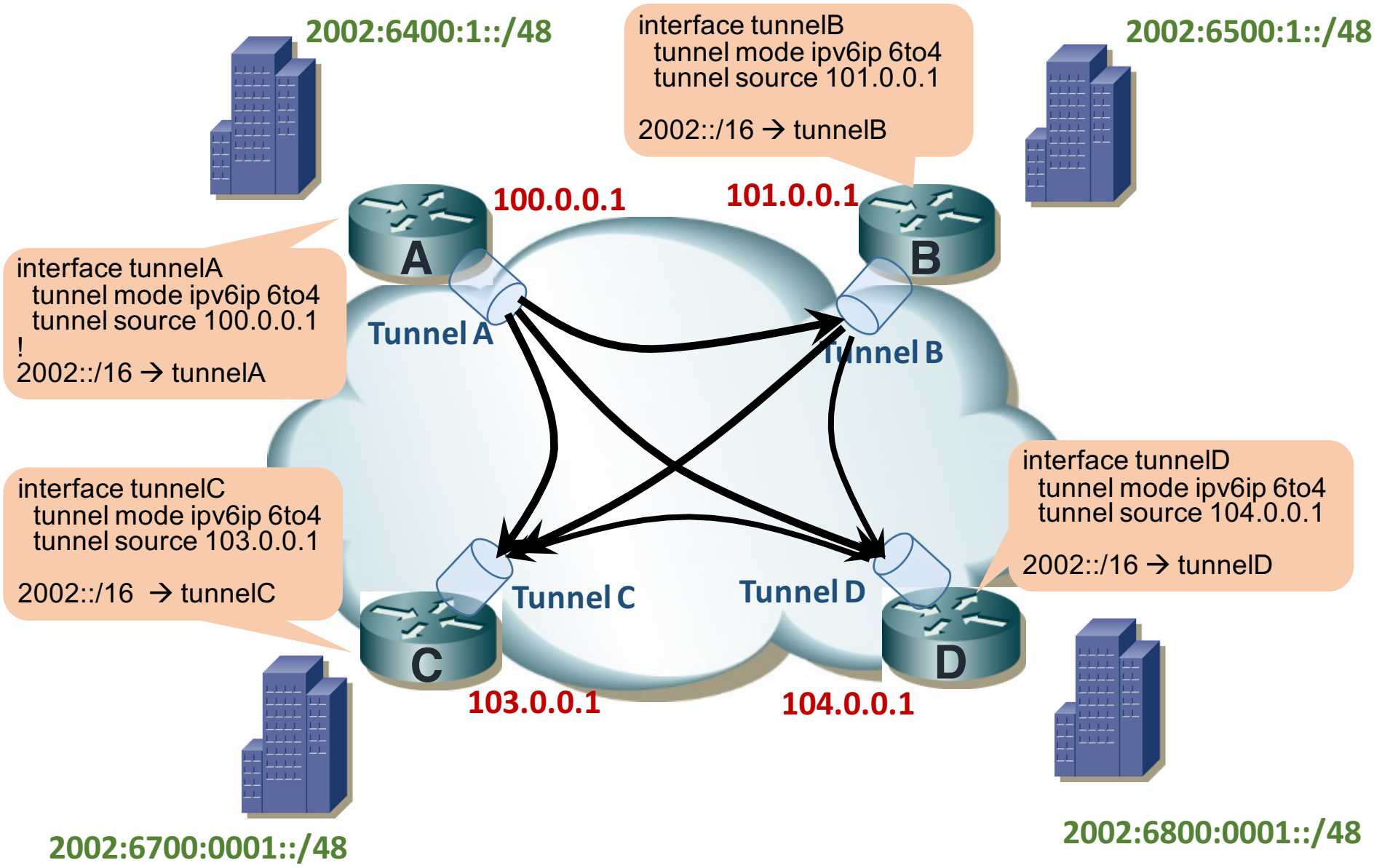


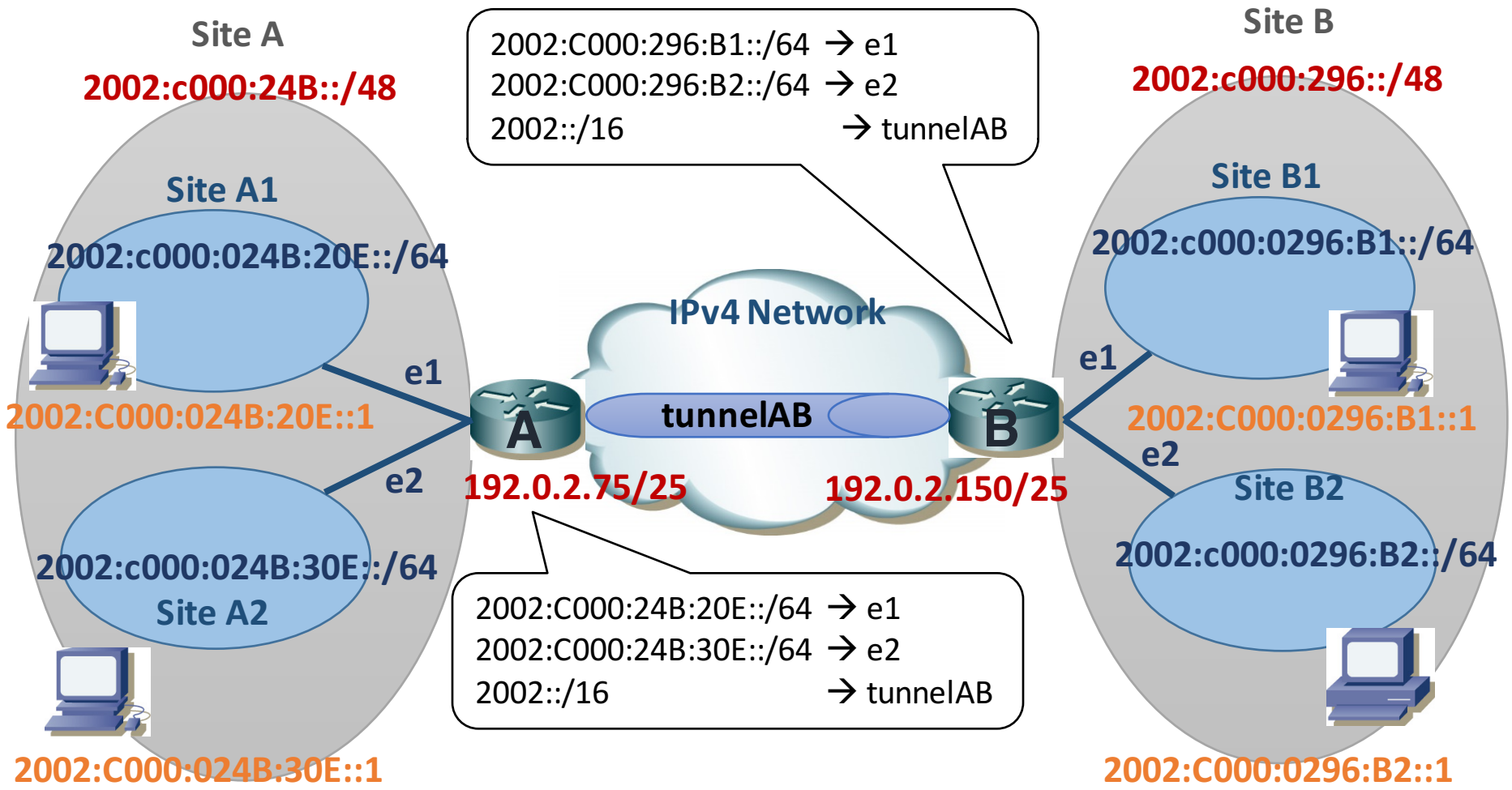
Encapsulation

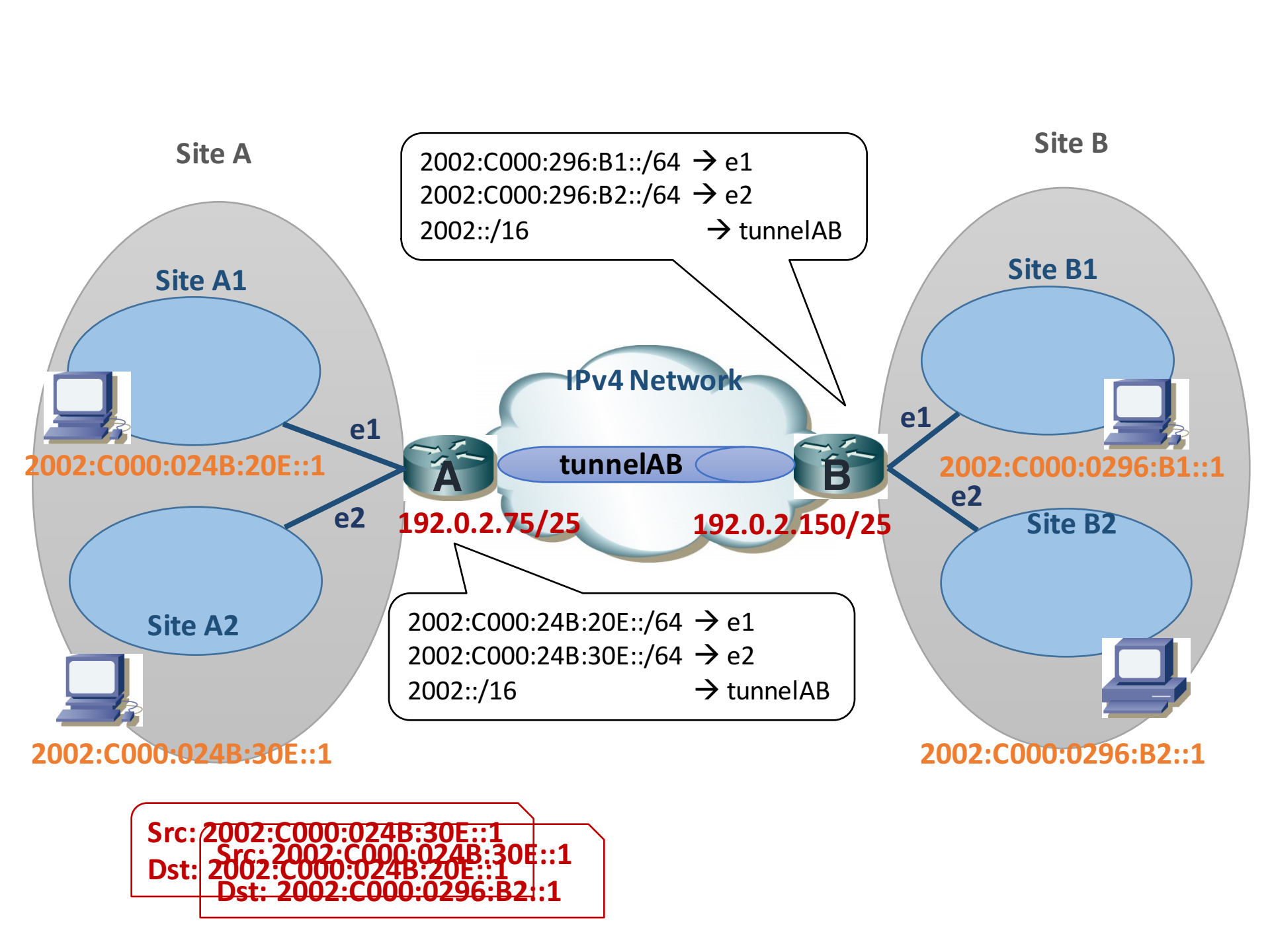


TTL from inner or configured
TOS from inner or configured




2002:6400:1::/48
interface tunnelB
tunnel mode ipv6ip 6to4
tunnel source 101.0.0.1
2002::/16 → tunnelB
2002:6500:1::/48
100.0.0.1
101.0.0.1
A
B
interface tunnelA
tunnel mode ipv6ip 6to4
tunnel source 100.0.0.1
!
2002::/16 → tunnelA
Tunnel A
Tunnel B
interface tunnelC
tunnel mode ipv6ip 6to4
tunnel source 103.0.0.1
2002::/16 → tunnelC
interface tunnelD
tunnel mode ipv6ip 6to4
tunnel source 104.0.0.1
2002::/16 → tunnelD
Tunnel C
Tunnel D
C
D
103.0.0.1
104.0.0.1
2002:6700:0001::/48
2002:6800:0001::/48

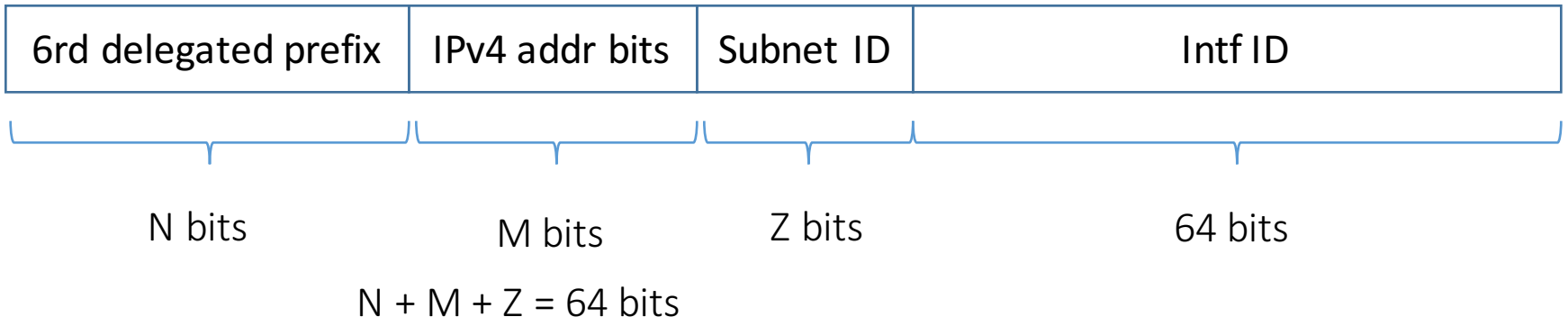

Site A
2002:c000:24B::/48
Site A1
2002:c000:024B:20E::/64
2002:C000:024B:20E::1
e1
e2
2002:c000:024B:30E::/64
Site A2
2002:C000:024B:30E::1
2002:C000:296:B1::/64 → e1
2002:C000:296:B2::/64 → e2
2002::/16 → tunnelAB
IPv4 Network
tunnelAB
A
B
192.0.2.75/25
192.0.2.150/25
2002:C000:24B:20E::/64 → e1
2002:C000:24B:30E::/64 → e2
2002::/16 → tunnelAB
Site B
2002:c000:296::/48
Site B1
2002:c000:0296:B1::/64
e1
2002:C000:0296:B1::1
e2
Site B2
2002:c000:0296:B2::/64
2002:C000:0296:B2::1


Site A
Site B
2002:C000:296:B1::/64 → e1
2002:C000:296:B2::/64 → e2
2002::/16 → tunnelAB
Site A1
Site B1
IPv4 Network
e1
tunnelAB
e1
2002:C000:024B:20E::1
A
B
2002:C000:0296:B1::1
e2
e2
192.0.2.75/25
192.0.2.150/25
Site B2
Site A2
2002:C000:24B:20E::/64 → e1
2002:C000:24B:30E::/64 → e2
2002::/16 → tunnelAB
2002:C000:024B:30E::1
2002:C000:0296:B2::1
Src: 2002:C000:024B:30E::1
Dst: 2002:C000:024B:20E::1
Src: 2002:C000:024B:30E::1
Dst: 2002:C000:0296:B2::1

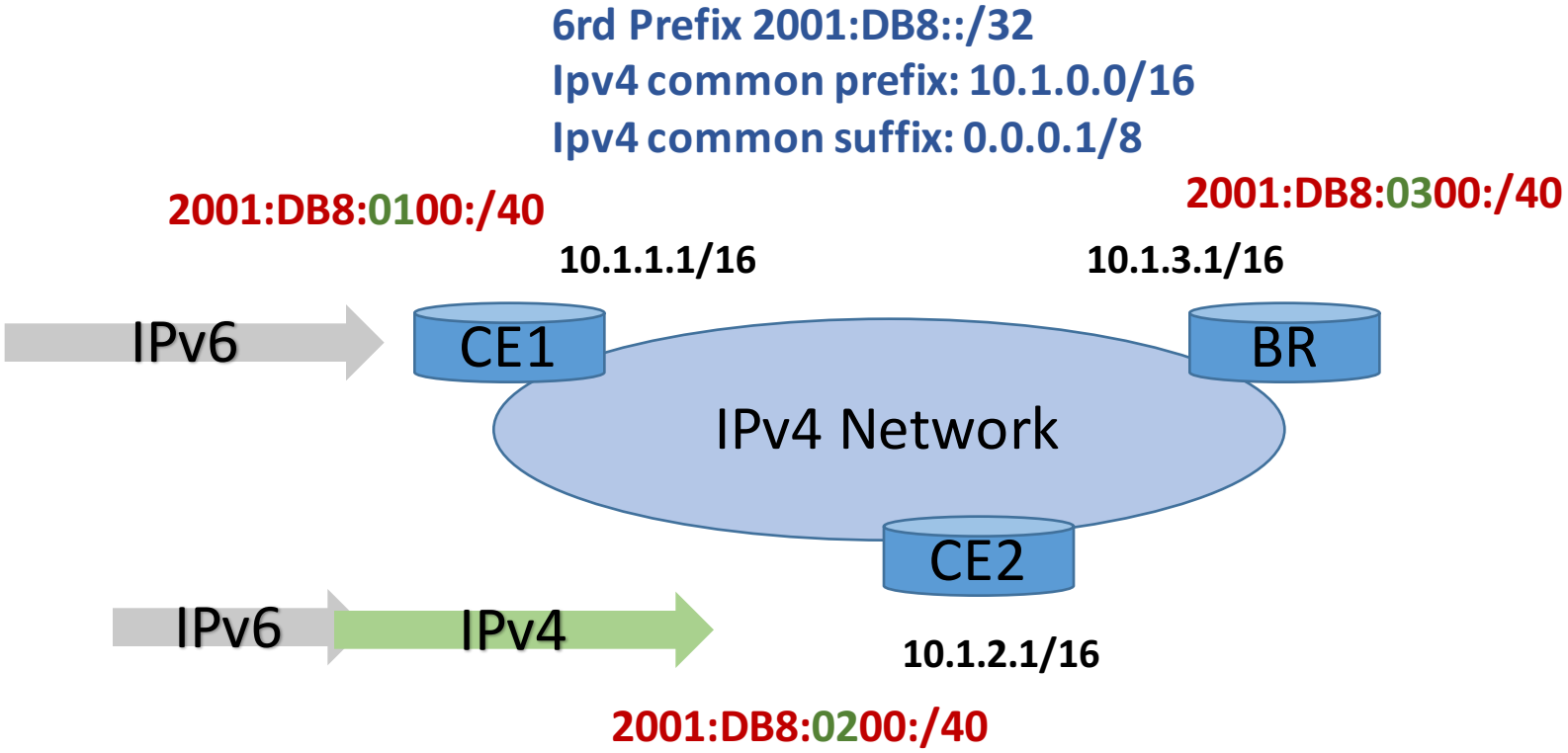
6RD Tunnel

- IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – RFC 5969
- Utilize an SP's own IPv6 address prefix rather than a well-known prefix (2002::/16)

6RD Address Format

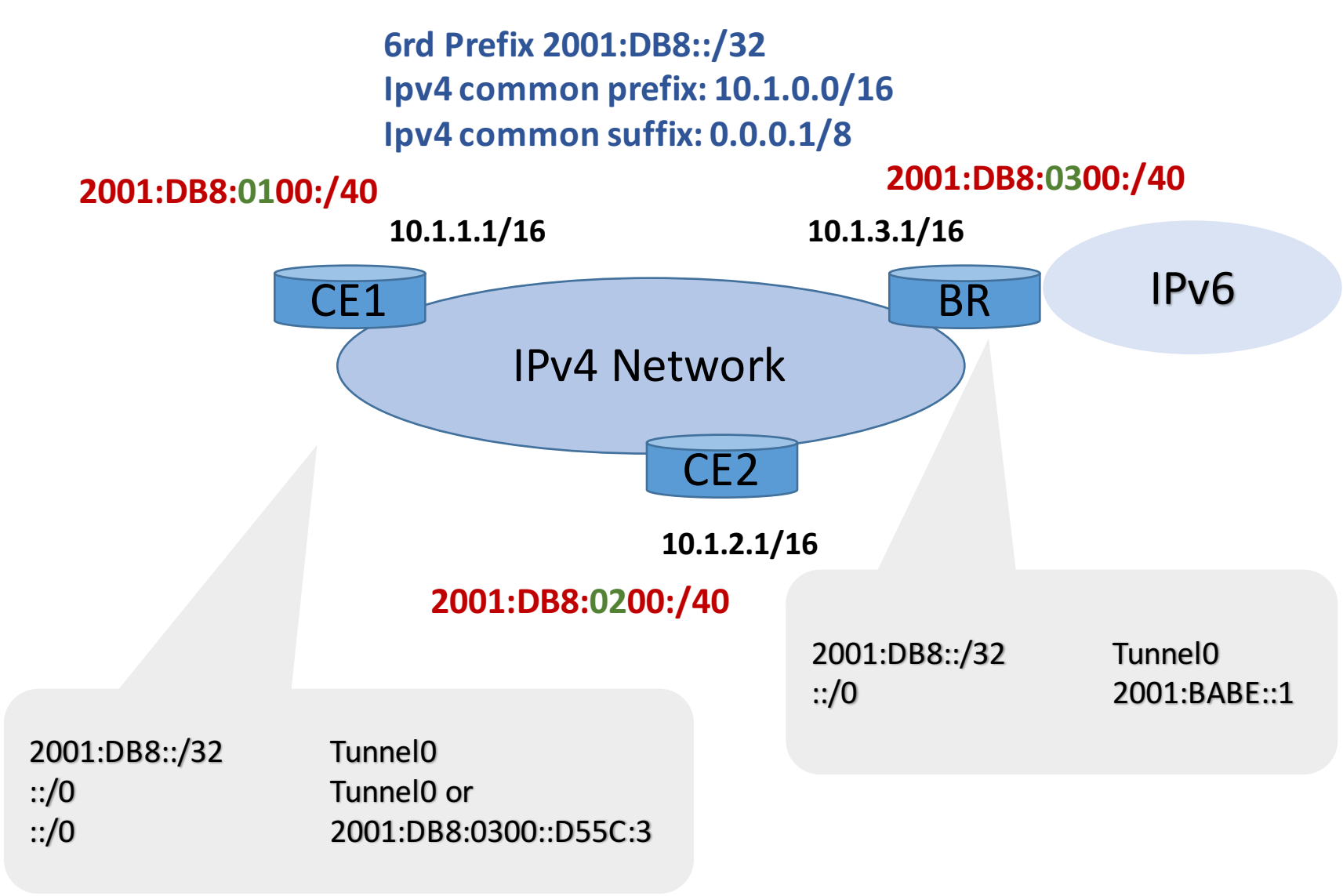


Parameter	Value
6rd Prefix/length	2001:DB8::/32
IPv4 Common prefix/length	10.1.0.0/16
IPv4 Common suffix/length	0.0.0.1/8



IPv6: 2001:DB8:0100::C15C:0 → 2001:DB8:0200::C26B:0

IPv4: 10.1.1.1 → 10.2.1.1


6rd Prefix 2001:DB8::/32
Ipv4 common prefix: 10.1.0.0/16
Ipv4 common suffix: 0.0.0.1/8
2001:DB8:0100:/40
2001:DB8:0300:/40
10.1.1.1/16
10.1.3.1/16
CE1
BR
IPv6
IPv4 Network
CE2
10.1.2.1/16
2001:DB8:0200:/40
2001:DB8::/32 Tunnel0
::/0 2001:BABE::1
2001:DB8::/32 Tunnel0
::/0 Tunnel0 or
::/0 2001:DB8:0300::D55C:3

IPv6 Tunnel Address Format