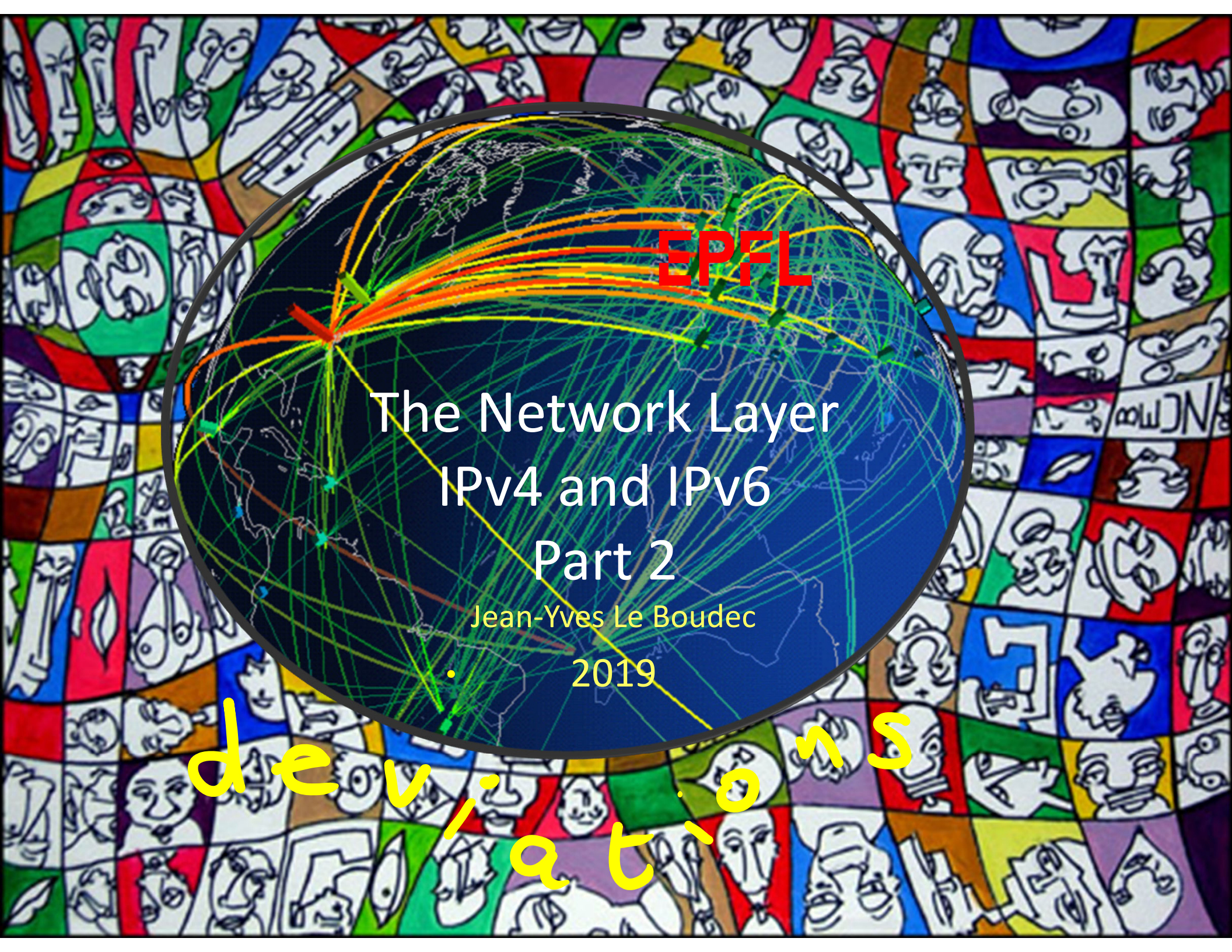
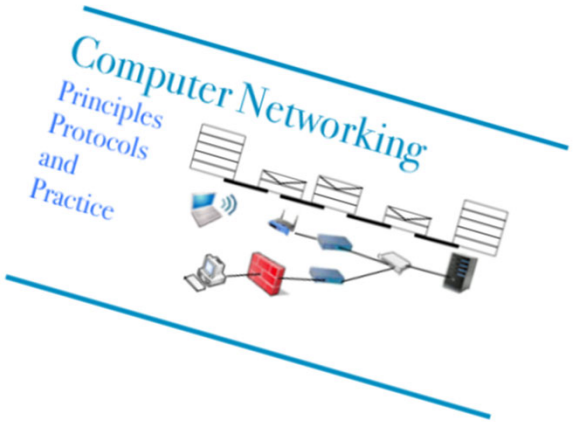

EPFL
The Network Layer
IPv4 and IPv6
Part 2
Jean-Yves Le Boudec
2019
deviations

Contents

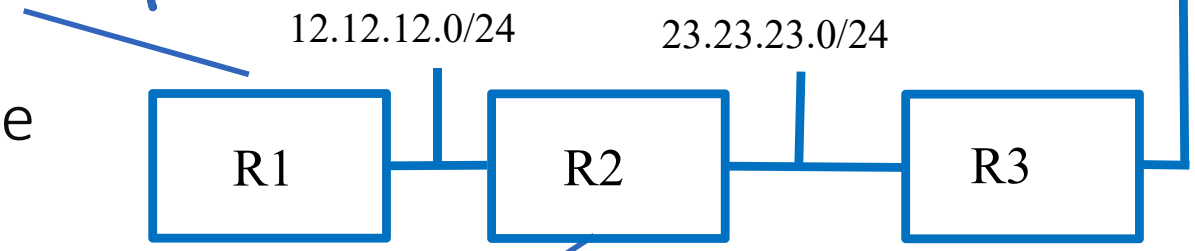
Textbook

Chapter 5: The Network Layer



9. Proxy-ARP

dest	next-hop	interface
12.12.12.0/24	On-link	eth0
12.12.12.4/32	12.12.12.2	eth0
0/0	12.12.12.2	eth0



Reminder: IP principle says one subnet = one LAN

dest	next-hop	interface
12.12.12.0/24	On-link	eth0
23.23.23.0/24	On-link	eth1
12.12.12.4/32	23.23.23.2	eth1

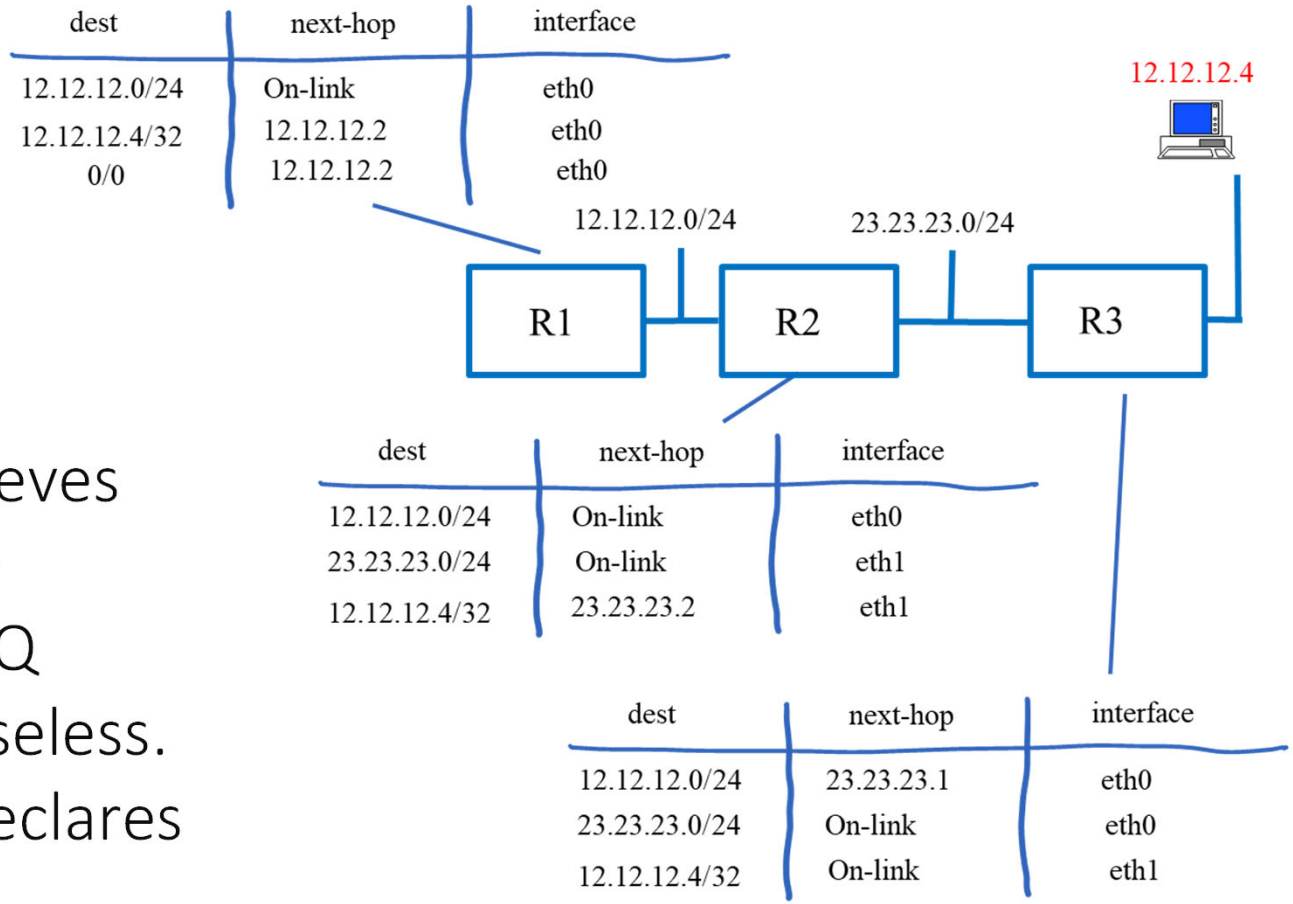
Assume you want to cheat with this principle, e.g. 12.12.12.4 is in the wrong place, but you want to keep it that way. You can support this configuration with /32 entries (with IPv4) in forwarding tables at R1, R2 and R3 (“Host Routes”)

dest	next-hop	interface
12.12.12.0/24	23.23.23.1	eth0
23.23.23.0/24	On-link	eth0
12.12.12.4/32	On-link	eth1

Does this solve the problem ?

For packets handled by R1, R2 and R3, yes.

Not for A:
when A has a packet to send to 12.12.12.4, A believes 12.12.12.4 is on the same LAN and sends an ARP REQ for 12.12.12.4, which is useless.
A receives no reply and declares 12.12.12.4 unreachable.



One solution is to write a complete routing table in A, but this is not usually done as A is not a router and does participate in IGP

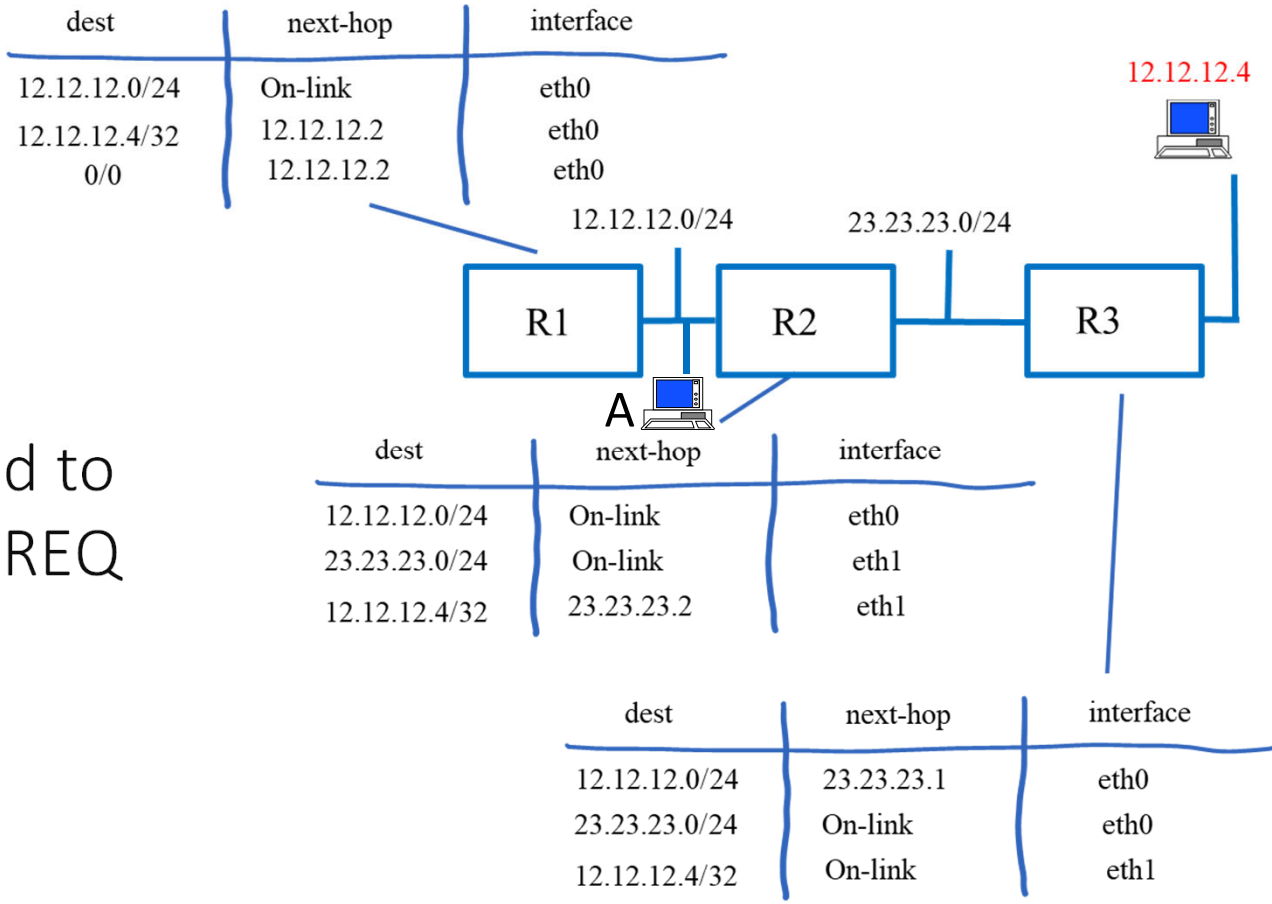
Another solution is Proxy ARP/Proxy NDP

Proxy ARP / ND Proxy

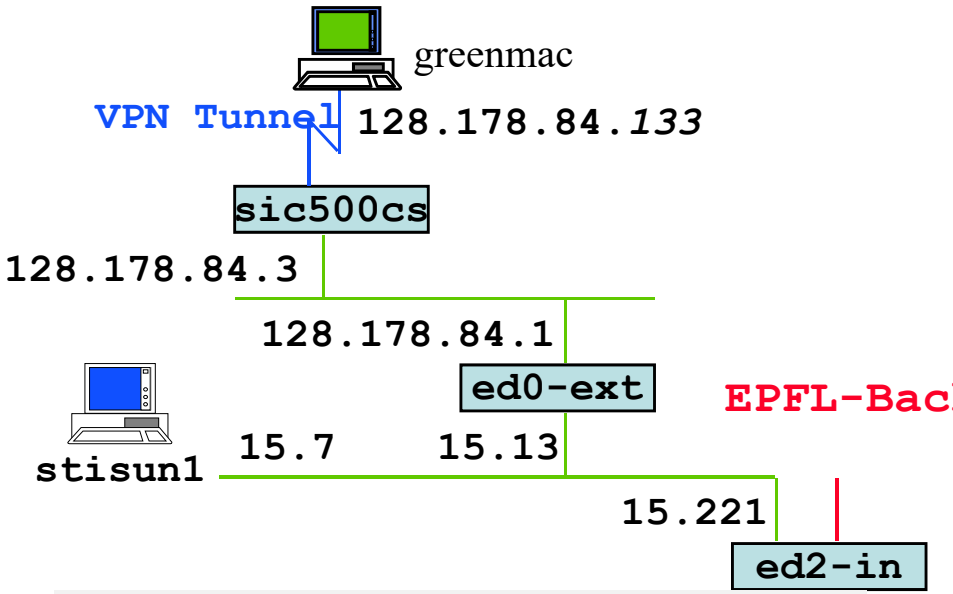
R2 performs Proxy ARP in lieu of 12.12.12.4: i.e. R2 answers all ARP as if it were 12.12.12.4.

When A has a packet to send to 12.12.12.4, it sends an ARP REQ on its LAN, with target IP address = 12.12.12.4. R2 responds with R2's MAC address. Packet sent by A is received by R2 and forwarded to R3.

R2 is a *proxy* for 12.12.12.4 on the LAN of A.



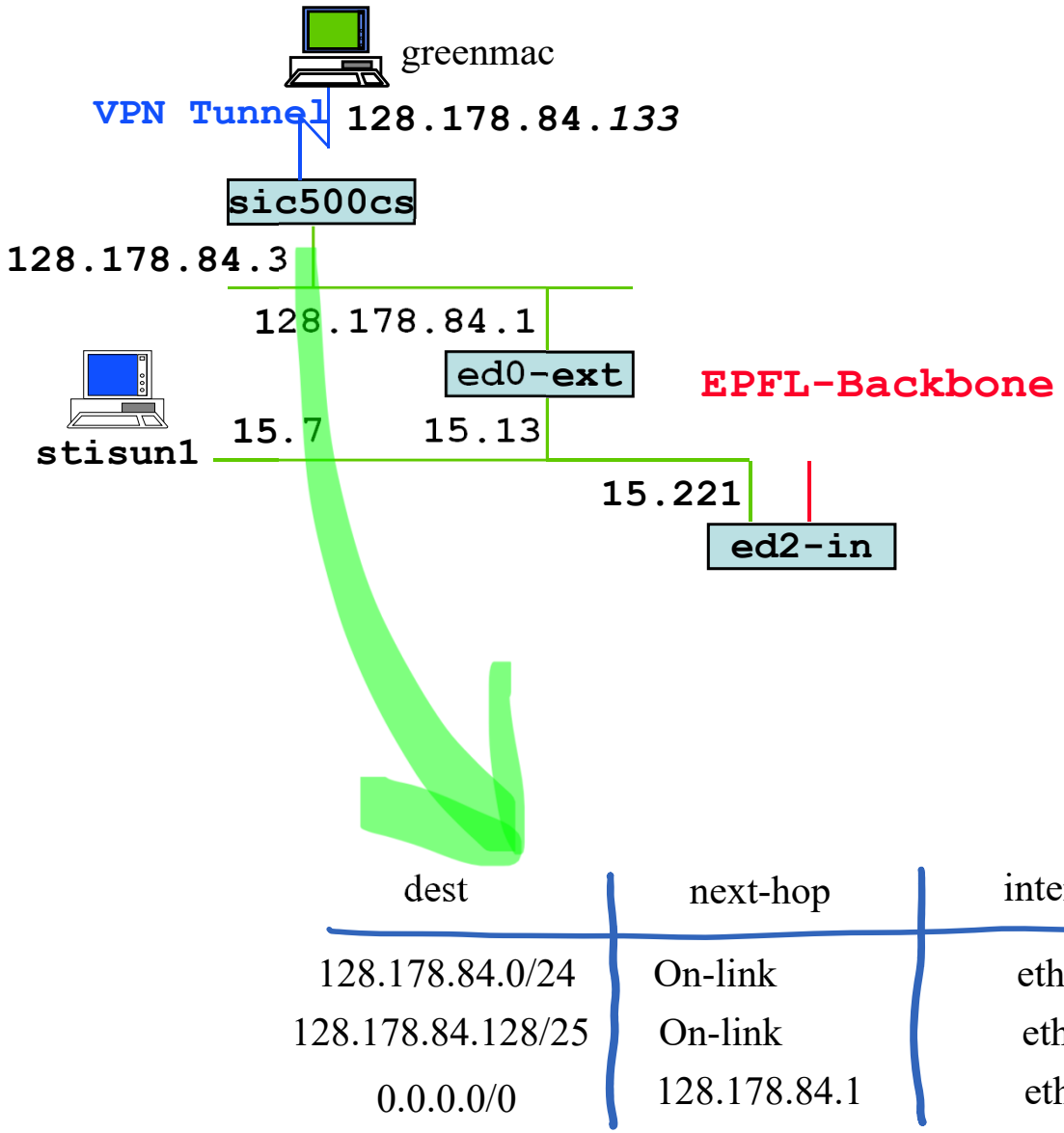
sic500cs is a router (not a bridge); ed0-ext has a packet to 128.178.84.133 and sends an ARP REQ



1. All subnets have 255.255.255.0 netmask
2. Subnet 84 is on both sides of sic500cs
3. sic500cs does PROXY-ARP on behalf of greenmac

A. sic500cs replies with own MAC address
B. sic500cs replies with the MAC address of greenmac
C. Greenmac replies with his MAC address
D. Greenmac replies with the MAC address of sic500cs
E. I don't know

Solution



Answer B

sic500cs responds with own MAC adddress – ed0-ext now believes that greenmac has the same MAC address as sic500cs

All traffic to greenmac is sent to sic500cs

sic500cs needs appropriate entry in routing table

10. Fragmentation

- Link-layer networks have different maximum frame lengths
- MTU (maximum transmission unit) = maximum frame size usable for an IP packet (including the IP header)
- MAC layer options and tunnels reduce MTU

Link-layer Network	MTU
Ethernet, WiFi	1500
VPN Tunnel in IPv4	1400
Token Ring 4 Mb/s	4464
16 Mb/s	17914
FDDI	4352
X.25	576
Frame Relay	1600
ATM with AAL5	9180
Hyperchannel	65535
PPP	296 to 1500
Maximum MTU for IPv6	4GB

```
Z:\>netsh interface ipv6 show subinterfaces

   MTU  MediaSenseState   Bytes In  Bytes Out  Interface
------  ---------------  ---------  ---------  -------------
4294967295                1          0        28980  Loopback Pseudo-Interface 1
1500                      5          0            0  Wireless Network Connection 2
1400                      1      19419        14328  VPN EPFL
1500                      1  126682357     17501036  Local Area Connection
1280                      5          0          536  Local Area Connection* 12
```

IPv4 Fragmentation

hosts or routers may have IP datagrams larger than MTU

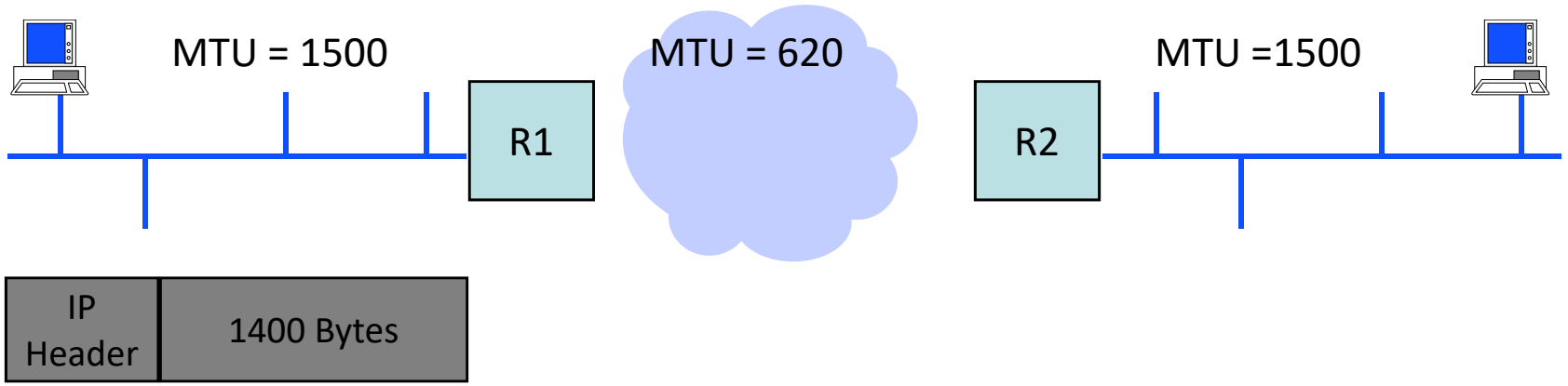
Fragmentation is performed when IP datagram too large

re-assembly is only at destination, never at intermediate points

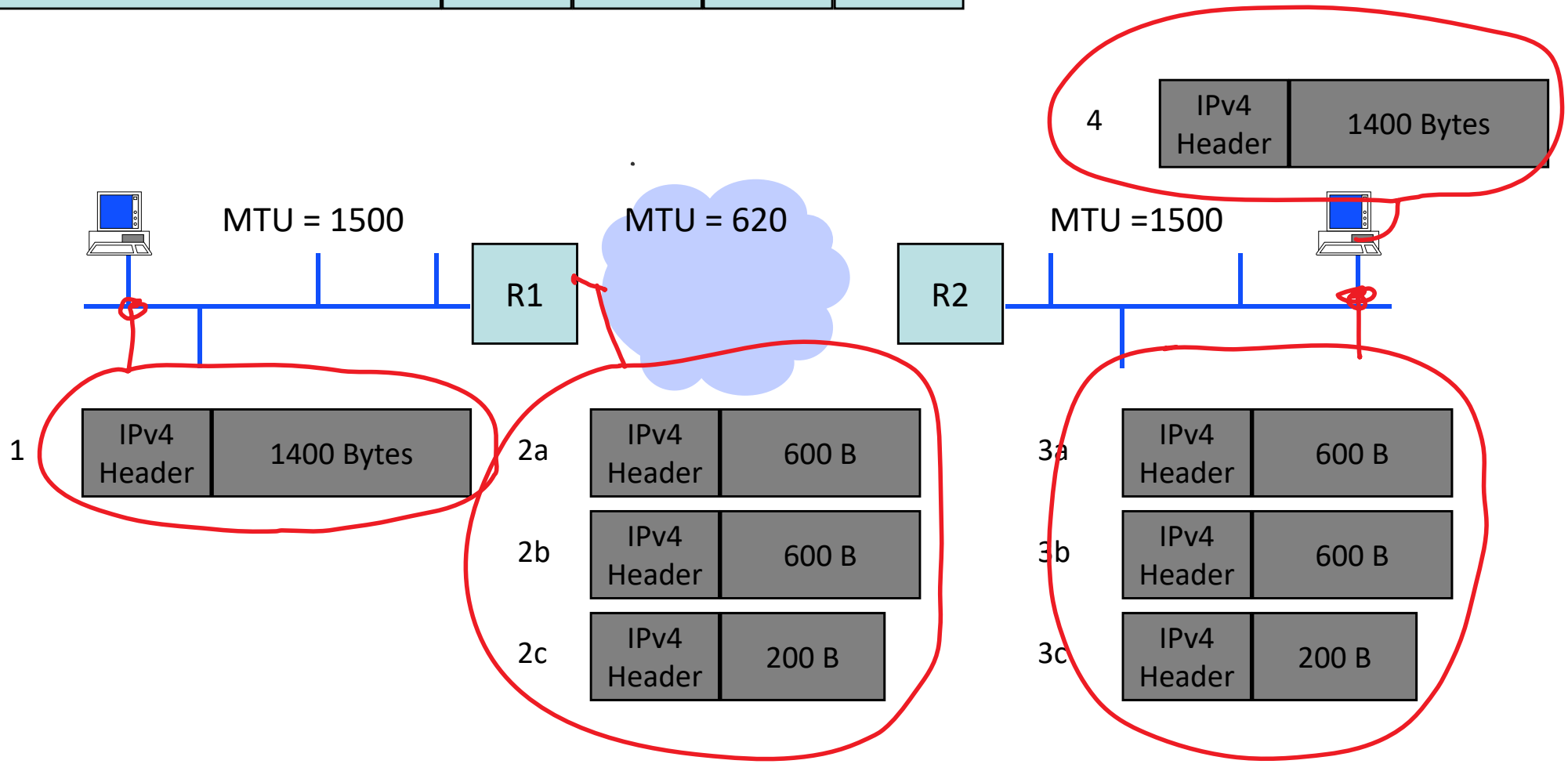
fragmentation is in principle avoided with TCP

all fragments are self-contained IP packets

IP *datagram* = original ; IP *packet* = fragments or complete datagram



IPv4 header fields	1 and 4	2a, 3a	2b,3b	2c, 3c
Length	1420	620	620	220
Identification	567	567	567	567
More Fragment flag	0	1	1	0
Offset	0	0	75	150
8 * Offset	0	0	600	1200



Fragment data size (here 600) is always a multiple of 8
Identification given by source

IPv6 Fragmentation

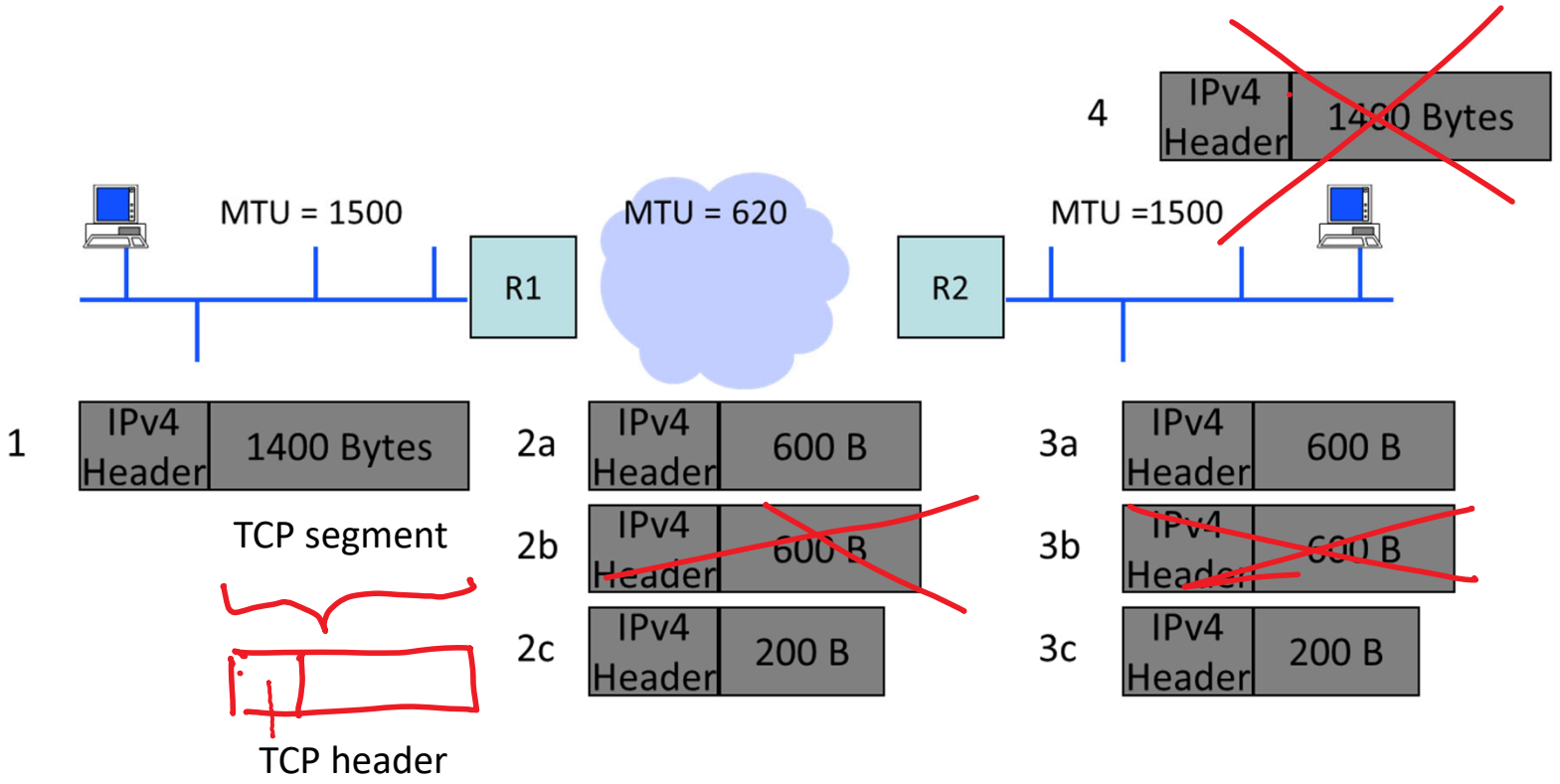
Same as IPv4 except

- IPv6 Routers never fragment – drop packet if too large
- Fragmentation header is a «next header»
- Minimum MTU is 1280 Bytes (vs 68 for IPv4)

One TCP segment is contained in one IPv4 datagram that is fragmented by a router on its way from source to destination. One of the fragments is lost. What will TCP re-transmit ?

A. The bytes that were in the missing fragment
B. The bytes that were in all fragments of the datagram, missing or not
C. It depends whether the loss is detected by fast retransmit or by timeout
D. I don’t know

Solution



If the TCP segment is in several fragments and one of the fragments is lost, the whole segment is lost and will be re-transmitted

fragmentation/re-assembly is done at IP layer

this is why fragmentation of IP packets that contain TCP should be avoided

TCP, UDP and Fragmentation

The UDP service interface accepts a datagram

- up to 64 KB (by default) or up to 4GB (with IPv6 "jumbo payload" extension header)
- UDP datagram passed to the IP service interface as one SDU
- is fragmented at the source if resulting IP datagram is too large to fit on Path MTU

The TCP service interface is stream oriented

- packetization is done by TCP – fragmentation should be avoided using "*Path MTU*" (= minimum MTU along one path) estimation
- Packetization Layer Path MTU Discovery (PLPMTUD): TCP makes Path MTU probes from time to time; estimate the largest possible Path-MTU that works by observing packets that were acknowledge
- TCP connections negotiate an MSS (maximum segment size)
  Host operating system should verify that
  $MSS \leq PathMTU - IP\ header\ size - TCP\ header\ size$

When a host generates UDP traffic, the port number is always present in all packets

A. True
B. False
C. True with IPv4, false with IPv6
D. True with IPv6, false with IPv4
E. I don’t know

Solution

False

If fragmentation occurs, only the first fragment contains the port number.

Fragmentation may occur at the source with UDP and with IPv6 as well as IPv4

11. Tunnels

Definition:
a *tunnel*, also called *encapsulation* occurs whenever a communication layer carries packets of a layer that is not the one above

e.g.: IP packet in UDP
IP in TCP
PPP(layer 2) packet in UDP
IPv4 in IPv6
IPv6 in IPv4

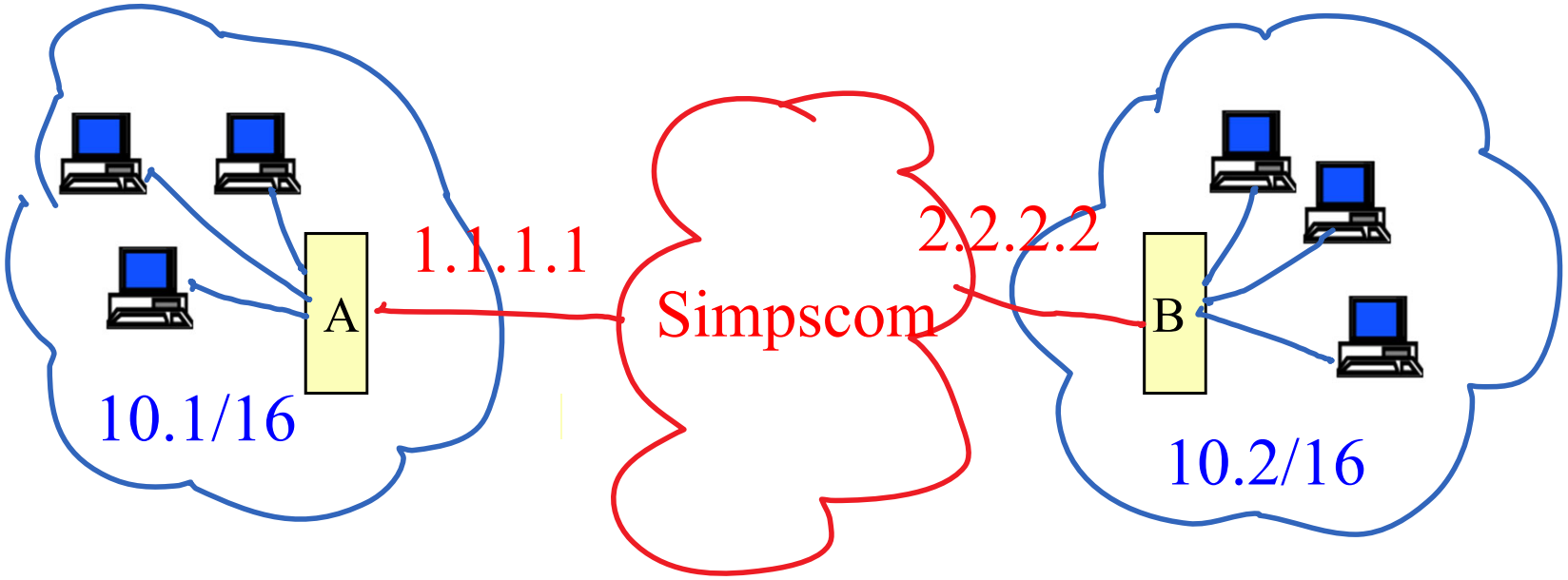
Why used ?

In theory: never

In practice: security / private networks / IPv6-IPv4 interworking

Homer's Network

Homer deploys 10.x addresses in two sites and wants to interconnect them as one (closed) private network



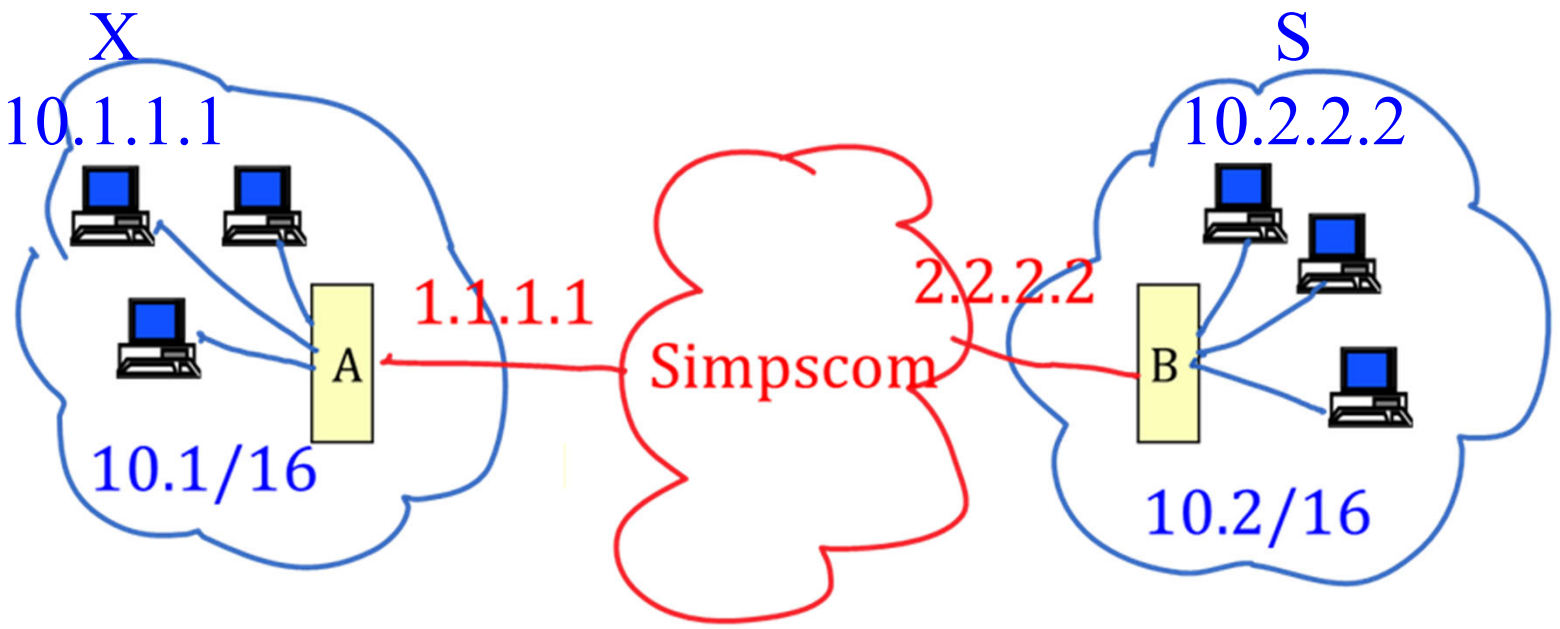
How can Homer use Simpscom's network for that ?

IP packets with destination or source address 10.x.x.x cannot be sent to the public internet !

Old-style solutions: leased lines – expensive (but secure).

Wide-spread solution: Virtual Private Networks.

Example 1: Homer uses an IP over IP Tunnel



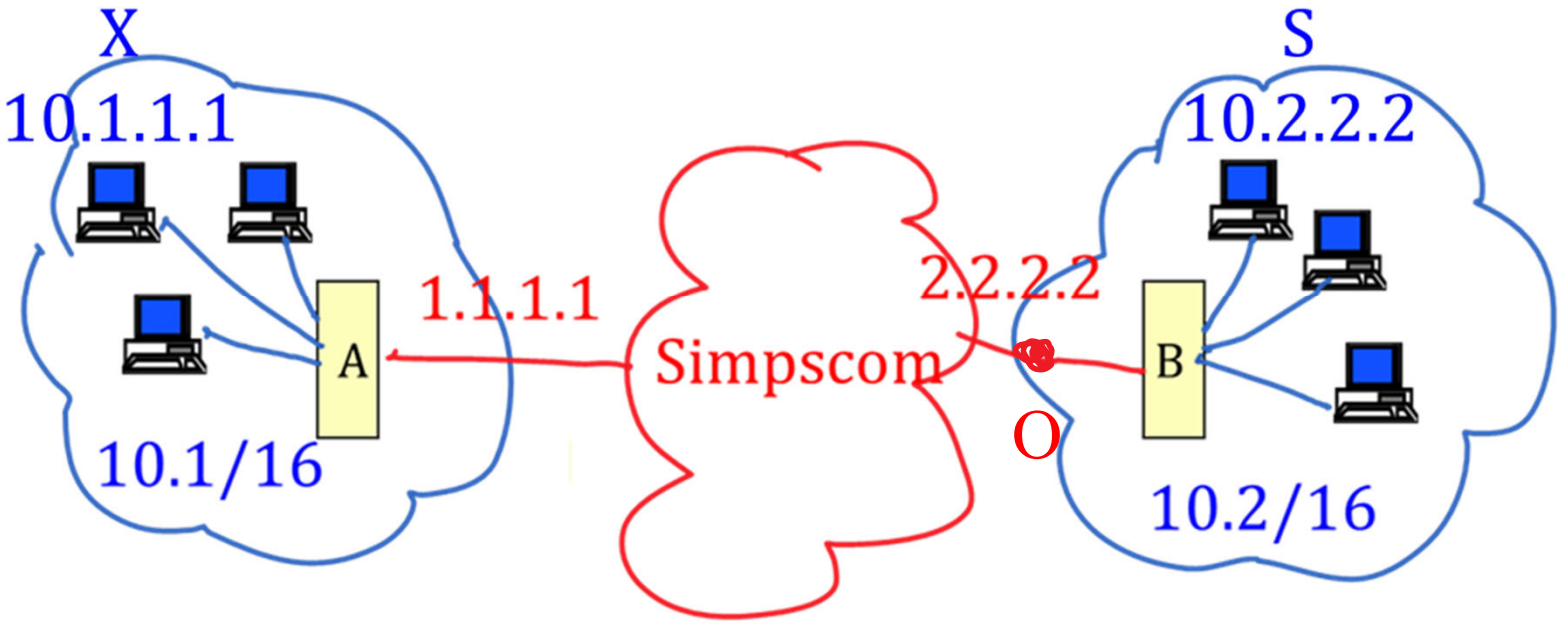
Homer configures a virtual interface in A (eth$x$); Associates this interface with an IP in IP tunnel, with endpoint 2.2.2.2

Similar stuff in B

Homer has a network with 2 routers and one virtual physical link; Homer configures routing tables at A and B (or runs a routing protocol)

IP Packets from S to X are carried inside IP packets across Simpscom

S sends a UDP packet to X.
What are the IP destination address and protocol at O ?

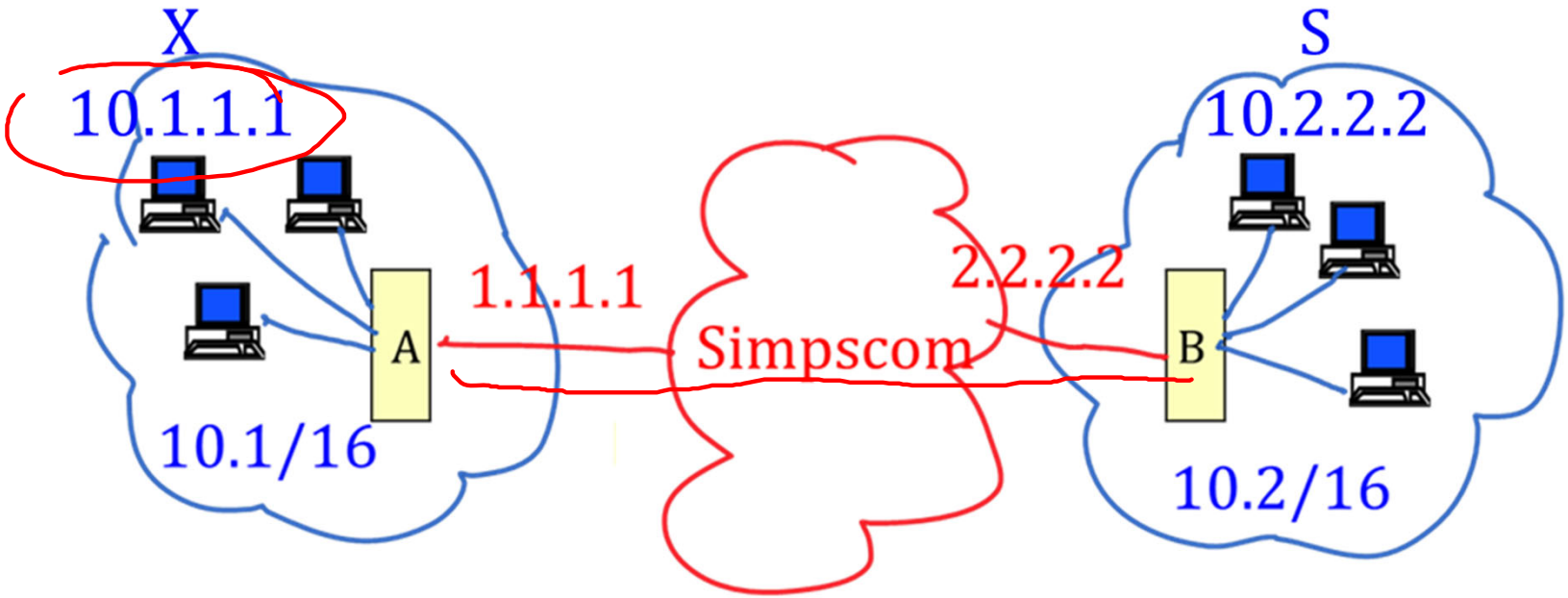


A. IP dest addr = 1.1.1.1, protocol = 17 (UDP)
B. IP dest addr = 10.1.1.1, protocol = 17 (UDP)
C. None of the above
D. I don't know

Solution

S sends a UDP packet to X.

What are the IP destination address and protocol at O ?



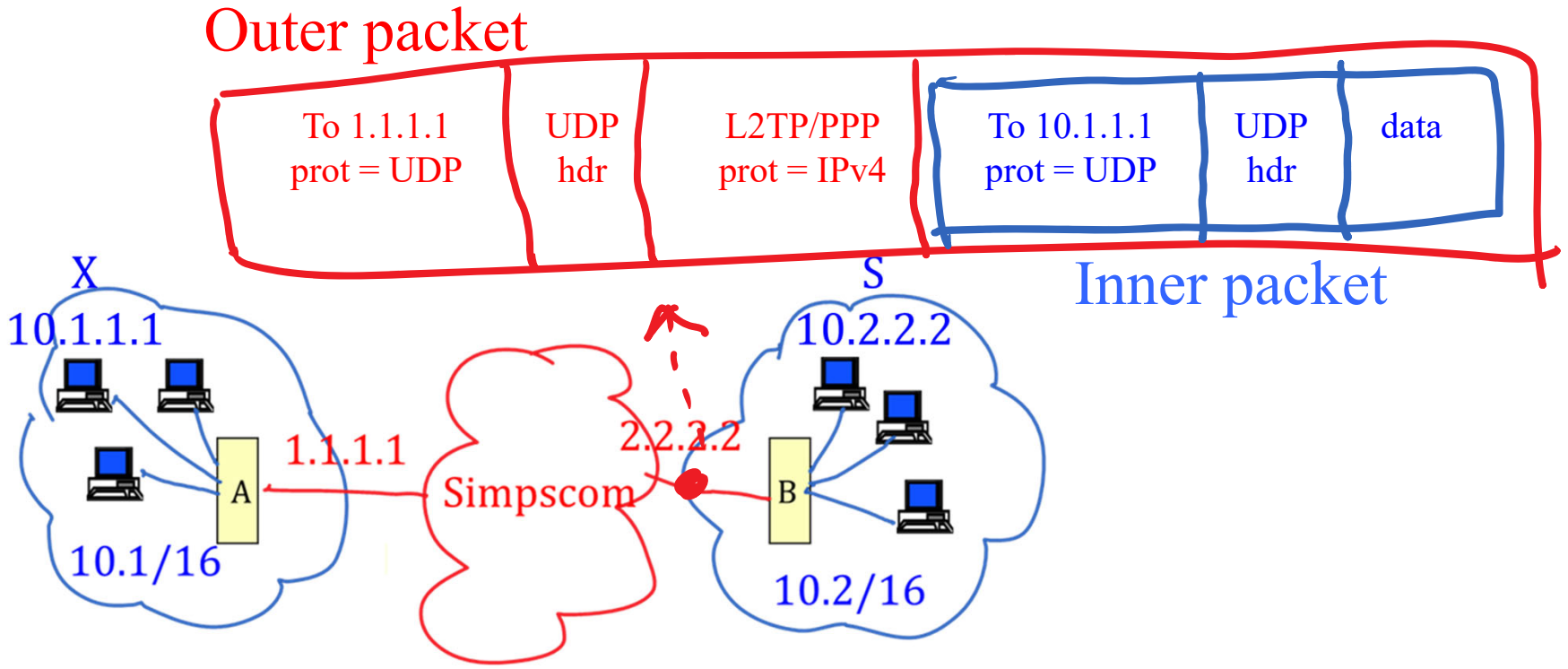
1. The IP destination address is the tunnel endpoint 1.1.1.1
2. The protocol is not UDP but 04 (IPv4)
   Answer C

Homer's IP in IP solution is often replaced by IP in UDP

Some firewalls kill IP in IP packets

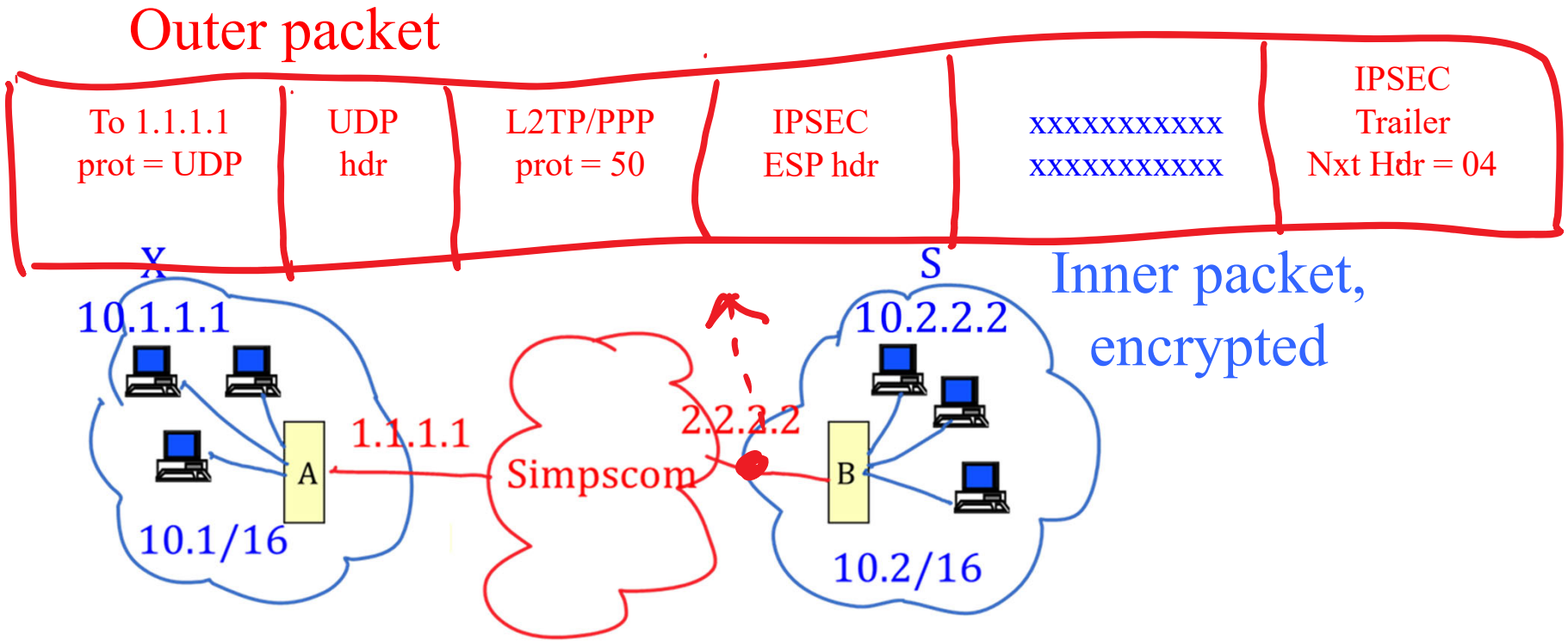
Therefore the tunnel is inside UDP

This requires a layer 2 header as well (to identify the protocol type) called L2TP / PPP



Bart does the same as Homer but wants a secure channel. He uses IPSEC.

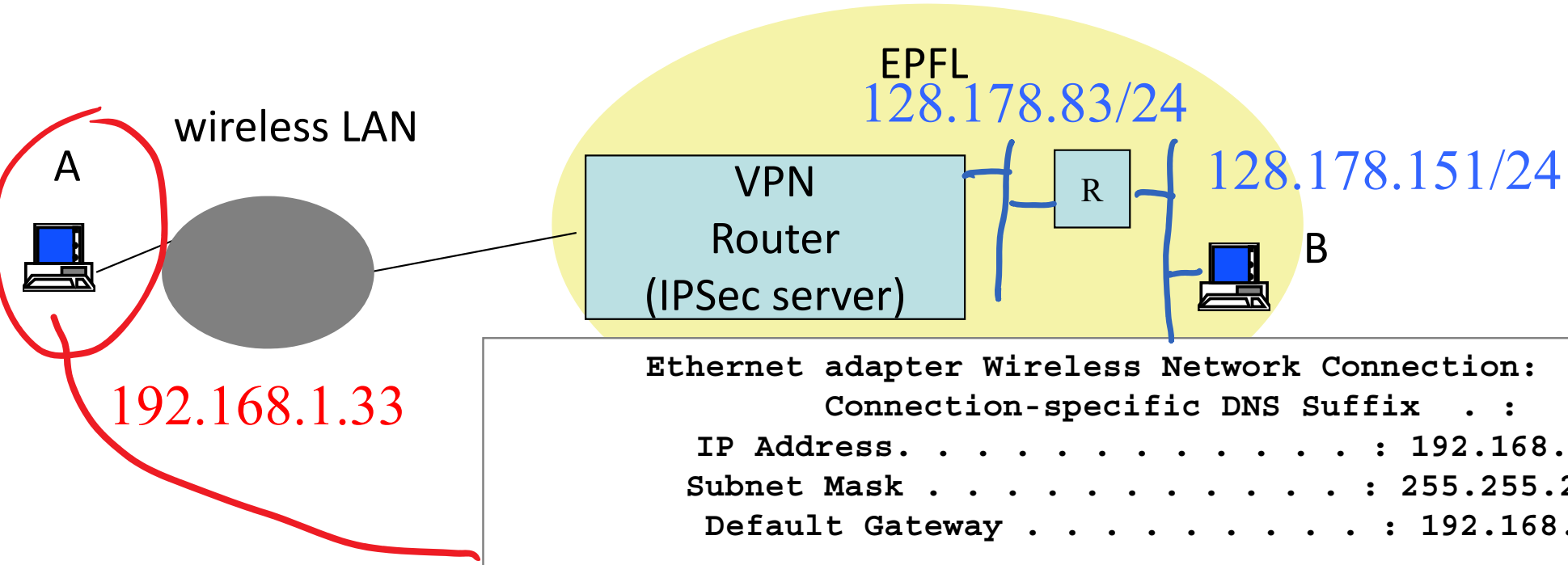
«IPSEC / ESP tunnel mode» encrypts the inner IP packet



This form of tunneling is called «L2TP/IPSEC VPN» (Virtual Private Network)

Variants (OpenVPN): IP in TLS over TCP ; IP in TLS over UDP

How does a packet from B to A find its way ?

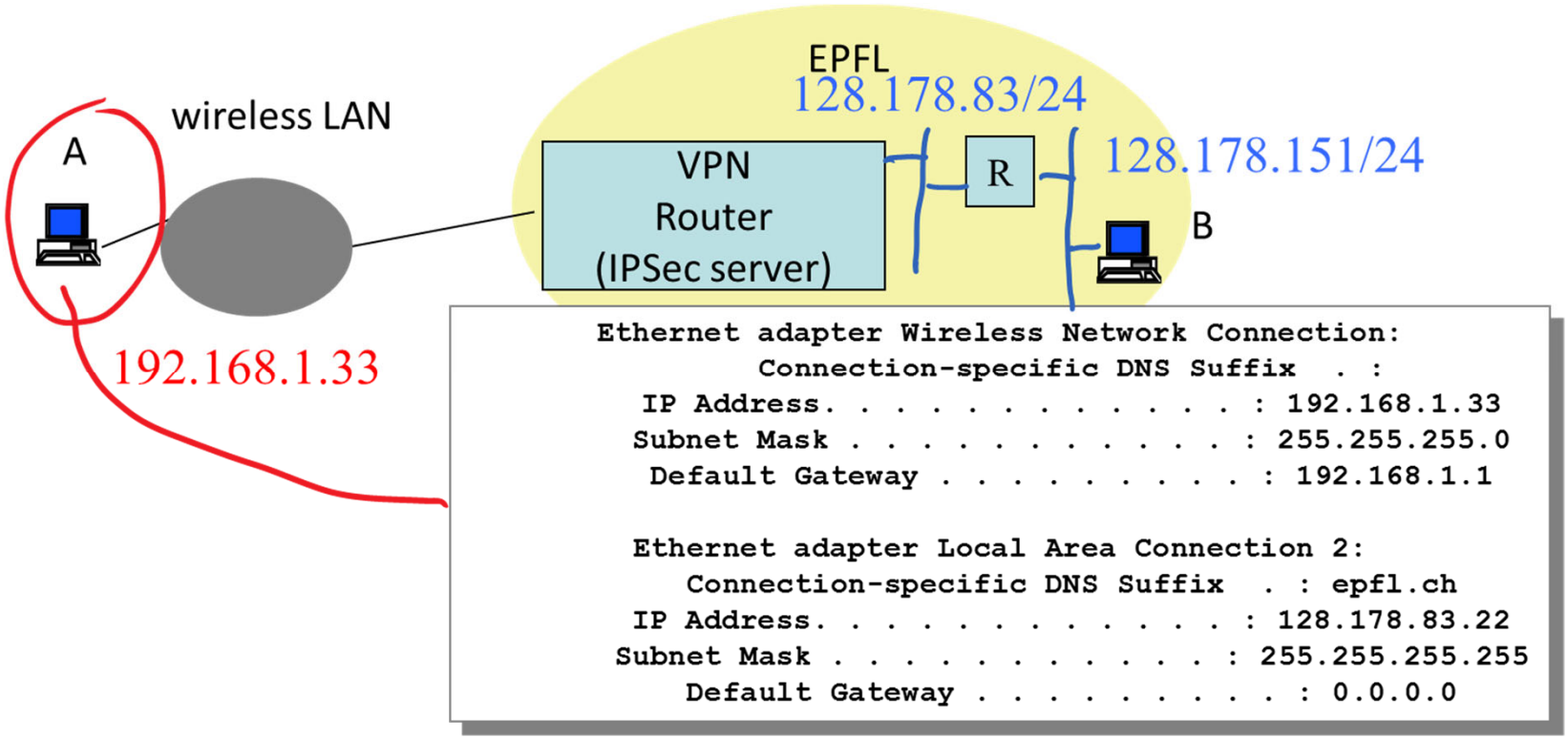



```
Ethernet adapter Wireless Network Connection:
        Connection-specific DNS Suffix  . :
   IP Address. . . . . . . . . . . . : 192.168.1.33
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Ethernet adapter Local Area Connection 2:
      Connection-specific DNS Suffix  . : epfl.ch
   IP Address. . . . . . . . . . . . : 128.178.83.22
   Subnet Mask . . . . . . . . . . . : 255.255.255.255
      Default Gateway . . . . . . . . . : 0.0.0.0
```

A. VPN router does proxy-ARP
B. R has a host route to A
C. Nothing special, the IGP takes care of it
D. I don't know

Solution



Answer A

A has two interfaces: one physical, with address 192.168.1.33, one virtual (tunnel) interface with address 128.178.83.99 (for example)

A appears to be on 128.178.83/24

VPN router does proxy ARP on behalf of A

R does not need a host route (but VPN Router may need one)

12. 6to6 over 4: Tunnel Brokers

IPv4 and IPv6 are incompatible

- v4 only host cannot handle IPv6 packets
- v6 only host cannot handle IPv4 packets

What needs to be solved:

interworking: h6 to h4

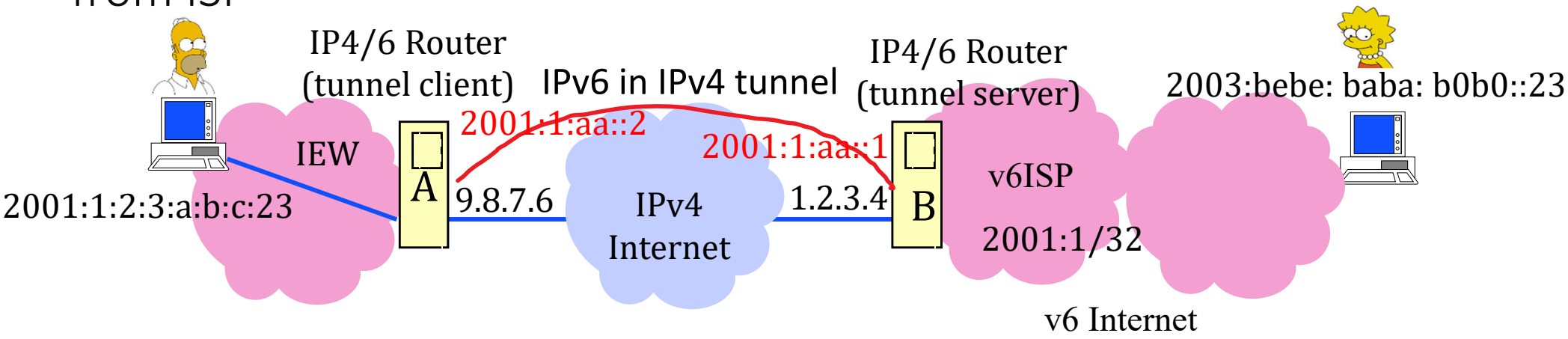
like-to-like access

- 6 to 6 over 4
- 4 to 4 over 6

In this and the next section we study mainly like-to-like

6 to 6 over 4 Solution: Tunnel Broker

Problem: 6to6 over 4 (early adopter) Homer runs IPv6 in local network, wants to connect to v6 hosts, but receives only IPv4 service from ISP



One solution: Tunnel Broker uses IPv6 in IPv4 encapsulation

Static tunnel configured at A; provided by v6ISP

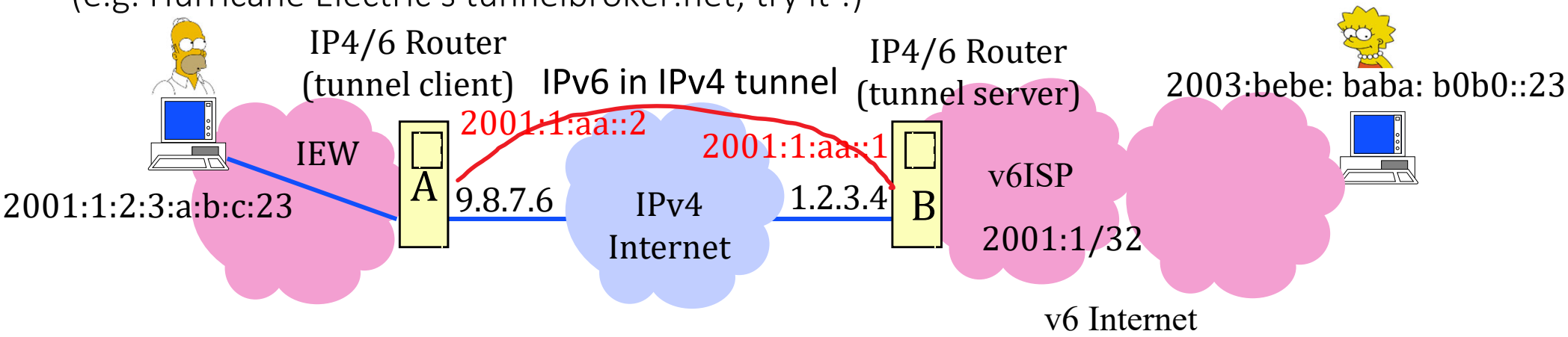
(e.g. Hurricane Electric's tunnelbroker.net; try it !)

6 to 6 over 4 Solution: Tunnel Broker

Tunnel Broker uses IPv6 in IPv4 encapsulation

Static tunnel configured at A; provided by v6ISP

(e.g. Hurricane Electric's tunnelbroker.net; try it !)



v6ISP delegates to IEW an IPV6 prefix e.g. 2001:1:2:3/64

v6ISP assigns the IPv6 addresses of tunnel end-points e.g. 2001:1:aa::2 and 2001:1:aa::1

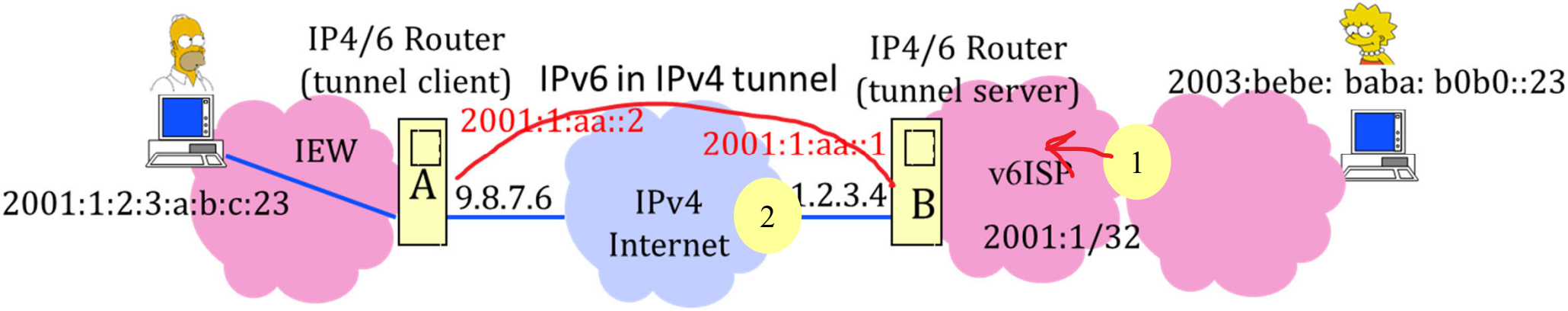
IEW can have multiple subnets

A's IPv6 default route is 2001:1:aa::1

To Lisa, Homer appears to be a customer of v6ISP

Other similar solutions: 6rd (tunnel endpoints are automatic), 6to4,Teredo

Lisa sends one packet to Homer; what do we see at (1) ?



A. The IPv4 destination address in the encapsulating packet is 9.8.7.6
B. The IPv4 destination address in the encapsulating packet is 1.2.3.4
C. The IPv6 destination address in the packet is 2001:1:aa::2
D. The IPv6 destination address in the packet is 2001:1:2:3:a:b:c:23
E. A and C
F. B and C
G. A and D
H. B and D
I. I don't know

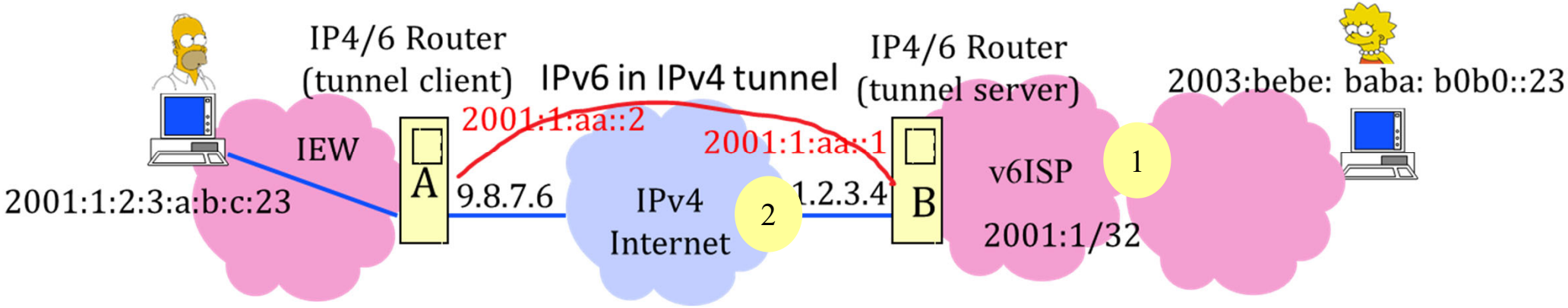
Solution

Answer D
To Lisa, Homer appears to be on the v6Internet. The packet is a regular IPv6 packet and its destination address is Homer's IPv6 address.

There is no encapsulated packet at this point.
Encapsulation is performed by B and is removed by A.

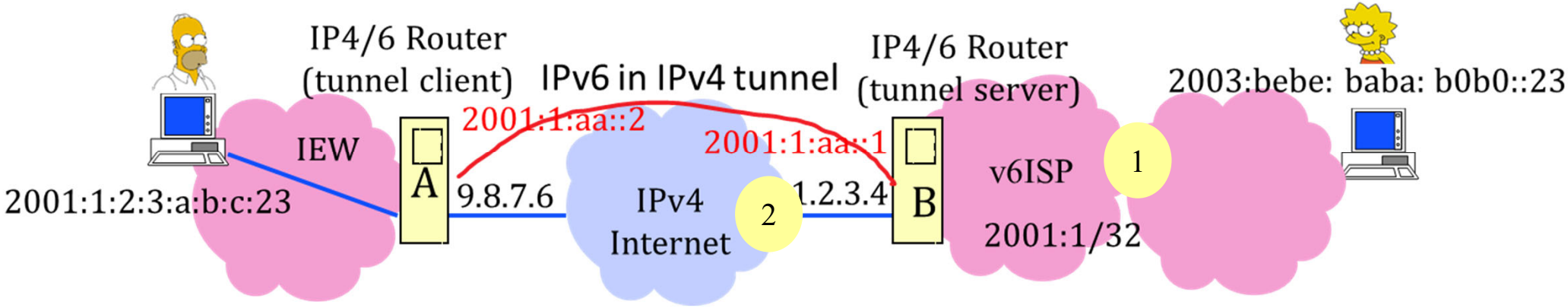
Between A and B (e.g. at (2)) we see an IPv4 packet with destination address 9.8.7.6, source address 1.2.3.4, protocol type = 41. Inside the IPv4 packet we see an IPv6 packet with destination address = Homer's IPv6 address.

All links are Ethernet v2 with MTU = 1500 Bytes. Assume all hosts perform Path-MTU and discover the best possible Path-MTU value. What is the value of Path-MTU between Lisa and Homer ?



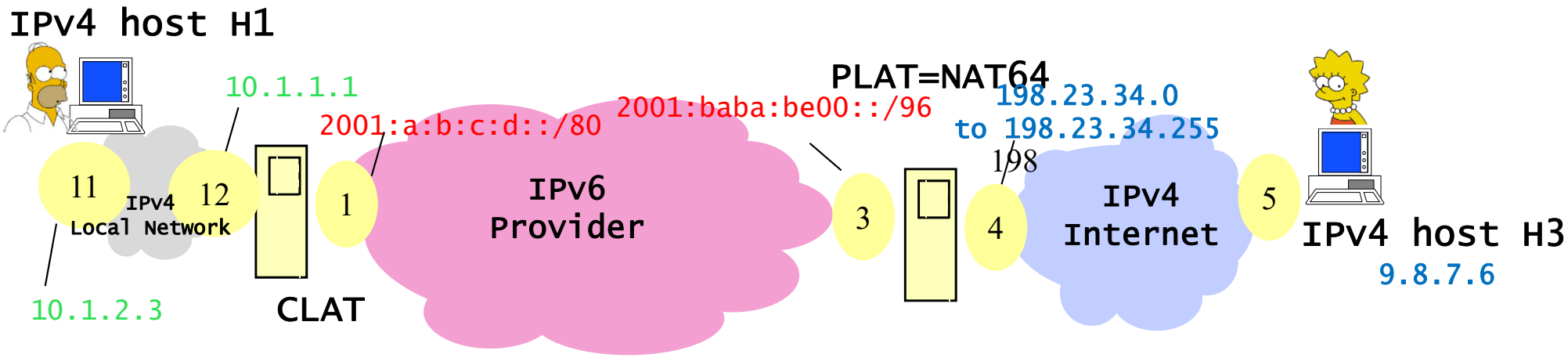
A. 1500 Bytes
B. 1480 Bytes
C. 1460 Bytes
D. None of these
E. I don't know

Solution



IPv6 packet of Lisa is encapsulated in an IPv4 packet by B. The encapsulation adds the IPv4 header, i.e. 20 bytes. The Path-MTU for Lisa to Homer is 1480 Bytes. Same thing in the reverse direction.

13. 4to4 over 6: 464XLAT



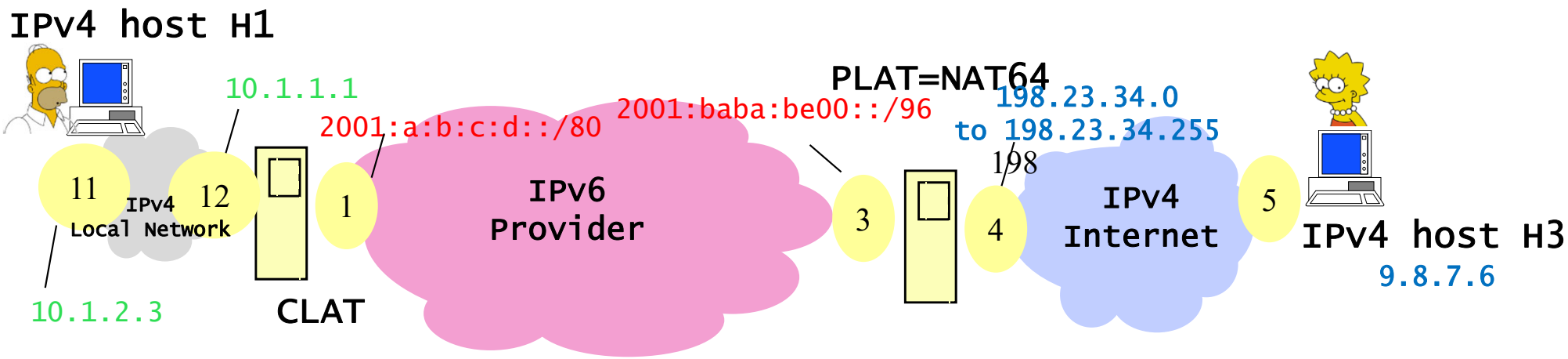
Problem: 4to4 over 6: (Legacy Problem) Homer's device is IPv4, Homer receives only IPv6 service from ISP and still wants to communicate with v4 host H3.

Homer's home network and device can run IPv6 but that does not solve all problems:

- IPv4-only applications (skype)
- IPv4-only remote correspondents (google scholar)

One Solution: 464XLAT ; uses NATs (XLAT = translation, no tunnels)

464XLAT: Customer-Side Translation



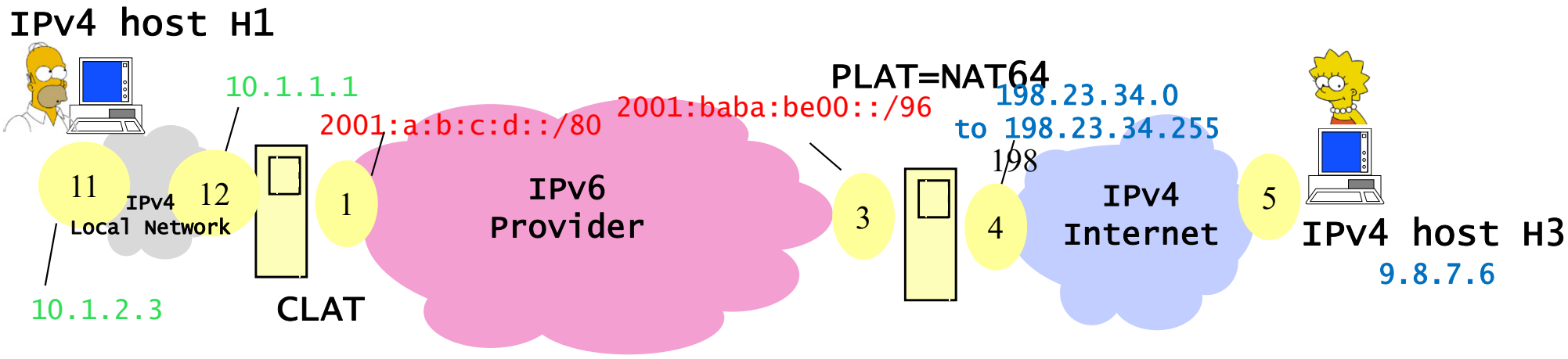
IPv6 provider reserves: one block of IPv6 addresses for the IPv4 internet (2001:baba:be00::/96), one block of IPv6 addresses per IPv4 customer (2001:a:b:c:d::/80 for Homer) and one block of IPv4 addresses for the set of all remote IPv4 customers such as Homer (198.23.34/24).

CLAT (**c**ustomer-side trans**lat**or) performs stateless address translation IPv4 <-> IPv6 for local and remote v4 addresses. It is a NAT, but does not modify port numbers.

10.1.2.3 is mapped to 2001:a:b:c:d::a01:203

9.8.7.6 is mapped to 2001:baba:be00::908:706

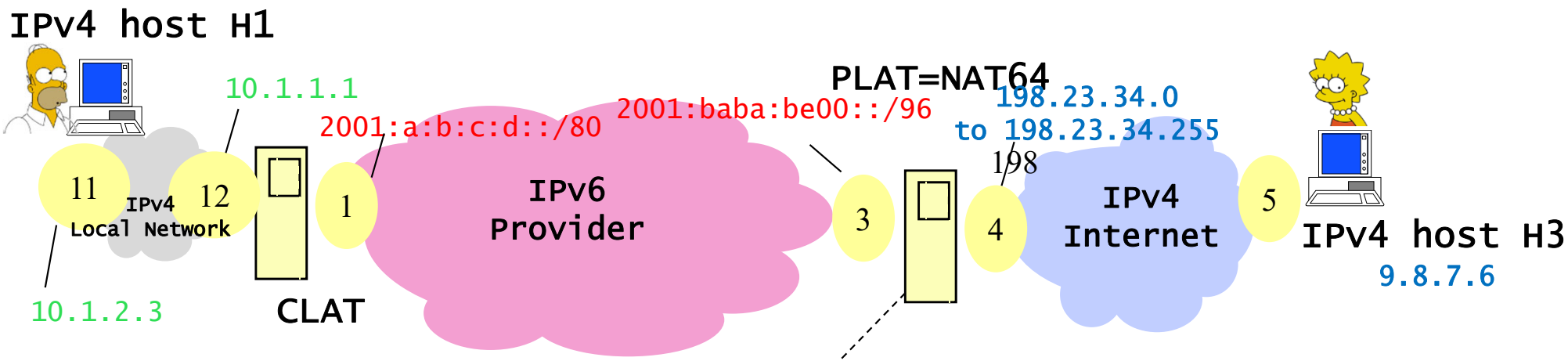
464XLAT: Provider-Side Translation



PLAT (provider-side translator), also called NAT64, performs stateful address translation IPv6 -> IPv4. Like a regular NAT, needs to modify port numbers. E.g. 2001:a:b:c:d::a01:203 port 3456 is mapped to 198.23.34.45 port 4567.

It also performs stateless address translation IPv4-> IPv6. E.g. 9.8.7.6 is mapped to 2001:baba:be00::908:706 (IPv4-embedded-IPv6 address)

PLAT (NAT64) is Stateful



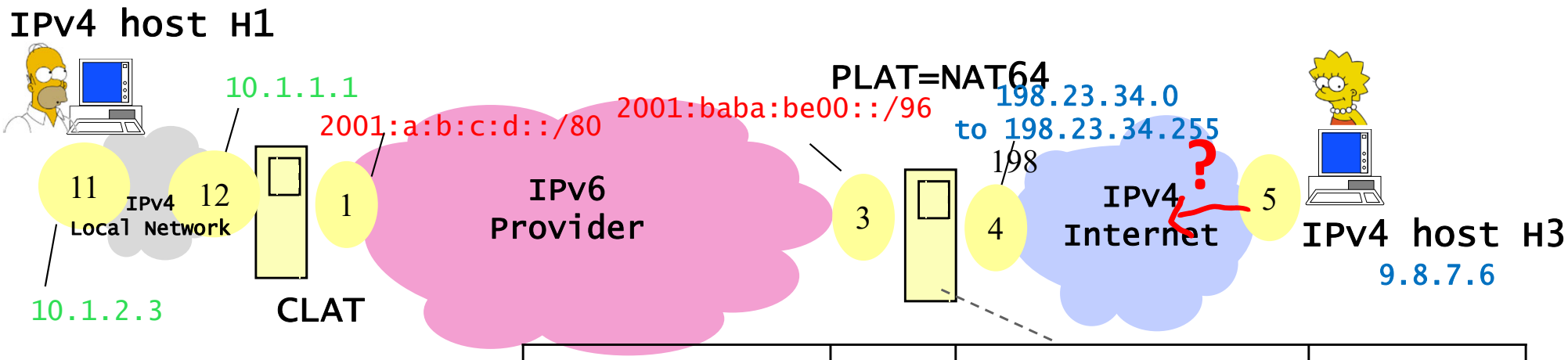
PLAT needs to remember the (v4 address, port) mapping + the IPv6 source address of Homer. In the NAT64 table we see:

IPv6 CLAT address	port	IPv4 translated address	translated port
2001:a:b:c:d::a01:203	3456	198.23.34.59	4567

PLAT does this for all customers and for every flow served by this provider.

The NAT table may be very large. This is called a “Carrier Grade NAT”.

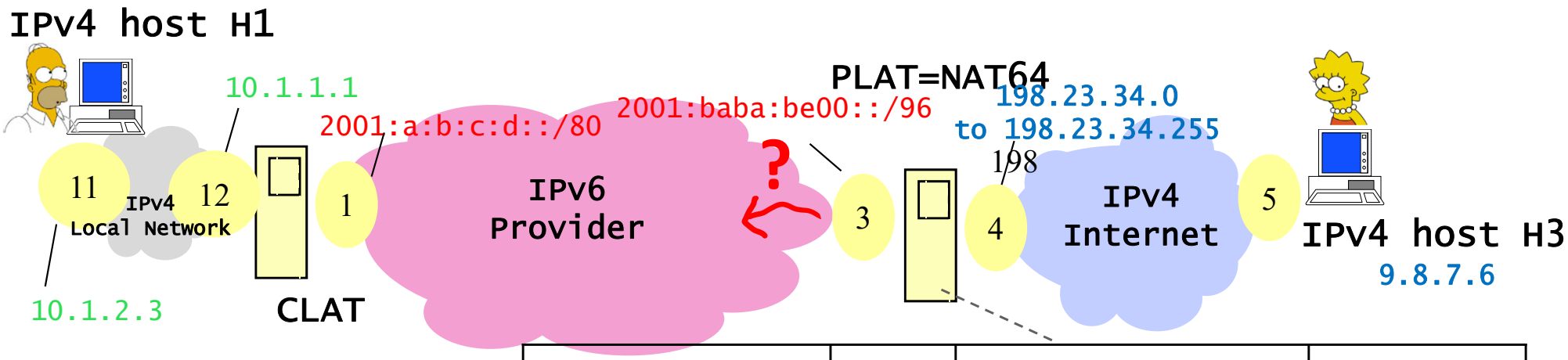
Homer sends one packet to Lisa and Lisa responds. We observe the response at 5. Say what is true.



IPv6 CLAT address	port	IPv4 translated address	translated port
2001:a:b:c:d::a01:203	3456	198.23.34.59	4567

A. The IPv4 destination address in the packet is 10.1.2.3
B. The IPv4 destination address in the packet is 198.23.34.59
C. The IPv6 destination address in the packet is 2001:a:b:c:d::a01:203
D. A and C
E. B and C
F. I don't know

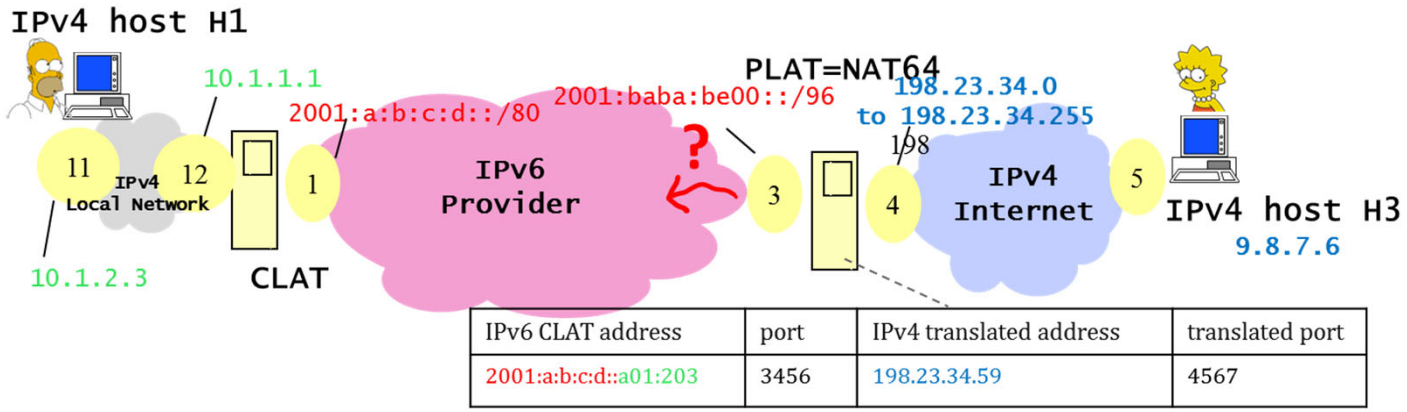
Homer sends one packet to Lisa and Lisa responds. We observe the response at 3. Say what is true.



IPv6 CLAT address	port	IPv4 translated address	translated port
2001:a:b:c:d::a01:203	3456	198.23.34.59	4567

A. The IPv4 destination address in the packet is 10.1.2.3
B. The IPv6 destination address in the packet is 2001:a:b:c:d::a01:203
C. The IPv6 source address in the packet is 2001:baba:be00::908:706
D. A and C
E. B and C
F. A, B and C
G. I don't know

Solution



IPv6 CLAT address	port	IPv4 translated address	translated port
2001:a:b:c:d::a01:203	3456	198.23.34.59	4567

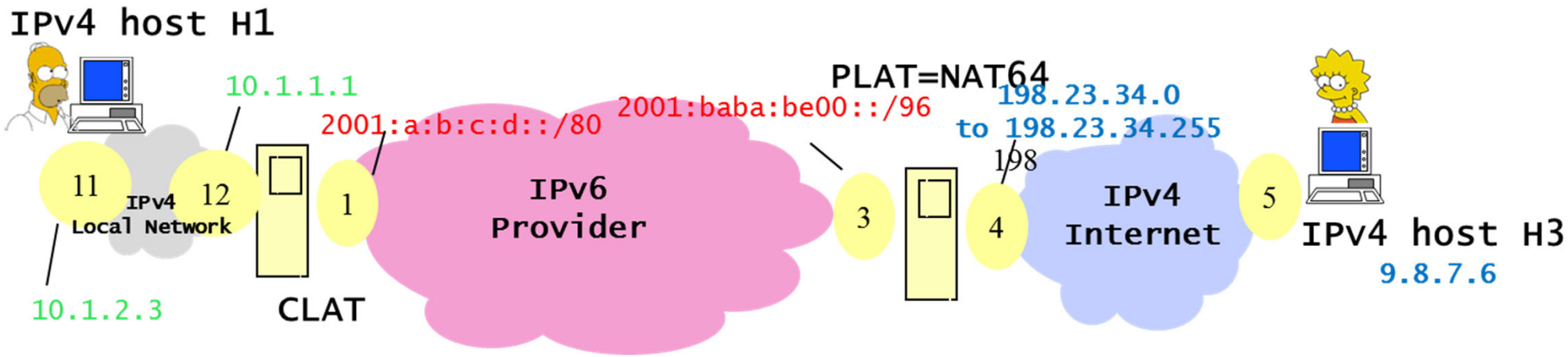
Answer B

There is no encapsulated IPv6 packet at (5), nor anywhere else in this scenario.

Answer E (B and C are true).

There is no encapsulated IPv4 packet at (3), nor anywhere else in this scenario.

All links are Ethernet v2 with MTU = 1500 Bytes. Assume all hosts perform Path-MTU and discover the best possible Path-MTU value. What is the value of Path-MTU between Lisa and Homer ?



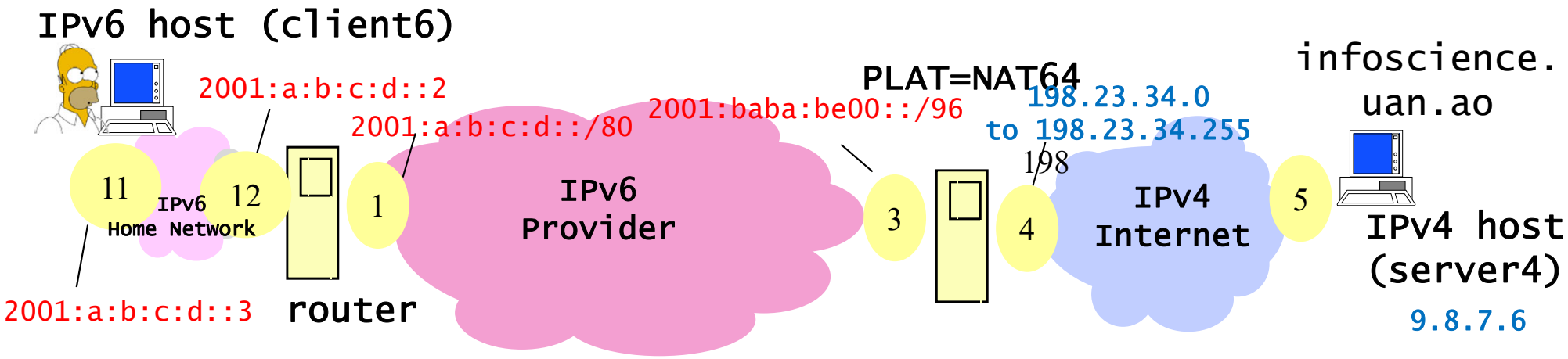
A. 1500 Bytes
B. 1480 Bytes
C. 1460 Bytes
D. None of these
E. I don't know

Solution

The IPv4 packets at points (4) and (12) are translated into IPv6 packets at points (1) and (3). During the translation, the header is increased by 20 bytes (from 20B IPv4 header to 40B IPv6 header) – assuming there are no header options. So the packet size increases by 20 B at these points.

Therefore the Path MTU is 1480 B in this case. If the IPv6 path uses header options (e.g. for segment routing), this is further reduced.

14. NAT64 and DNS64

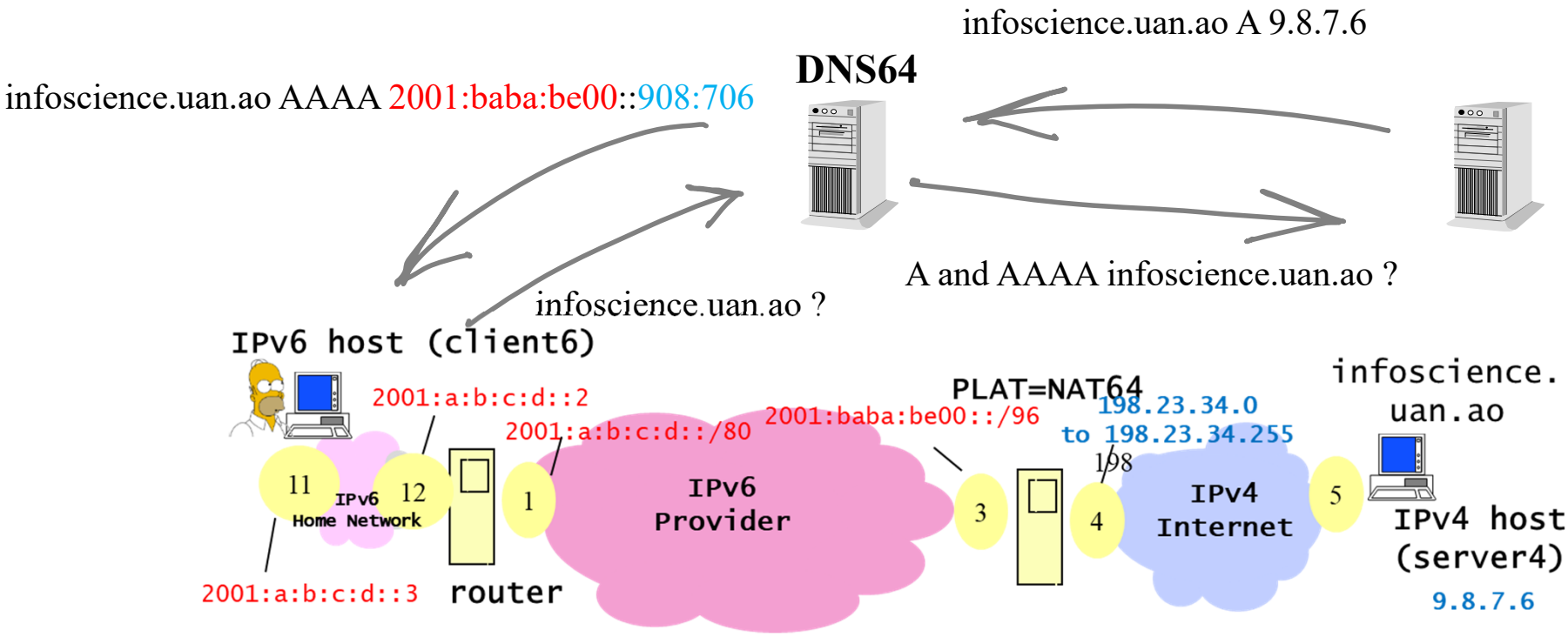


Problem: 6to4 (Interworking Problem) Homer has IPv6-only service and IPv6 only host and wants to communicate with v4 host
One solution: re-use elements of 464XLAT

v6ISP reserves one block of IPv6 addresses for the IPv4 internet (2001:baba:be00::/96), one block of IPv4 addresses for all remote IPv6 customers (198.23.34/24) and a NAT64; NAT64 performs stateful header translation [NAT64 is the same as PLAT].

To client6, server4 appears under the IPv4-embedded-IPv6 address 2001:baba:be00::908:706

How does client6 know the IPv4-embedded-IPv6 address of server4 ?



DNS64 is used in combination with stateful NAT64.

DNS64 responds with translated IPv6 address if no AAAA record is found.

This is deployed by v6ISP and is transparent to client6.

Mechanisms for Transition to IPv6

- Like-to-like access
  - 4to4 over 6: 464XLAT,
    - MAP-E, 4rd: similar to 464XLAT but stateful address translation is performed on customer side (scalable)
    - MAP-T same as MAP-E with encapsulation instead of NATs
  - 6to6 over 4: Tunnel brokers, 6rd, Teredo
- Interworking
  - With NATs : NAT64, DNS64
  - With Application Layer Gateways
- Example: mobile operator launches IPv6-only service
  - Android devices support 464XLAT (CLAT in device)
  - IOS devices do not but require that all apps work with IPv6
  - ⇒ mobile operator deploys NAT64 (=PLAT, for Android and for IOS) and DNS64 (for IOS)

Conclusion

Proxy ARP / ND Proxy is a trick used to solve the problems caused by a subnet present at different locations

Fragmentation is due to different MAC layers having different packet sizes. Fragmentation occurs only at IPv6 hosts, IPv4 hosts or IPv4 routers. Re-assembly is never done by routers. Fragmentation may cause problems and should be avoided if possible.

Tunnels are used e.g. to create virtual private networks

Transition to IPv6 creates many problems that can be solved with various methods involving automatic tunnels, header translation (CLAT, NAT64 = PLAT,) and DNS manipulation (DNS64).