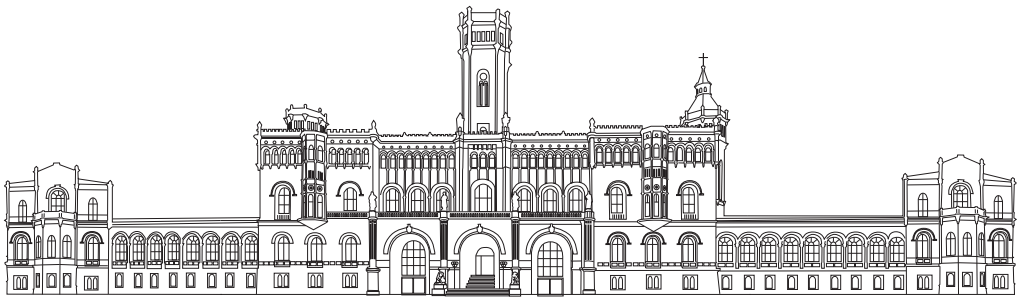
Rechnernetze - Computer Networks

Problem Set 9 - Internetworking
Addressing

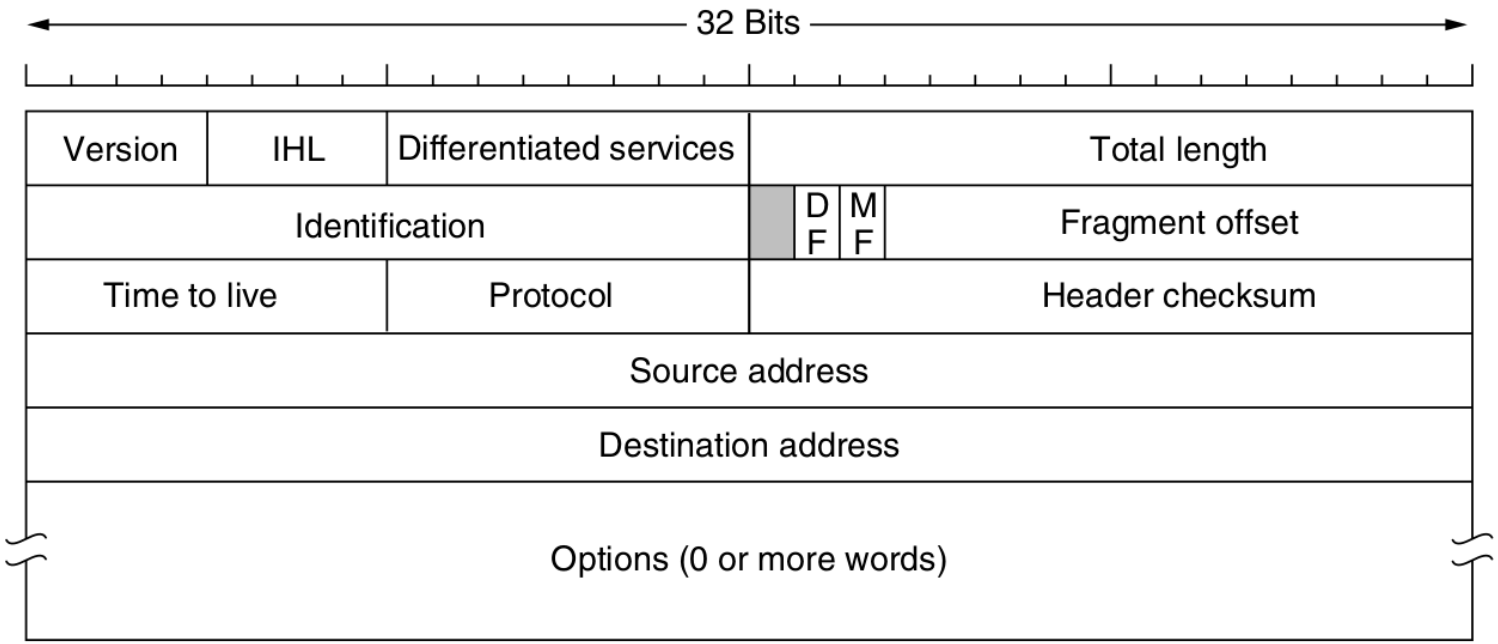
Markus Fidler, Mark Akselrod, Lukas Prause



Institute of Communications Technology
Leibniz Universität Hannover

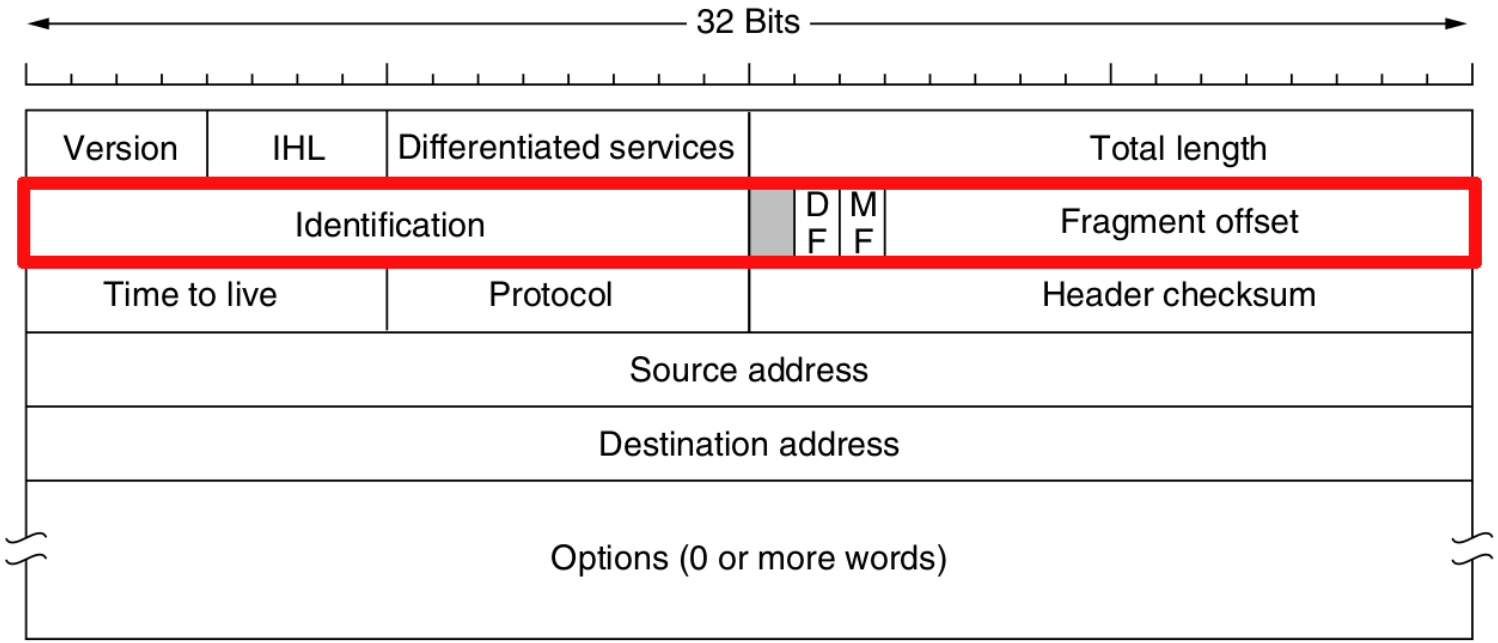
June 17, 2024

Which of the IP header fields are used for fragmentation?



[Source: Tanenbaum, Computer Networks]

IP Datagram Fragmentation

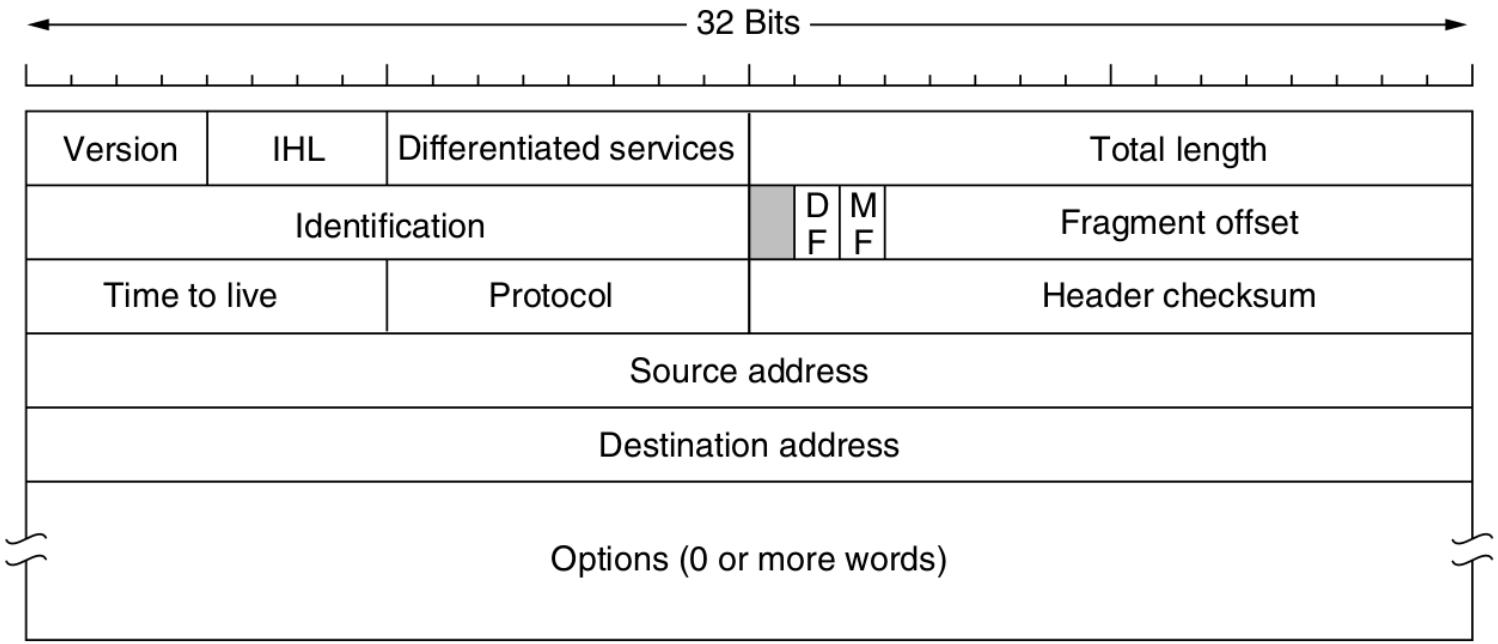


[Source: Tanenbaum, Computer Networks]

IP Datagram Fragmentation

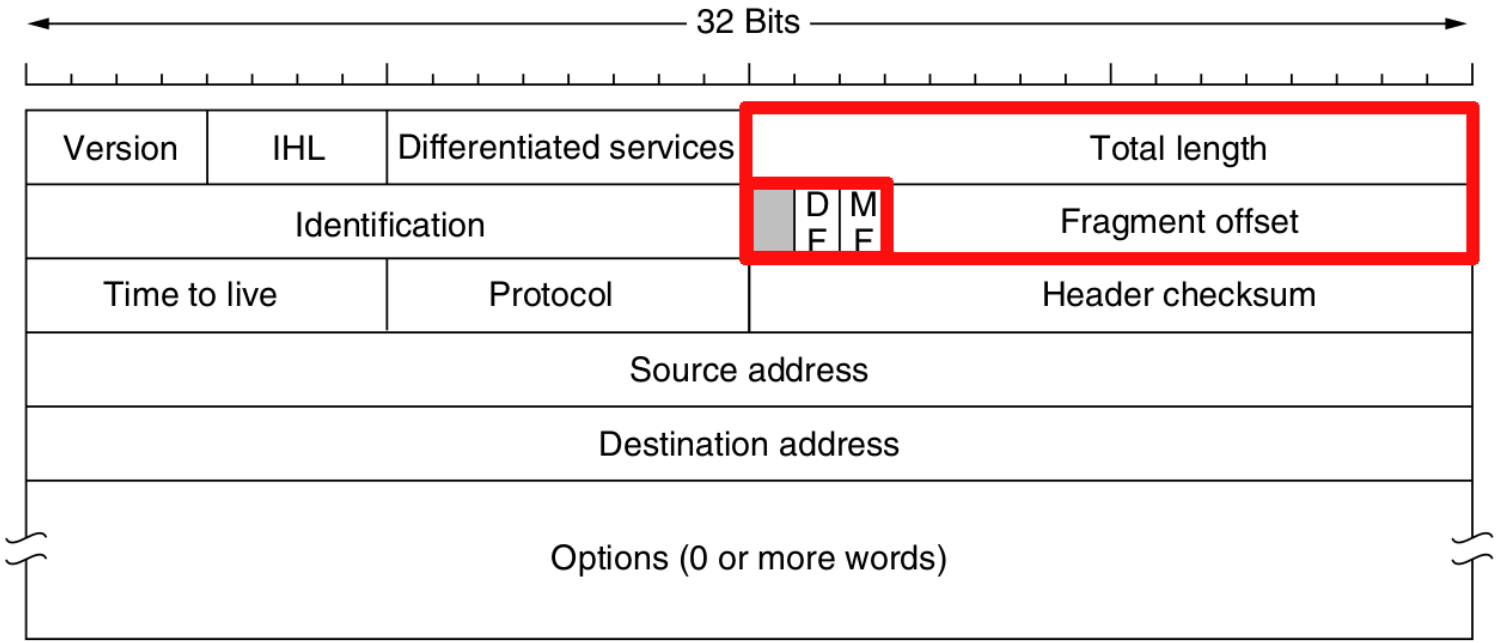
- *Identification*: needed to allow the destination host to determine which packet a newly arrived fragment belongs to
- *DF* stands for Don't Fragment. It is an order to the routers not to fragment the packet
- *MF* stands for More Fragments. All fragments except the last one have this bit set.
- The *Fragment offset* tells where in the current packet this fragment belongs. All fragments except the last one in a datagram must be a multiple of 8 bytes.

Why must fragment length be a multiple of 8 bytes?



[Source: Tanenbaum, Computer Networks]

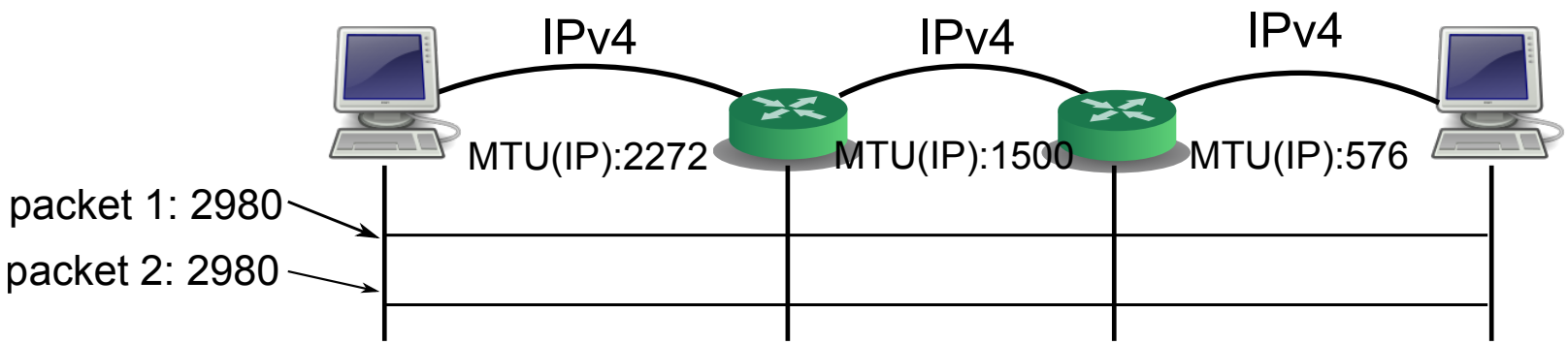
IP Datagram Fragmentation



[Source: Tanenbaum, Computer Networks]

- Because of the flag/reverved bits, the *Fragment offset* field is 3 bits shorter than the *Total length* field
- *Fragment offset* must therefore be in multiples of $2^3 = 8$ bytes so that the maximum packet length is supported by fragmentation

Two successive IPv4 packets with a size $L = 3000$ byte (including the IP header) are sent over links with MTUs of 2272 byte, 1500 byte, and 576 byte. How will the packets be fragmented (use random identification numbers)? What overhead will be introduced (assume 20 byte IP headers)?



At the **first hop** with MTU = 2272:

Packet 1 contains 2980 B of data:

packet	fragment	length	Identification	MF	offset	data
1	1	2268	123	1	0	2248
1	2	752	123	0	281	732

Because of the MTU of 2272 Byte, one IP fragment can hold 2252 bytes of data and a 20 bytes IP header

However: 2252 is not a multiple of 8 $\rightarrow$ 2248 is the highest possible multiple of 8 The second fragment is offset by 2248 bytes. Since the *Fragment offset* field is given in multiples of 8 bytes, the offset is 2248 / 8 = 281

Since packet 2 has the same size as packet 1, it is fragmented in the same way. Only the Identification has to be different so that the receiver can determine which packet a newly arrived fragment belongs to:

packet	fragment	length	Identification	MF	offset	data
1	1	2268	123	1	0	2248
1	2	752	123	0	281	732

packet	fragment	length	Identification	MF	offset	data
2	1	2268	456	1	0	2248
2	2	752	456	0	281	732

At the **first hop** with MTU = 2272:

packet	fragment	length	Identification	MF	offset	data
1	1	2268	123	1	0	2248
1	2	752	123	0	281	732

At the **second hop** with MTU = 1500:

packet	fragment	length	Identification	MF	offset	data
1	1	1500	123	1	0	1480
1	2	788	123	1	185	768
1	3	752	123	0	281	732

Packet 2 is fragmented in the same way:

packet	fragment	length	Identification	MF	offset	data
1	1	1500	123	1	0	1480
1	2	788	123	1	185	768
1	3	752	123	0	281	732

packet	fragment	length	Identification	MF	offset	data
2	1	1500	456	1	0	1480
2	2	788	456	1	185	768
2	3	752	456	0	281	732

IP Datagram Fragmentation

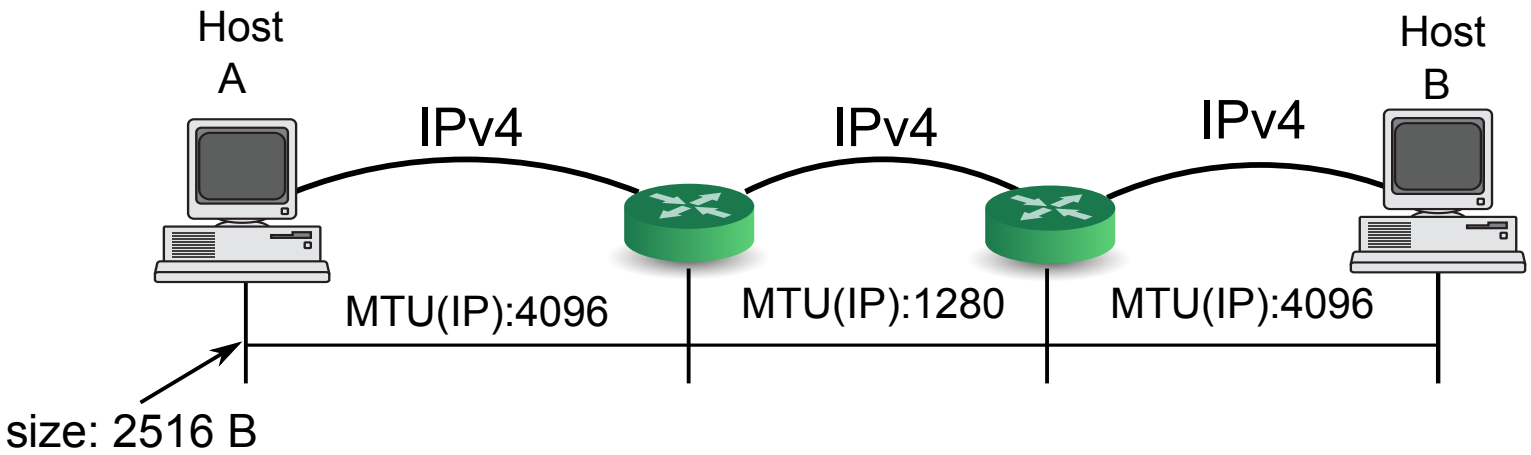
At the **second hop** with MTU = 1500:

packet	fragment	length	Identification	MF	offset	data
1	1	1500	123	1	0	1480
1	2	788	123	1	185	768
1	3	752	123	0	281	732

At the **third hop** with MTU = 576, the packets are fragmented once again:

packet	fragment	length	Identification	MF	offset	data
1	1	572	123	1	0	552
1	2	572	123	1	69	552
1	3	396	123	1	138	376
1	4	572	123	1	185	552
1	5	236	123	1	254	216
1	6	572	123	1	281	552
1	7	200	123	0	350	180

An IPv4 packet with a size $L = 2536$ byte (including the IP header) is sent over links with MTUs of 4096 byte, 1280 byte, and 4096 byte. How will the packet be fragmented (use random identification numbers)? What overhead will be introduced (assume 20 byte IP headers)?



At the **first hop**, the Packet sent from Host A contains 2516 B of data. Since the MTU is large enough, no fragmentation will take place:

packet	fragment	length	Identification	MF	offset	data
1	1	2536	123	0	0	2516

IP Datagram Fragmentation

At the **second hop** with MTU = 1280, the following fragmentation will take place. The router breaks it into 2 fragments:

packet	fragment	length	Identification	MF	offset	data
1	1	1276	123	1	0	1256
1	2	1280	123	0	157	1260

In the last fragment the data length doesn't have to be a multiple of 8 bytes, hence we can fit the last 4 bytes into the second fragment.

At the **third** and last hop, there is no fragmentation since the MTU is larger than the fragments:

packet	fragment	length	Identification	MF	offset	data
1	1	1276	123	1	0	1256
1	2	1280	123	0	157	1260

Note that the fragments are not reassembled by the router even though the MTU would allow for larger fragments again.

Fragmentation ≠ Segmentation

What is the difference between fragmentation and segmentation?

Fragmentation ≠ Segmentation

- Fragmentation
  - IP layer functionality
    - IP mechanism which breaks datagrams into smaller fragments, so that they do not exceed the maximal transmission unit (MTU) limit of a given link
- Segmentation
  - Transport layer functionality
    - applies only at end-systems, i.e. hosts
    - intermediate systems, i.e. routers, cannot perform segmentation
  - implemented in TCP

A private home network with 128 hosts

How would you setup a private home network with 128 hosts?

- Which address ranges are you allowed to use?
- What is the minimum number of host bits that your subnet needs to have?

A private home network with 128 hosts

The following IP address ranges can be freely used in a home network:

$$\begin{aligned} 10.0.0.0 &- 10.255.255.255 \\ 172.16.0.0 &- 172.31.255.255 \\ 192.168.0.0 &- 192.168.255.255 \end{aligned}$$

To address 128 hosts, 8 bits are required ($2^8 - 2 = 256 - 2$) and thus the netmask would have to be 24 bits long (notation: $/24$ or $255.255.255.0$). Addresses with all host bits set to zeros, and all host bits set to ones are reserved.

The private network could use the addresses/subnet combination:
172.16.0.0/24 with the netmask $255.255.255.0$

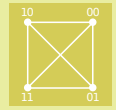
Hosts in the subnet $192.168.1.0/26$?

How many hosts can be addressed in the subnet $192.168.1.0/26$? What is the address range? What is the associated netmask in the $a.b.c.d$ notation?

Hosts in the subnet $192.168.1.0/26$?

- In subnet $192.168.1.0/26$ the first 26 bits identify the network
- 6 bits remain to address hosts. The lowest and highest host IDs are reserved
- only $2^6 - 2 = 62$ hosts are addressable
- The address range is $192.168.1.1 - 192.168.1.62$
- The netmask in binary notation is
  $11111111.11111111.11111111.11000000$
  (26 highest bits are set to 1 since the subnet prefix is 26 bits long. 6 lowest bits are set to zero, as they are used to address the hosts.)
- The netmask in the $a.b.c.d$ notation is $255.255.255.192$

Subnets A,B and C



What addresses would you assign to three subnets A, B and C considering the following constraints: networks A and B should be able to contain 5000 hosts each and network C should be able to support 500 hosts. All addresses must be allocated from $130.83.128.0/17$.

Subnets A,B and C

In $130.83.128.0/17$, we can use 15 bits to address hosts or divide the address space into additional subnets:

$$10000010.01010011.10000000.00000000$$

To address 5000 hosts, we need 13 host address bits
(sufficient for $2^{13} - 2 = 8190$ hosts)

$$10000010.01010011.1XX00000.00000000$$

$\rightarrow$ 2 additional bits can be used to address different subnets.

Subnets A,B and C

If we choose 00 for A and 01 for B we get:

$$A : 10000010.01010011.10000000.00000000 \text{ or}$$

$$130.83.128.0/19$$

and

$$B : 10000010.01010011.10100000.00000000 \text{ or}$$

$$130.83.160.0/19$$

Subnets A,B and C

For the subnet C we only need 9 host address bits (sufficient for $2^9 - 2 = 510$ hosts) and therefore can choose 6 additional bits to address the subnet:

$$10000010.01010011.1XXXXXX0.00000000$$

However, XXXXXX cannot start with 00 or 01, as these are already used by subnets A and B.

If we choose 100000 we get:

$$C : 10000010.01010011.11000000.00000000 \text{ or}$$

$$130.83.192.0/23$$

Subnets A,B and C

The resulting IP ranges look like this:

subnet	hosts	usable IP range
$130.83.128.0/19$	8190	$130.83.128.1 - 130.83.159.254$
$130.83.160.0/19$	8190	$130.83.160.1 - 130.83.191.254$
$130.83.192.0/23$	510	$130.83.192.1 - 130.83.193.254$

Addresses on the same Network

A sender with the IP $130.75.64.184$ and netmask $255.255.248.0$ wants to send an IP packet to the address $130.75.68.10$. Is the destination address on the same subnet or will the packet need to be routed? What is the address of the attached network in *a.b.c.d/p* notation?

Addresses on the same Network

Two addresses are on the same network if

$$(\text{SRC.ADDR} \;\&\; \text{NETMASK}) = (\text{DST.ADDR} \;\&\; \text{NETMASK})$$

$$(130.75.\mathbf{64.184} \;\&\; 255.255.248.0) \stackrel{?}{=} (130.75.\mathbf{68.10} \;\&\; 255.255.248.0)$$

$$\begin{array}{rrr} & 01000000\ 10111000 & (64.184) \\ \& & 11111000\ 00000000 & (248.0) \\ \hline & 01000000\ 00000000 & (64.0) \end{array} \stackrel{?}{=} \begin{array}{rrr} & 01000100\ 00001010 & (68.10) \\ \& & 11111000\ 00000000 & (248.0) \\ \hline & 01000000\ 00000000 & (64.0) \end{array}$$

Hence, the two hosts are on the same network: $130.75.64.0/21$

Default Route

What is the default route?

Default Route

- The default route is used to forward packets which do not match any other routes/destinations in the routing table to a default gateway.
- The default route has the address $0.0.0.0/0$ and therefore, due to longest prefix matching, it is always the last address to be looked up.
- In home routers the default gateway is used to forward all external traffic to the ISPs gateway.

Differences between IPv4 and IPv6

What are the main differences between IPv4 and IPv6 and what is the motivation behind these?

Differences between IPv4 and IPv6

- IPv6 uses 128 bit long addresses to provide a sufficient number of addresses
- A fixed length, 40 bytes header is used in IPv6. There are no header options in order to simplify and speedup packet processing at routers
- No fragmentation at intermediate routers and no checksums $\rightarrow$ speeds up packet processing at routers
- The IPv6 header contains a flow field which can be used to label connections

IPv4-mapped IPv6 addresses

How can the IPv4 address 137.226.4.59 be expressed as an IPv6 address? When can this notation be used?

IPv4-mapped IPv6 addresses

To express an IPv4 adress in IPv6 notation, 80 zeros and 16 ones have to be attached in front of it. The IPv4 address 137.226.4.59 can be written as:

- 0000:0000:0000:0000:0000:FFFF:89E2:043B
- ::FFFF:89E2:043B
- ::FFFF:137.226.4.59

This notation can be used by applications that run on dual stack machines (i.e. they support both IPv4 and IPv6) in order to work with both IPv4 and IPv6 addresses in the same format.

Network Address Translation Router (NAT)

A NAT-enabled router

- appears to the outside world (Internet) as a single device
- hides the home network's internals from the outside world

- all outgoing datagrams have the same source IP address
- all incoming datagrams have the same destination IP address

- maintains a translation table for the home network's IP addresses to certain port fields

Network Address Translation Router (NAT)

Modification of outgoing datagrams (from the local network)

- substitute the source IP address by the global IP address of the NAT router
- substitute the source port by an arbitrary, available source port
- send modified datagram into the Internet

Entry into the NAT translation table

- one-to-one mapping of source IP address and source port to NAT IP address and new source port

Modification of incoming datagrams (from the Internet)

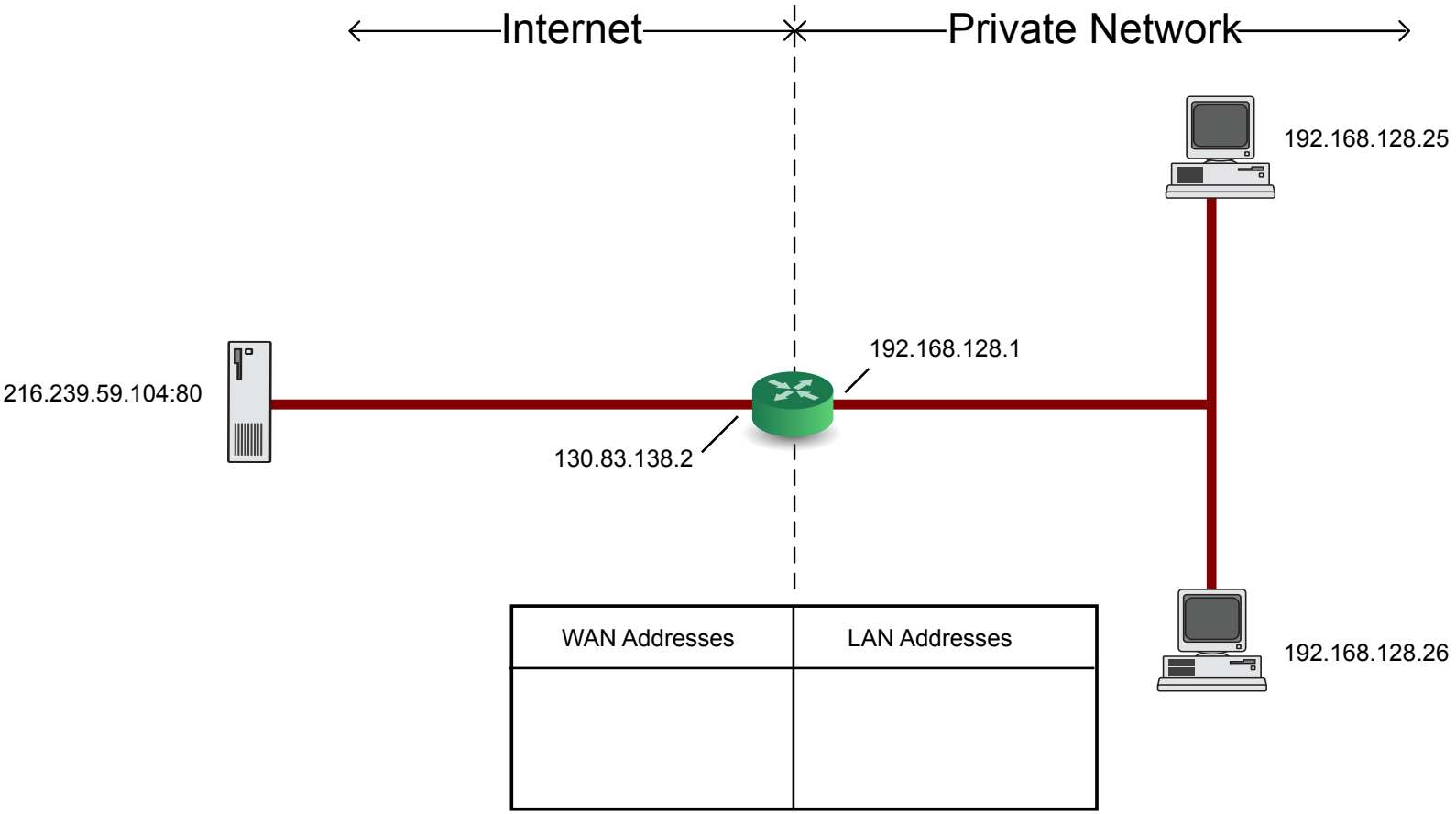
- check NAT translation table for entries that match the destination IP address and the destination port field
- exchange the destination IP address and port for the original IP address and port from the NAT translation table
- send modified datagram into the local network

NAT Example

Two hosts with the private IP numbers 192.168.128.25 and 192.168.128.26 want to access 216.239.59.104:80 from behind a NAT router. Outline what the address translation process could look like. (The NAT router has the external address 130.83.138.2.)

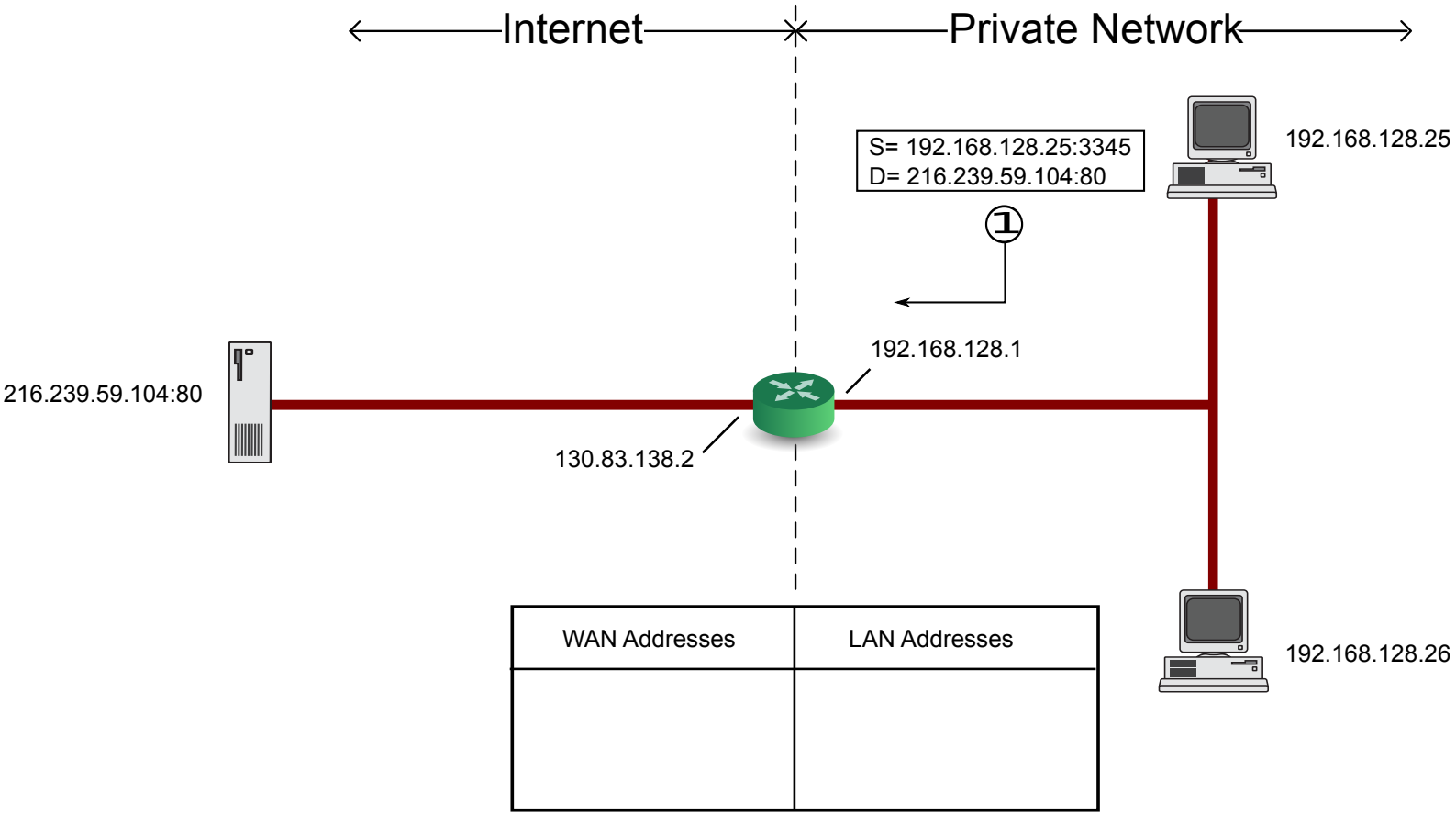
NAT Example

1. What are the source and destination addresses of a message sent by the host 192.168.128.25 to 216.239.59.104:80?



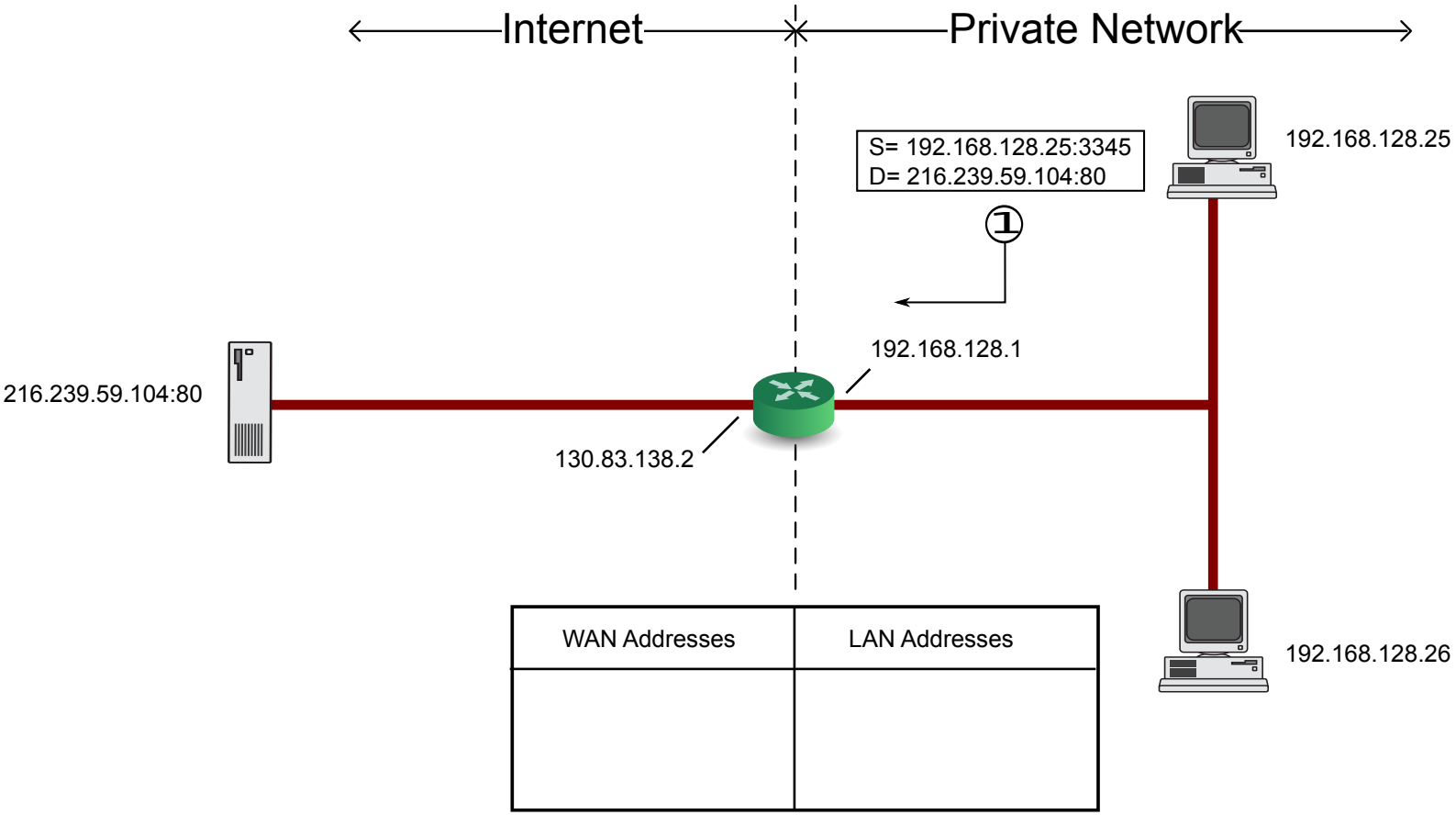
NAT Example

1. The source is the local IP of the host and a random port, the destination is the address and port of the server.



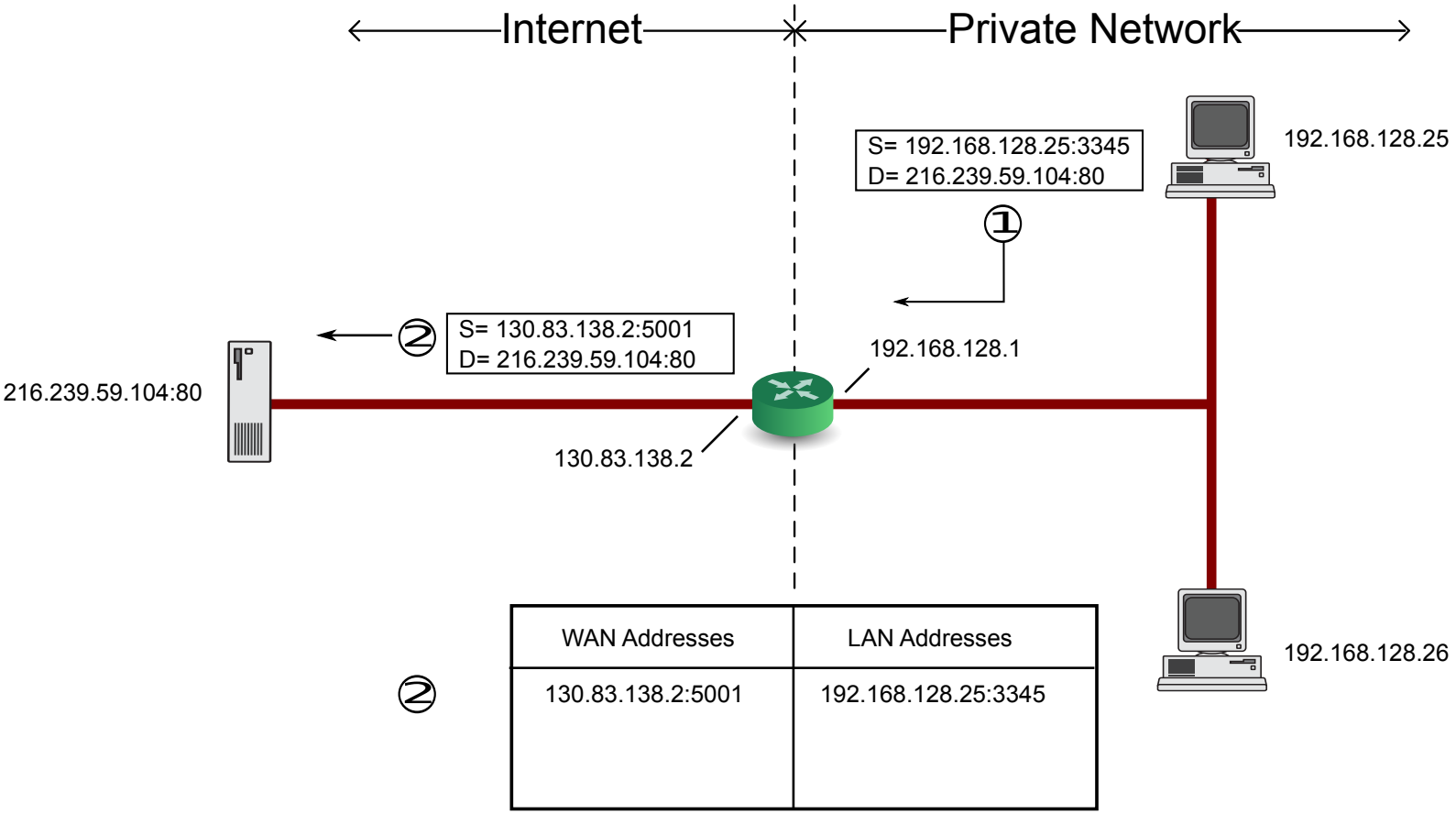
NAT Example

2. What are the source and destination addresses of the message when it exits the router?



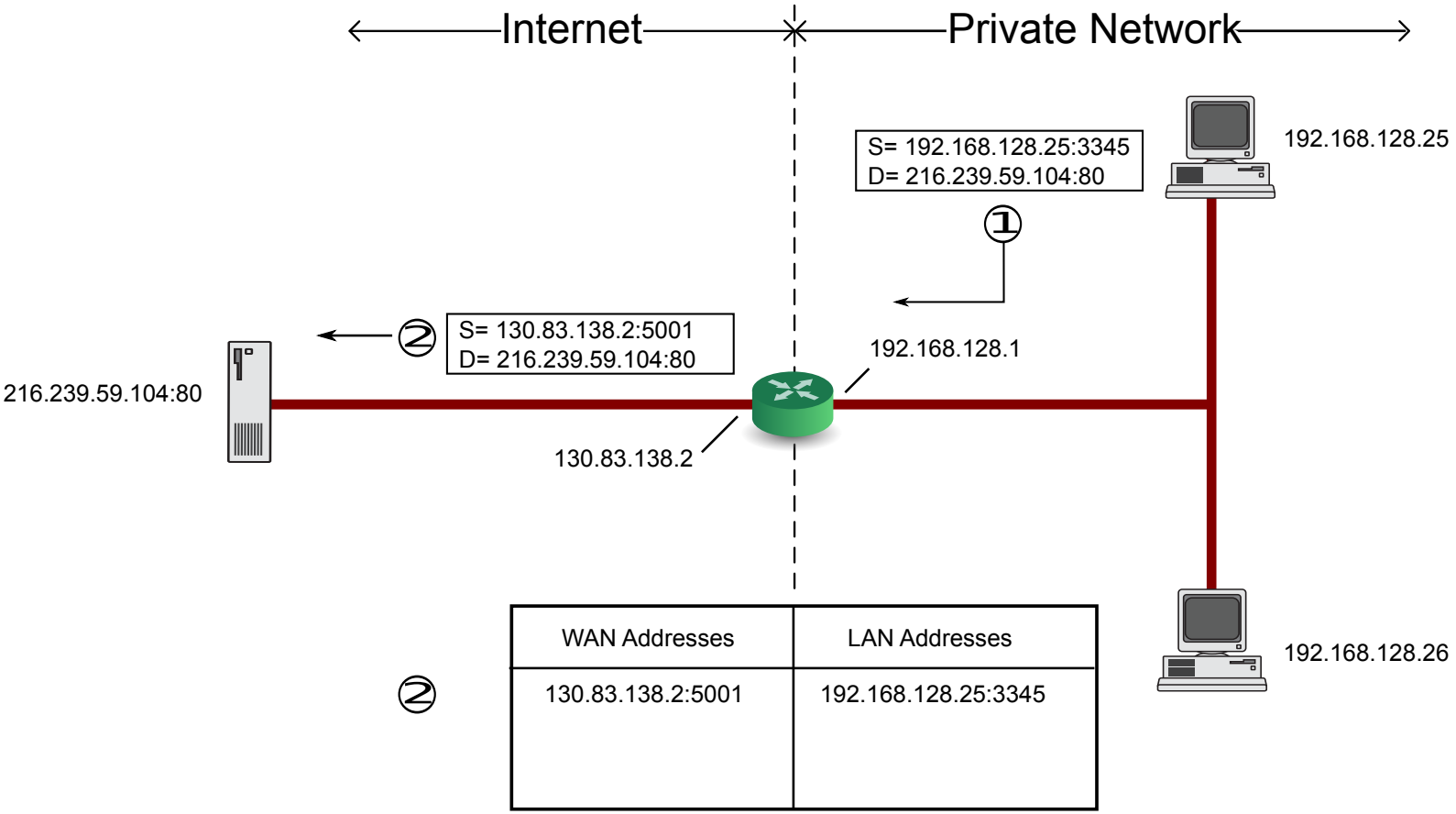
NAT Example

2. The NAT router replaces the sender IP with its own external IP and uses a random available port number. The mapping is saved in the NAT table.



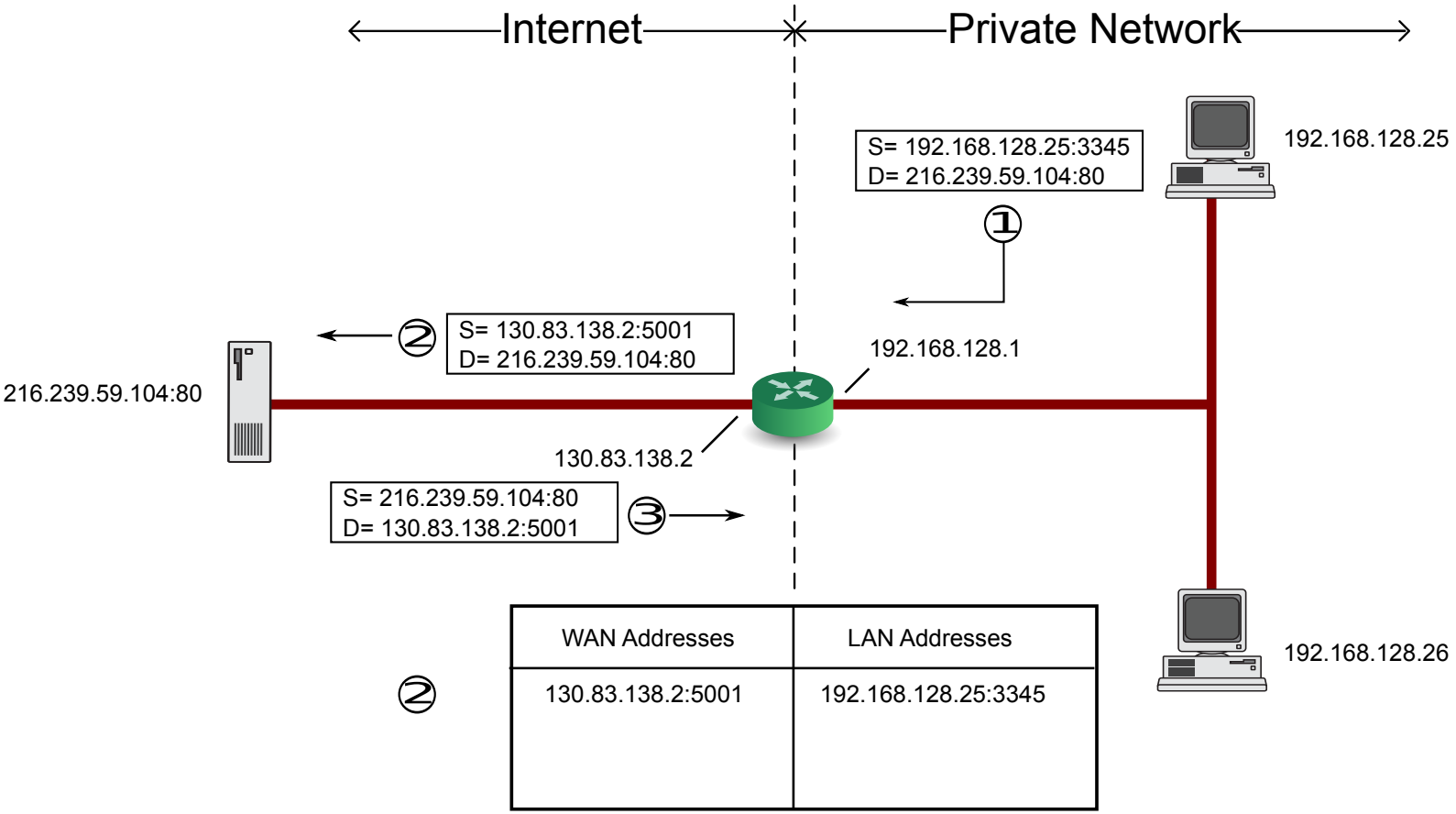
NAT Example

3. What are the source and destination addresses of the server response?



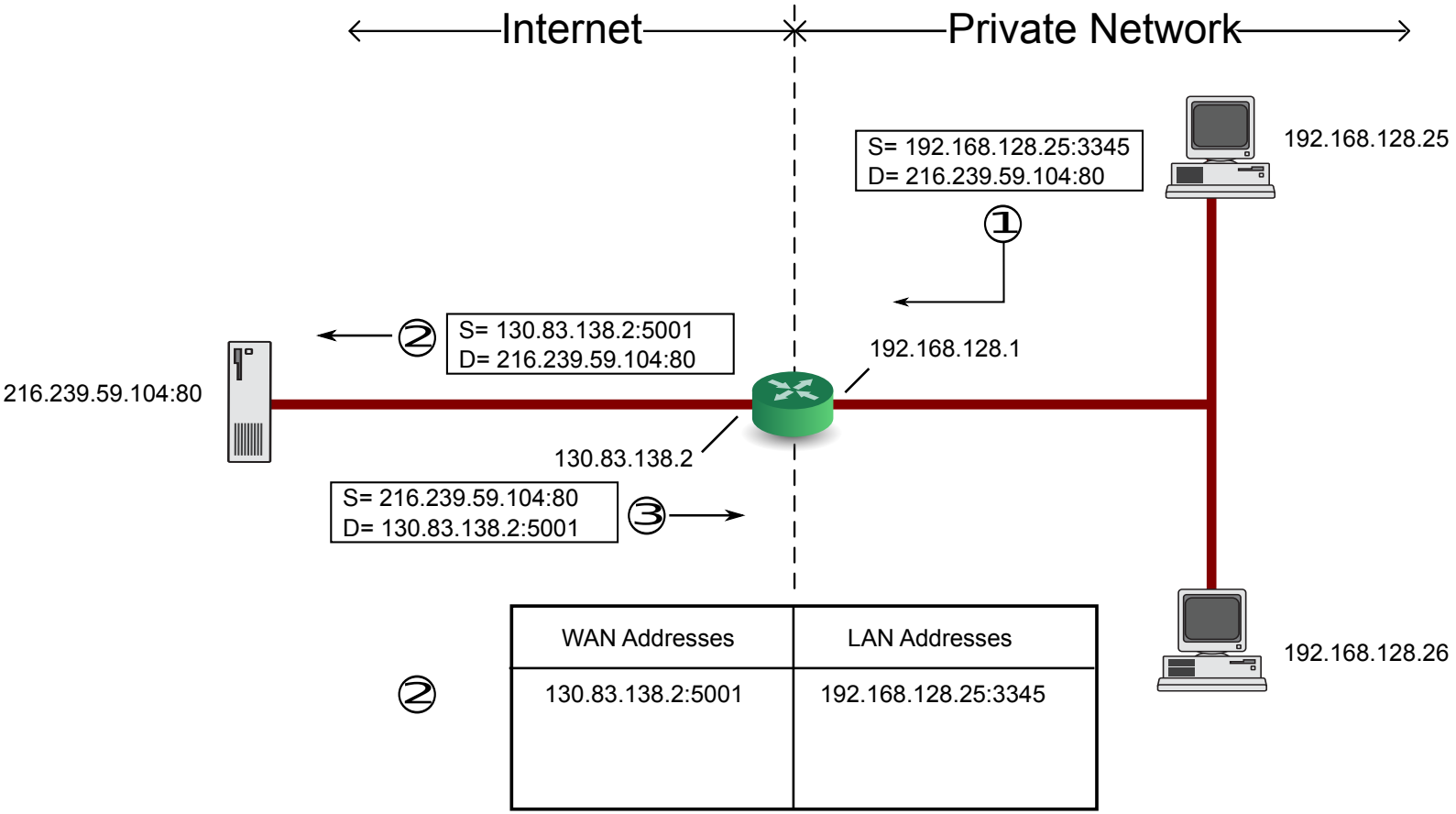
NAT Example

3. The server has no knowledge of the NAT. It just sends the message back to the external IP of the router.



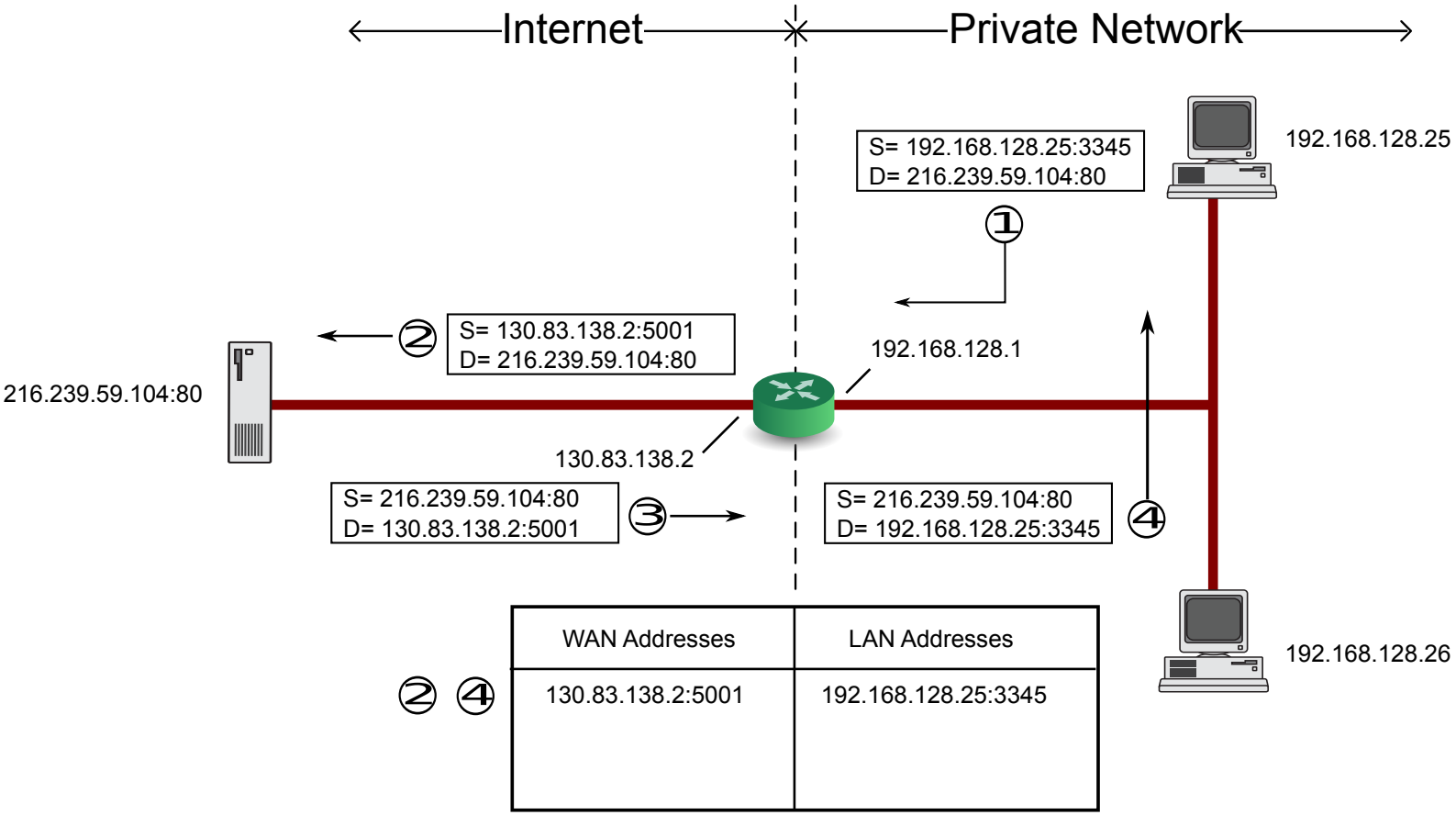
NAT Example

4. What are the source and destination addresses of the message when it goes from the router into the local network?



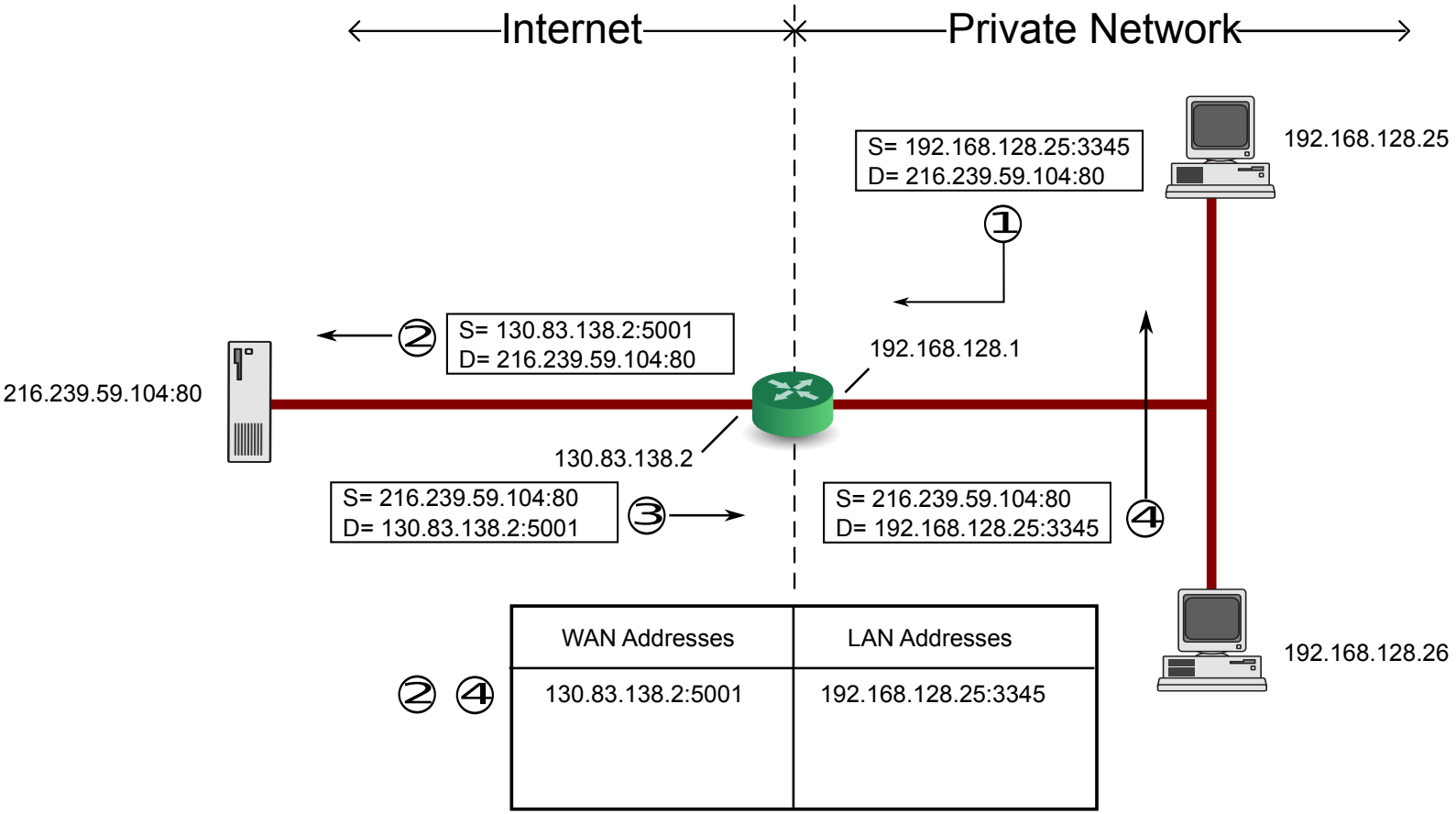
NAT Example

4. The router replaces the destination IP and port with a local IP and port according to the NAT table entry.



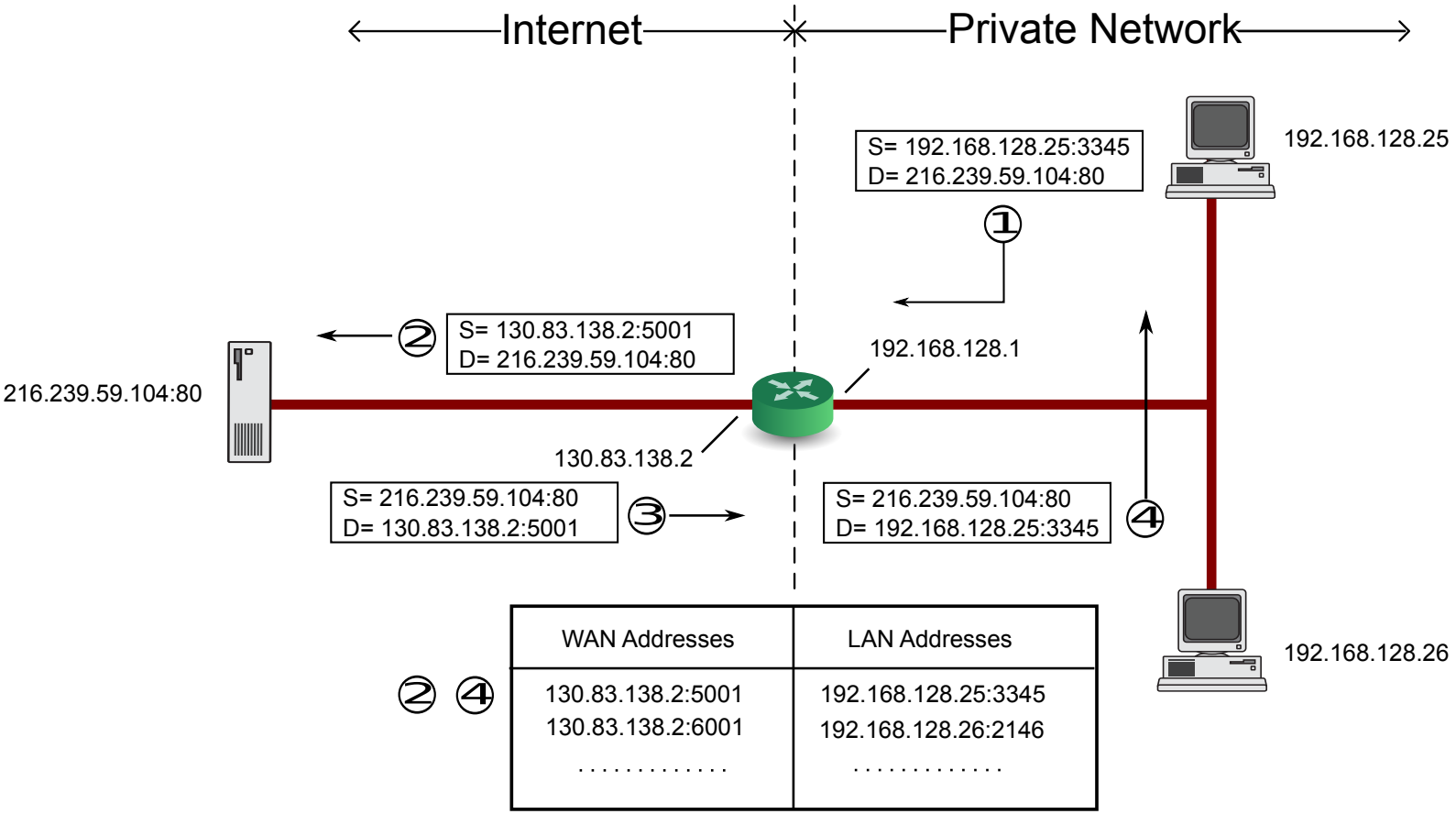
NAT Example

5. What could a NAT table entry look like when the other host wants to communicate with the same server?



NAT Example

5. The router has to assign the other host a different available port.



Webserver behind NAT

- In order to reach a server behind a NAT router, so-called port forwarding can be enabled.
- This feature allows a static mapping of an external address/port pair to an internal host.

NAT and the IP layer model

NAT violates a number of networking principles and best-current Internet practices

- port numbers are originally meant to address application layer processes and not hosts.
- NAT violates layering mixing network and transport layer functionality.
- Routers are supposed to work only on layers one up to three but not on the transport layer.
- NAT violates the End-to-End Principle.
- NAT does not address the fundamental problem of IPv4 address shortage. (IPv6 does).
- NAT complicates tunneling protocols such as IPsec because NAT modifies values in the headers which interfere with the integrity checks done by IPsec and other tunneling protocols.