Politecnico di Torino

Exercises on Packet Sniffing and Traffic Analysis

Fulvio Risso

September 17, 2017

License

Acknowledgments

The author would like to thank all the persons that contributed to those exercises. Particularly, special thanks go to Flavio Marinone and Guido Marchetto.

Contents

Part I.

Introduction

1. Methodology

Exercises related to traffic analysis focus on the prediction of the network frames that can be generated over a specific network.
In general, these exercised can be solved if the student has a reasonable knowledge about:

- The most common network protocols (summarized below), i.e. the student must be aware of the packet/frames generated by each protocol and the possible interactions between one protocol and another.
- The IP protocol, particularly with respect to routing and the Longest Prefix Matching algorithm.

In this case, the student can use the the proposed methodology in order to write network frames. Finally, the student has to take into account that, due to the possible different behavior of the network (i.e. shared/switched network), only a portion of those frames can be seen in the viewpoint proposed by the exercise.

Please note that the following information must be intended as a brief overview of the topics required to solve these exercises; some details (that the student is expected to know anyway) are omitted for the sake of brevity.

1.1. Common network protocols

The most common network protocols that will be used in our exercises are the following.

1.1.1. Address Resolution Protocol (ARP)

This protocol provides a mapping between an IP address and the MAC address of the Network Interface Card (NIC) that is associated to that IP, which must be connected to the current LAN.

In case an host knows the IP address it wants to reach on the current LAN (obviously, the given IP address must be reachable at the IP level), the ARP protocol enters into play. If the ARP cache already contains a mapping from the IP address and the corresponding MAC address of its NIC card, a L2 frame with that MAC address as destination is generated. Vice versa, if the MAC address is unknown, an *ARP Request* message is sent in broadcast, so that all the hosts on the LAN will receive that frame. All the hosts will process that frame, and only the host that recognizes its IP address inside the ARP request will reply with an *ARP Reply* message containing its MAC address, directed (in unicast) to the requester.

At the end of the process, both hosts will insert the mapping (*IP address - MAC address*) in their ARP cache. This entry will be valid until a specific timeout, whose value depends on the operating system. The requester host will insert the mapping related to the IP destination address, while the requested host will insert the data related to the requester.
The ARP cache will be refreshed (i.e. the current age of that entry will be reset to zero) each time an IP packet coming with that source IP address *and* that MAC address is received.

Please beware that a router whose IP address is IP(R), which deliver traffic generated by a remote source to a L2-attached host does not refresh the ARP cache of the host. In fact, that traffic does not come with the right IP source address. Vice versa, any IP packet sent by the host and whose first hop is the router under examination, will refresh the ARP cache of the router.

It is worthy noting that the ARP protocol can discover only MAC addresses that are reachable on the current LAN. This can be used either by a device that has to forward an IP packet to its final destination, or by a device that has to send the packet to an intermediate router on its journey to the destination (which is still far away). The choice between these two possibilities is delegated to the *Longest Prefix Matching* algorithm, which is part of the IP protocol.

A final note on the ARP protocol concerns the capability to cache ARP requests by all host connected to the current LAN. For instance, the ARP Request packet is sent in broadcast and hence is received by all the hosts on the current LAN, which therefore have the possibility to insert the mapping `IP source - MAC source` in their cache. While technically feasible, the ARP RFC does not say anything about this and all the recent operating systems do not cache that entry unless they are the target of the ARP Request. The rational behind this choice is that is useless to cache an IP-MAC address if we do not have to use that entry. Since many hosts can be on the current LAN but our host is probably not interested in taking to all of them, most of the given IP-MAC entries turns to be useless. Useless entries will cause an unnecessary overhead when the cache has to be managed (purging dead entries, refreshing existing entries), in addition to increase the search time when an entry has to be located in the cache, which is required each time a new IP packet is sent by the host.

1.1.2. Internet Control Message Protocol (ICMP)

This protocol takes care of some "service" functions required in the IP world. While we do not care here of all the functions available on this protocol, we may use the following ICMP packets:

- **ICMP Echo Request**: it is a "test" IP packet that can be sent to another host. Upon receival, the contacted host has to reply with a similar packet (the *ICMP Echo Reply*), directed to the original sender. When an ICMP Echo Request is sent, the sender starts a timer that waits for the answer: if the ICMP Echo Reply comes in time (usually the timeout is on the order of 2 seconds), it means that the two hosts are reachable at the IP level. In fact, the ICMP Echo Request/Reply couple is used to test the availability of paths at the IP level and it used by the `ping` program.
- **ICMP Echo Reply**: it is the companion of the ICMP Echo Request packet.
- **ICMP Redirect**: when a router R1 receives an IP packet whose best path toward the destination requires to forward that packet on another IP device (R2) reachable on the same IP network on which the original packet has been received, an ICMP Redirect packet is sent to the sender S0[1]. Upon receival, the sender S0 will known that a better next hop exist for that destination; next packets directed to that destination will no longer sent to router R1, but they will be sent to router R2.
- **ICMP Time Exceeded**: it is an ICMP packet generated by a router that has to forward an IP datagram to a next hop, but whose Time-to-live field is now equal to 1. This ICMP packet is sent to the original source of the IP datagram to inform it that one of its packets was lost in transit.

[1]Please note that the *source* is usually intended as the host which generates the packet, while the *sender* is usually intended ad the host which forwards the packet on. Source and sender can coincide at the first hop.

- **ICMP Destination Unreachable**: it is generated by an IP device (typically a router) when it is not able to deliver a packet to the destination. This message is sent to the source of the original IP packet in order to inform it that one of its packets had to be discarded. The reason of the unreachability is contained in the ICMP message, and ranges from network unreachable, host unreachable, port unreachable, datagram too big, etc.

1.1.3. Domain Name System (DNS)

It provides a way to map a literal name (e.g., `www.mydomain.com`) to an IP address (e.g., `130.192.3.21`). While not strictly required for the Internet, it represents a fundamental piece of software that allows humans to have a more friendly interaction with IP addresses.

The basic interaction with the DNS is carried out by two packets, encapsulated in UDP/IP:

- **DNS Query**: the host that wants to deliver a packet to an host known only by name, it sends a DNS Query to its DNS server (the address of the DNS server is usually statically configured in the host), asking the resolution of the given name.
- **DNS Answer**: upon receiving a DNS Query, a DNS server takes care of "translating" the literal name into the corresponding IP address, and returns it to the requester through a direct DNS Answer packet.

Although the functions (and the possible traffic generated by a DNS server) are definitely complex, in the most common cases those two messages are enough to cover the vast majority of the traffic generated by the DNS, at least as perceived by normal users.

1.1.4. Transmission Control Protocol (TCP)

This transport-level protocol is in charge of handling resilient connections over the Internet, possibly recovering IP datagrams lost in the network. The overhead of this protocol results in (usually) 20 bytes added after the IP header (and before the application-layer message), plus some additional packets generated to handle the connection.

Among the additional packets, we have:

- **Three-way handshake**: three TCP packets are generated in order to establish the connection between H1 and H2 (the *TCP SYN* from H1 $\rightarrow$ H2, the *TCP SYN-ACK* from H2 $\rightarrow$ H1, and the *TCP ACK* from H1 $\rightarrow$ H2). After the handshake, the application-layer messages can be transported over the established connection.
- **TCP ACK packets**: although the behavior varies according to different conditions, usually each TCP packet transporting application data from H1 $\rightarrow$ H2 is followed by a void TCP ACK packet, sent from H2 $\rightarrow$ H1 (and vice versa).
- **Modified three-way handshake**: it encompasses four TCP packets that are exchanged between two hosts in order to close an established connection.

1.1.5. User Datagram Protocol (UDP)

It is the transport-level protocol used when resiliency is not required and hence TCP is an overkill. Basically, it adds a small header of 8 bytes between the IP header and he application-layer message (e.g., the DNS header). It does not include any additional message (such as in TCP).

1.1.6. Hyper-Text Transfer Protocol (HTTP)

HTTP was born to transport web data, but it is now used by several different applications. It is based on a client/server paradigm, where the client can use several types of request message.

Most common is the GET message, which is followed by the appropriate answer coming from the server. Answers use numeric a code (e.g., 1xx, 2xx, 3xx, etc.) in order to notify the client if the request was successful or some errors occurred.

Usually, the request fits into a single IP packet (and hence TCP segment), while answers can span across multiple TCP segments.

1.2. The IP Longest Prefix Match algorithm

The *Longest Prefix Match* algorithm must be used to determine if the current IP packet can be sent directly to the final destination, or it has to be relayed through a router. A direct transmission implies that sender and receiver are physically on the same network, and hence some existing L2 mechanism (outside the IP protocol) exist that transports the packet to the local destination.

The longest prefix match checks if the the host that has to send the packet (that may not be the original source of the IP datagram) and the final destination are in the same IP network. In that case, the direct delivery available at L2 will be used. Otherwise, the IP routing will select the proper next hop (e.g., an IP router) that represents the next step toward the destination.

Obviously, the next hop must be directly reachable: in other words, if we apply the longest Prefix Match algorithm between the current sender and the host selected as next hop, the algorithm must return a positive answer (i.e., they both belong to the same IP network). Otherwise, the sender can no longer send the packet, which is then discarded.

1.3. Methodology for network frames generation

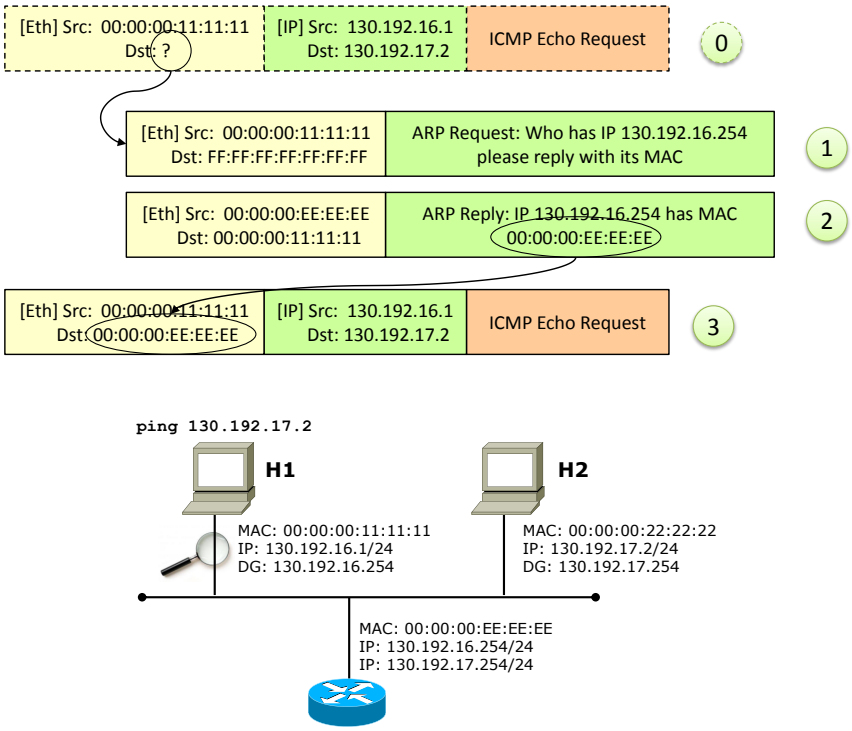
Exercises related to traffic analysis can be solved by taking into account that, often, the application-layer protocol that is require to generate the packet (e.g. a PING), cannot send that frame directly on the network, because some information (e.g. IP or MAC addresses) is missing in the frame. For instance, if an host want to ping a server "foo", the name has to be first converted into an IP address through a couple of packets DNS Query / DNS Answer. In the same way, if an host wants to send an Ethernet frame to another host known only through its IP address, has to issue an ARP Request / ARP Reply in order to know the corresponding MAC address.

In order to solve these exercises, our suggestion is to start by writing the application-layer packet down (e.g. the packet marked as (0) in the picture below). The packet has to be written in its entirety, including also the IP and Ethernet layers[2]. In case some information is missing (the MAC address of the router, in the example), the appropriate protocol has to be invoked in order to generate the packets needed to retrieve the missing information. Those packets must be written using the same criteria shown before: the application layer first, and then the IP and Ethernet encapsulation, with the appropriate addresses. If all those addresses are known, the packet will be sent on the physical network, otherwise the transmission is delayed and another additional protocol has to be invoked in order to retrieve the missing information.

[2]For simplicity, we assume that hosts generate IP packets over an Ethernet network.

This recursive algorithm will terminate when the original packet is finally sent on the network, which will happens when all the other service protocols completed their steps. In the example below, the MAC address of the router is unknown in frame (0), and therefore a couple of ARP Request / Arp Reply are sent on the network (frames (1) and (2)). Once this completes, the missing MAC address is now known by host H1, and the frame (0) can be finally sent on the network, where it will appear as frame (3).

The packet trace will obviously continue, since that frame is now received by the router, but the frame has still to be delivered to the actual destination, host H2. However we omit following packets as we believe the methodology is now clear.



1.3.1. Algorithm

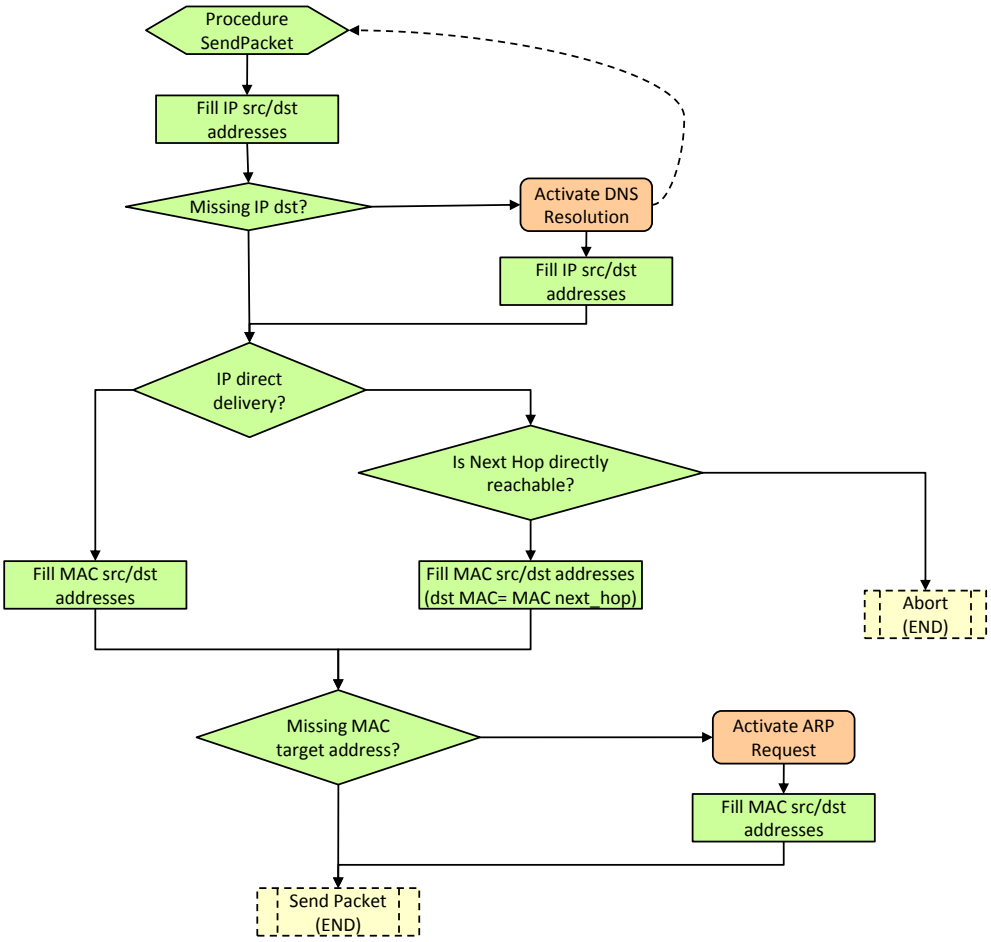
The (almost) full algorithm for generating frames is shown in the figure below[3]. The sending host generates the application-layer data and then it invokes the `SendPacket` procedure. This function fills the IP addresses first, and in case the target address is missing (because the user specified a *name* instead of a *target IP address*) it invokes the DNS resolution step. For instance, this procedure will basically invoke the SendPacket again, but with different data (hence the dotted arrow in the picture).

When the target IP address is known (e.g., the DNS resolution returned), the IP longest prefix match algorithm will determine whether the packet is entitled for direct delivery (i.e., the IP target address is on the same IP network of the current host) or not (i.e., a router has to be used to deliver the packet). In the first case, the Ethernet frame will require the MAC address of the final IP destination, while in the second case it will need the MAC address of the *next hop* router. In the second case we

[3]Please note that some details are omitted in this algorithm for the sake of clarity. For instance, we assume that the transport-layer protocol and the associated ports are known, that the *ethertype* codes for the L3 protocol are known, etc.

still have to check that the next hop is reachable, i.e., that it resides in the same IP network of the sending host. If this condition is no true, the sending process aborts.

At this point, we may have the case in which the target MAC address (either the destination host or the next hop router) is unknown. However, we have the corresponding IP address and therefore we can begin an ARP transaction in order to retrieve that data. At this point all the data is known and the packet can finally be delivered on the network.



1.3.2. Caveat

When writing the network packets generated by a given application (e.g., ping, http, etc.), a gold rule to remember is that the packet flow usually creates a connected graph. In other words, usually a packet that travels A $\rightarrow$ B is followed by a packet that travels B $\rightarrow$ A or, in case some additional resolution is required (e.g., DNS resolution), this may be followed by a packet that travels B $\rightarrow$ C. If we follow the packet stream with a pen we should always be able to track the entire packet flow without rising the pen from the sheet. If this does not hold, our packet flow is probably wrong.

1.4. Network behavior

In our exercises we focus on Local Area Networks. However, the underlying technology used to create the LAN can have a big impact on how the frames generated over that network are forwarded to the different endpoints. Particularly, we can differentiate the behavior of the network based on the technology used to create the network at level 1-2 according to the following options:

- **Shared network**: in a shared network (i.e., when the network infrastructure at level 1-2 includes only physical cables and/or repeaters/hubs), all the frames transmitted by one endpoint are automatically transmitted to all the endpoints of the same L2 network, even if they are not interested in that frame (e.g, because the destination MAC address in the frame does not match the MAC address of the local interface). In other words, the frames captured on that network do not depend on the position of the capturing device, since all the links/endpoints will receive the same traffic.
- **Switched network**: in a switched network (i.e., when the network infrastructure at level 2 includes bridges/switches), the frames transmitted by one endpoint are transmitted to only on the links that bring to the wanted destination (i.e., on the path that will be used to reach the endpoint whose MAC address matches the destination MAC address contained in the network frame). In this case, a frame directed to MAC(A) will never be received by a capturing device that resides on the link that connects an host B to the network; only broadcast (and, for simplicity, multicast) frames at L2 will be delivered to all the endpoints[4]. In other words, in a switched network the traffic is "filtered" by the intermediate network devices and the frames captured on that network depend on the position of the capturing device, since all the links/endpoints will not receive the same traffic.

Obviously, mixed situations in which both hubs and switches coexist in the same LAN are possible. In that case, part of the network will operate as a shared network, while the rest will operate as a switched one.

1.4.1. NICs in promiscuous mode

A capturing software operating on an host can work using the NIC either in *normal* or *promiscuous* mode. Those modes are similar to the previous concept of shared/switched network, because in the first case the hardware of the NIC will activate a function that filters all the frames not directed to the station out. In other words, if a frame is received whose MAC destination address is different from the MAC address of the station, that frame is deleted from the NIC memory and never delivered to the capturing software.
Vice versa, a NIC operating in promiscuous mode does not have this filter turned on, and all the frames delivered by the network to that NIC are received and transferred to the capturing software.

In other words, the promiscuous mode is similar to the behavior of a shared network (no filtering is done on network frames), while the normal mode is similar to the behavior of a switched network (only frames interesting for the user are delivered to the capturing software).

It is worthy noticing that usually the capturing software operates by putting the NIC in promiscuous mode.

[4]This is true if the bridge/switch works properly, i.e., its filtering database is filled with the proper information. In this set of exercises we assume that those network devices will operate in their normal working conditions, without considering the transient or the possible exceptions (e.g., filtering database full, etc.).

Part II.

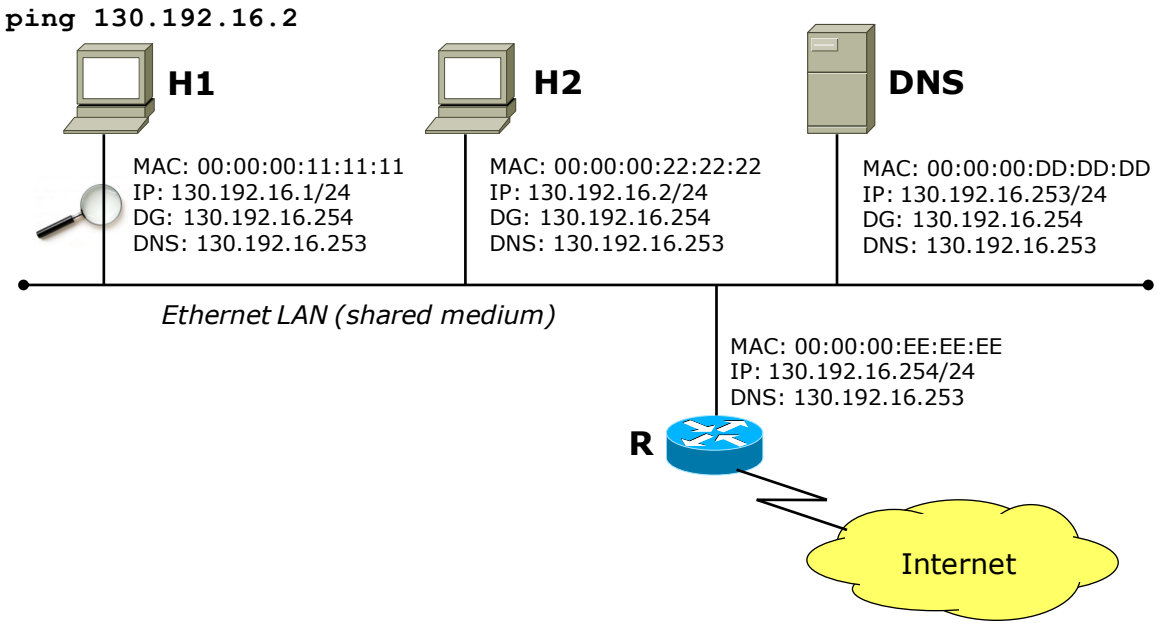
Exercises

2. IP traffic analysis

2.1. Exercise 1

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping 130.192.16.2`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

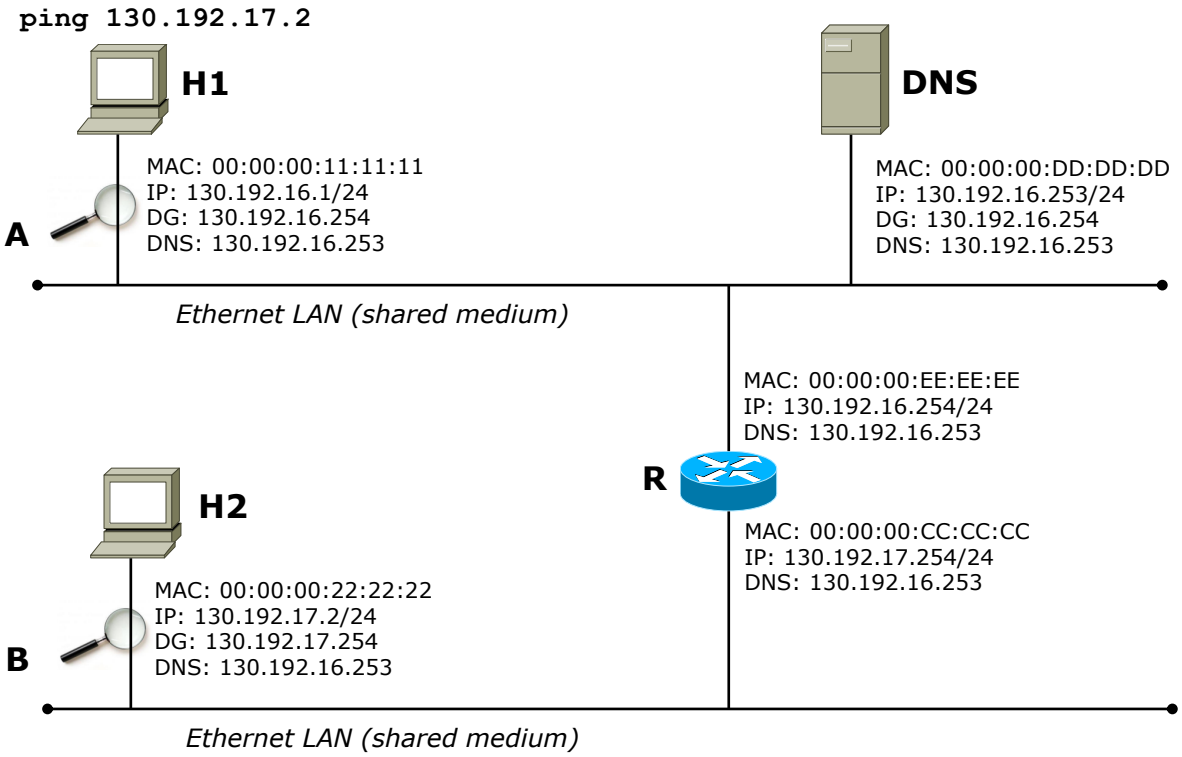
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.2. Exercise 2

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping 130.192.11.2`". Determine the number and the type of the frames captured by the two sniffers depicted in the figure, supposing that:

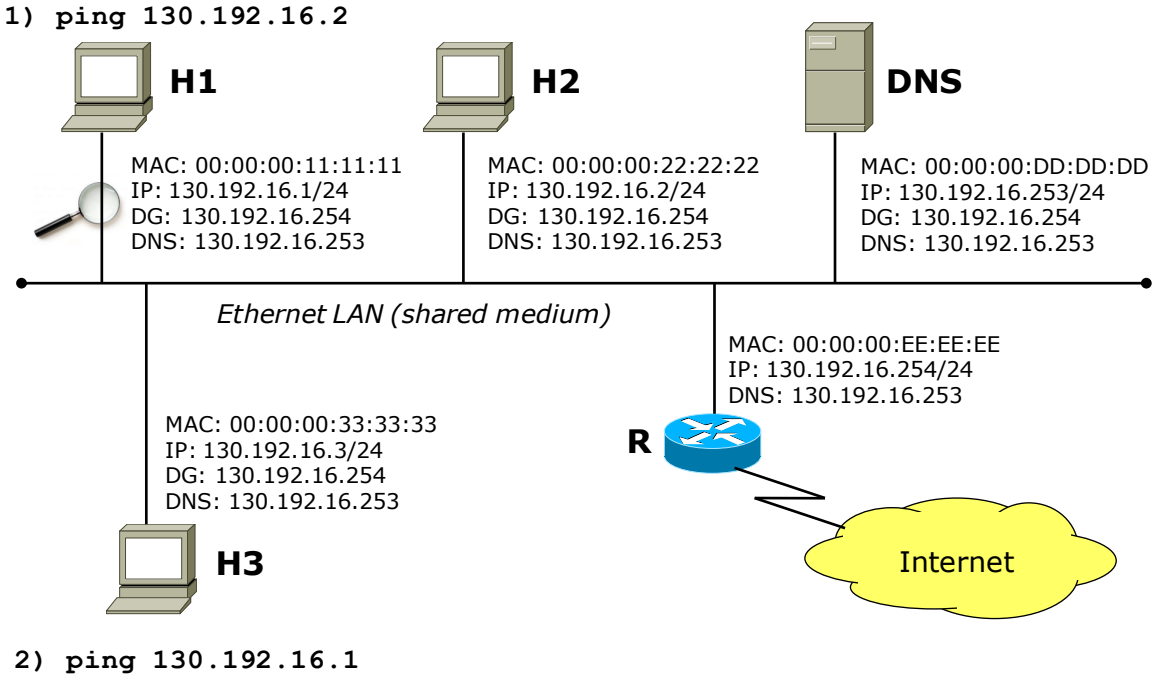
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.3. Exercise 3

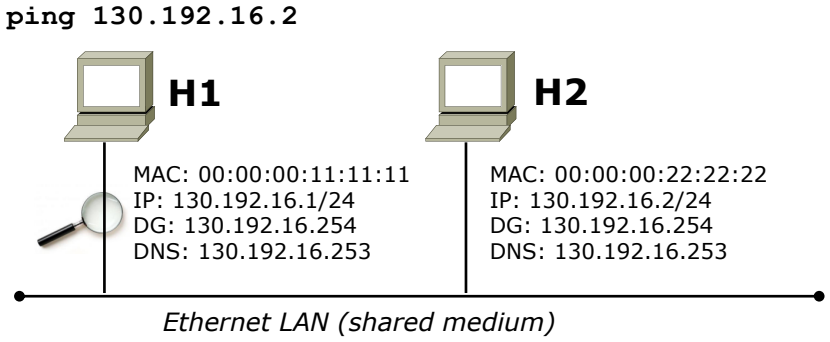
Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping 130.192.16.2`". Then, when the command completes, the owner of host H3 types the command "`ping 130.192.16.1`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.4. Exercise 4

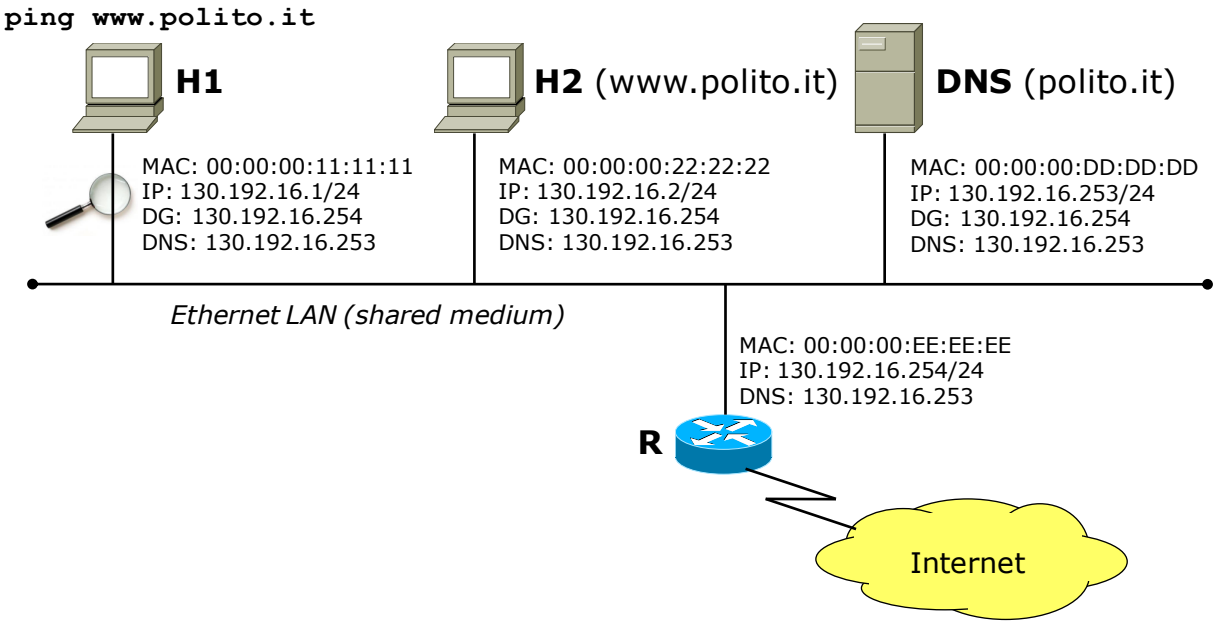
Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping -t 130.192.16.2`" (i.e. continuous ping until interrupted by the user). Determine the behavior of the network in case host H2 is disconnected from the network after some minutes. In addition, write a possible set of frames captured by a sniffer located on the cable that connects host H1 to the LAN.



2.5. Exercise 5

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

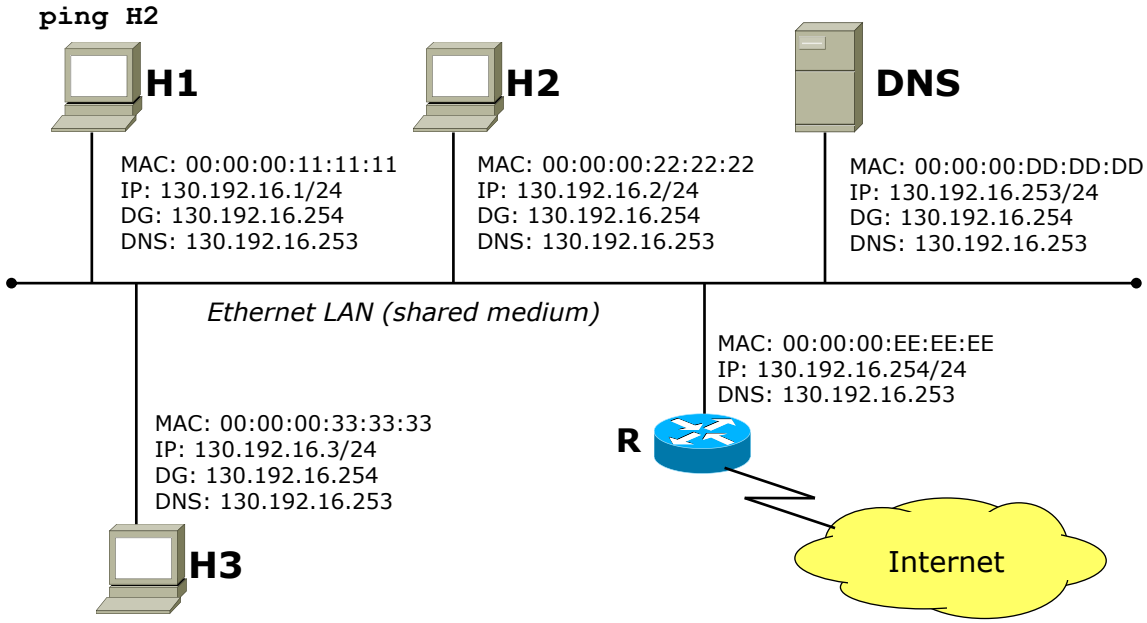
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.6. Exercise 6

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping H2`", which, on the network below, generates the frames shown in the following table. Determine:

- which frames are received by the network card of host H2
- which frames are received by the operating system of host H2 when the network card is set in promiscuous mode
- which frames are received by the operating system of host H2 when the network card is set in the standard operating mode.



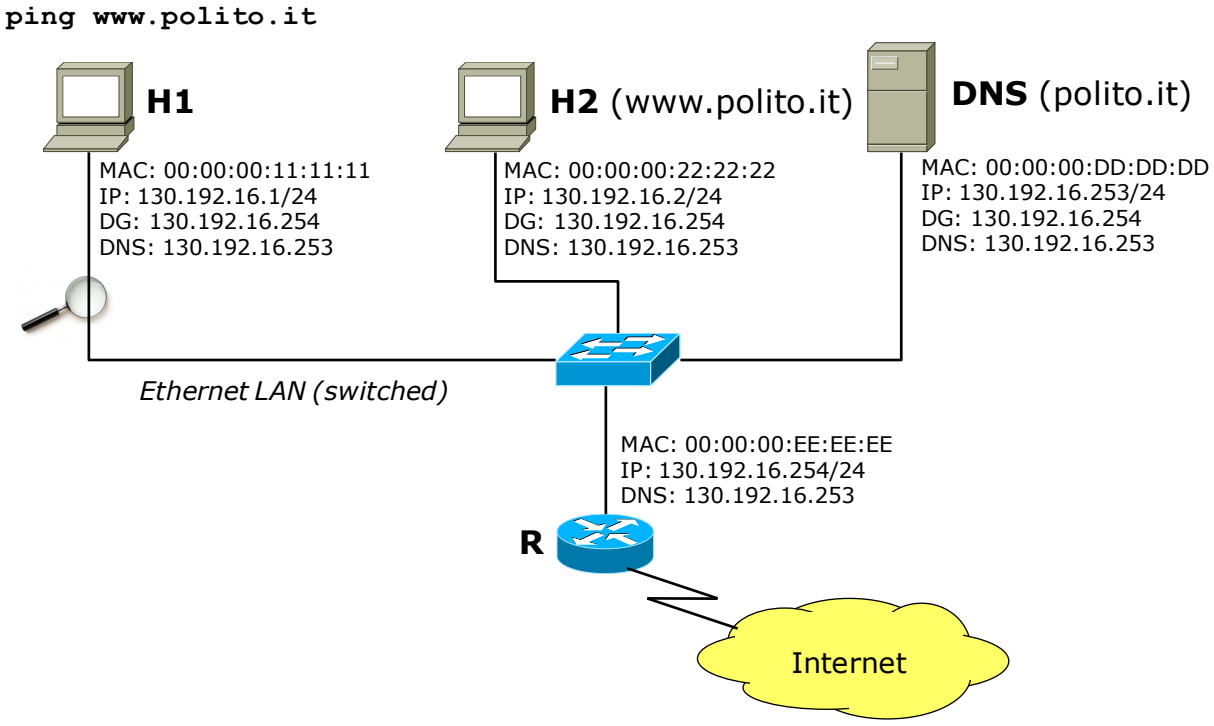
N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.253 please reply with its MAC address
2	00:00:00:DD:DD:DD → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.253 has MAC = 00:00:00:DD:DD:DD
3	00:00:00:11:11:11 → 00:00:00:DD:DD:DD	130.192.16.1 → 130.192.16.253	DNS Query	Get the IP address corresponding to name "H2"
4	00:00:00:DD:DD:DD → 00:00:00:11:11:11	130.192.16.253 → 130.192.16.1	DNS Answer	Host "H2" has IP= 130.192.16.2

5	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
6	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.2 has MAC = 00:00:00:22:22:22
7	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
8	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply

2.7. Exercise 7

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it.`" Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

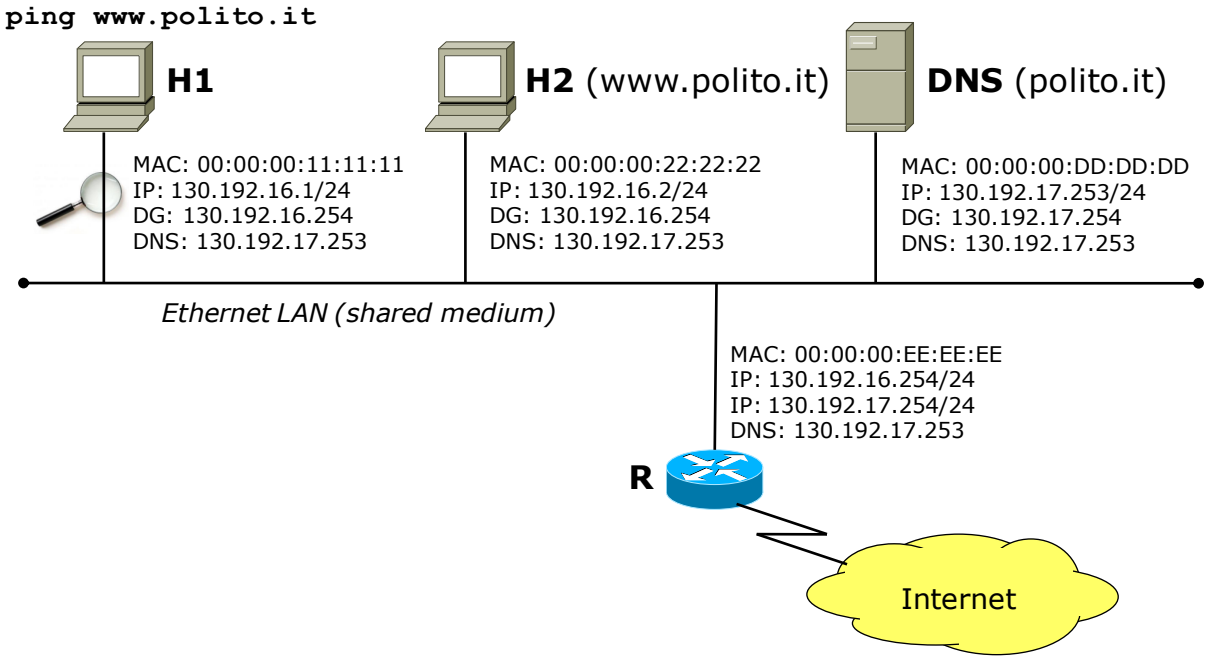
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.8. Exercise 8

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

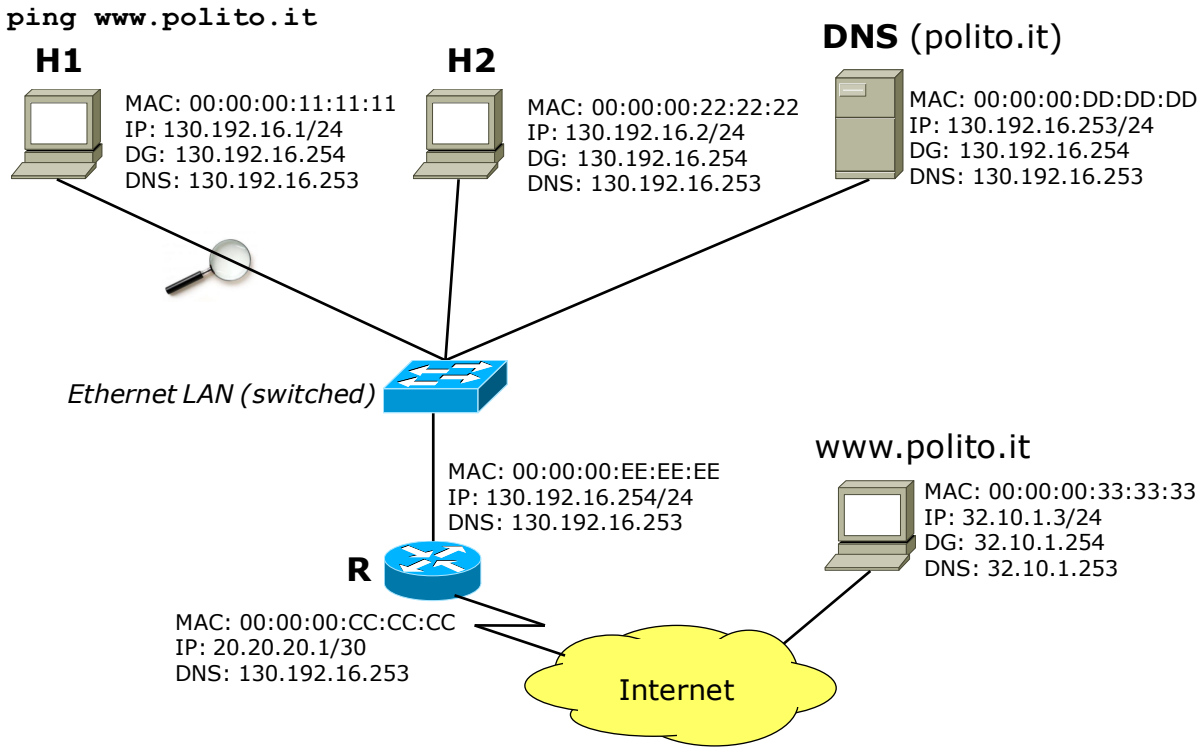
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.9. Exercise 9

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

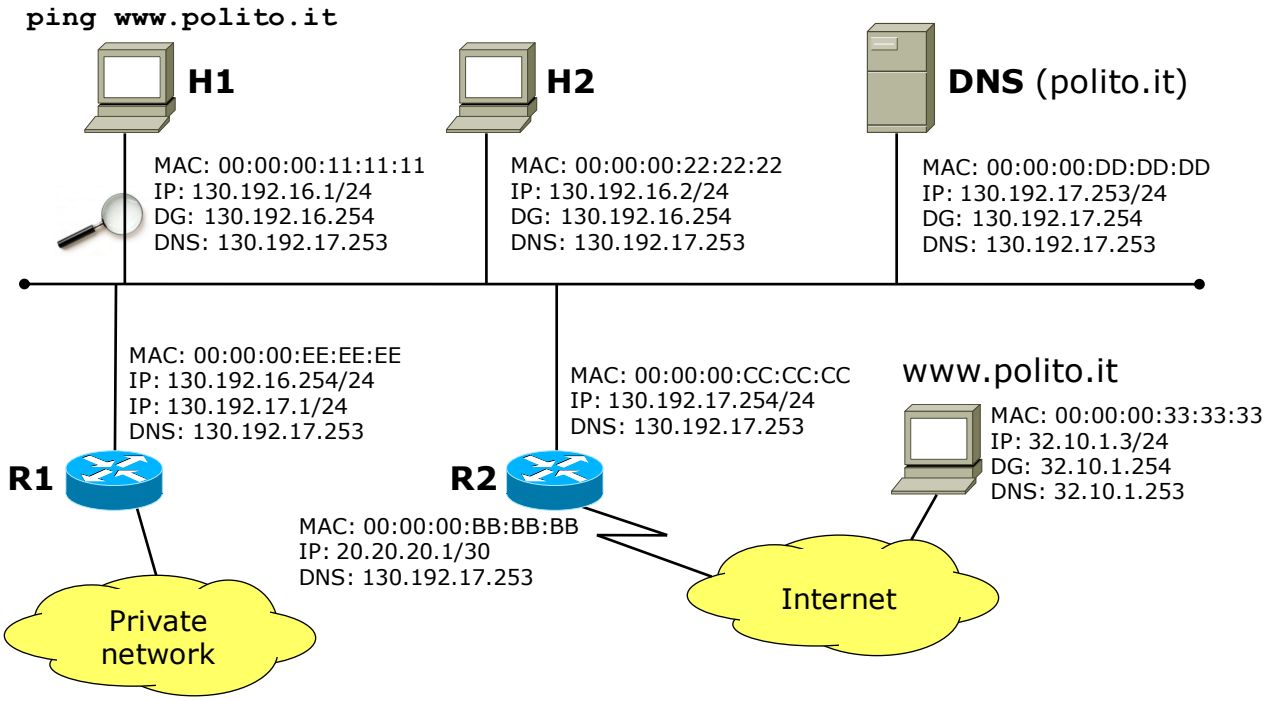
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.10. Exercise 10

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

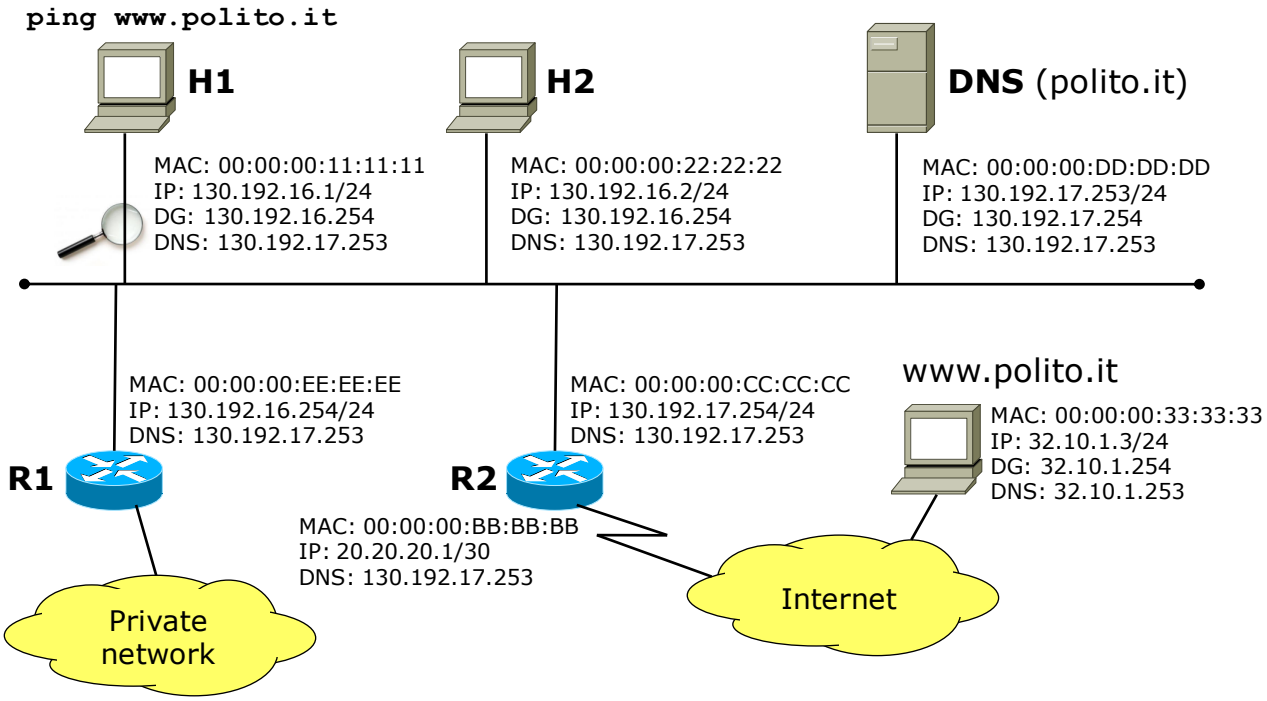
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.11. Exercise 11

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

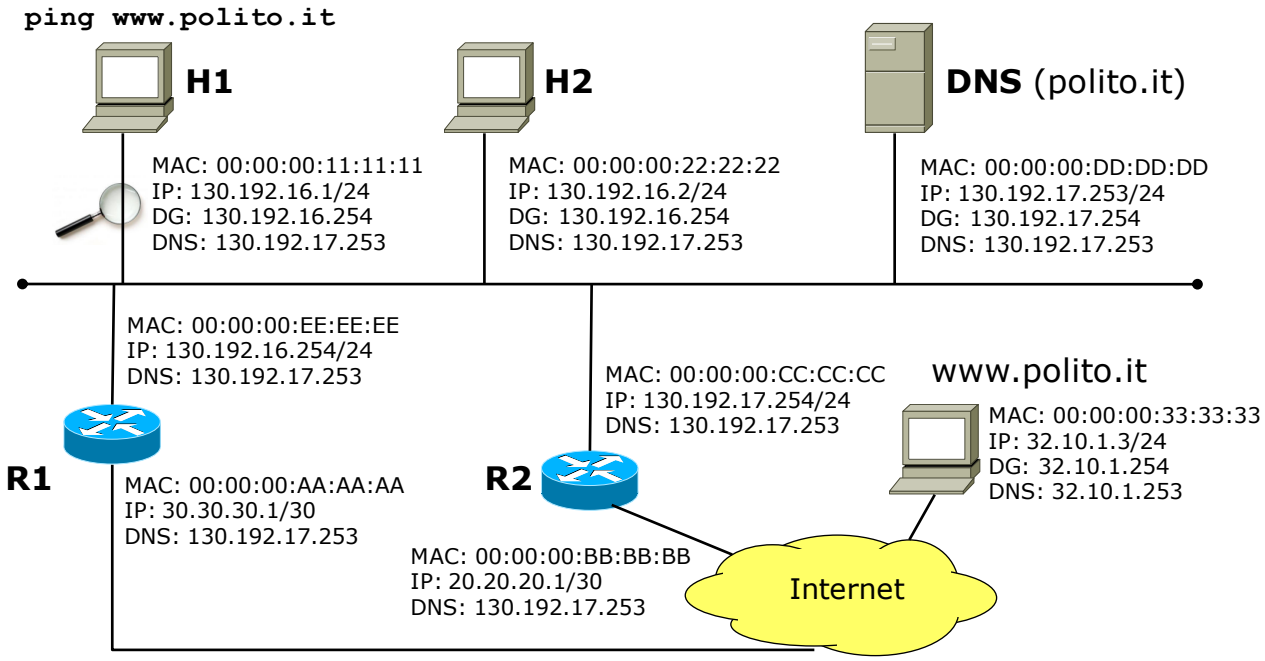
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.12. Exercise 12

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

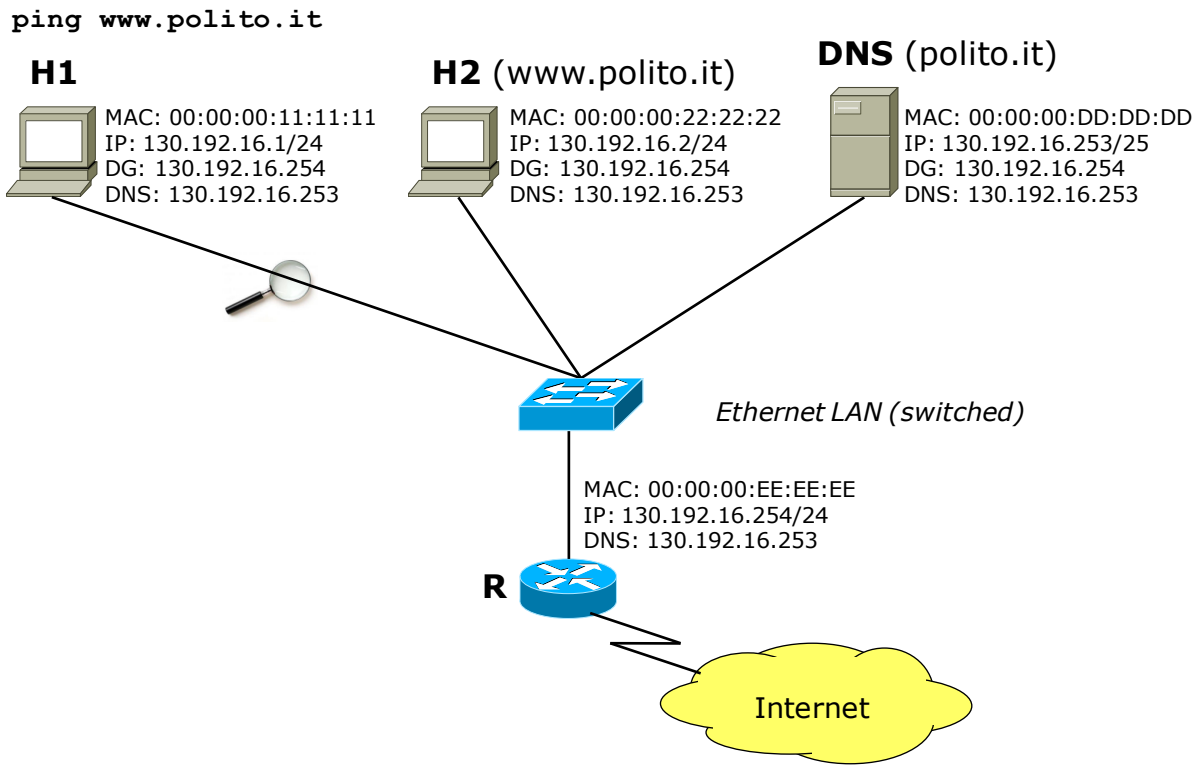
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.13. Exercise 13

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`", and that the DNS has been configuring by mistake with the wrong netmask. Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

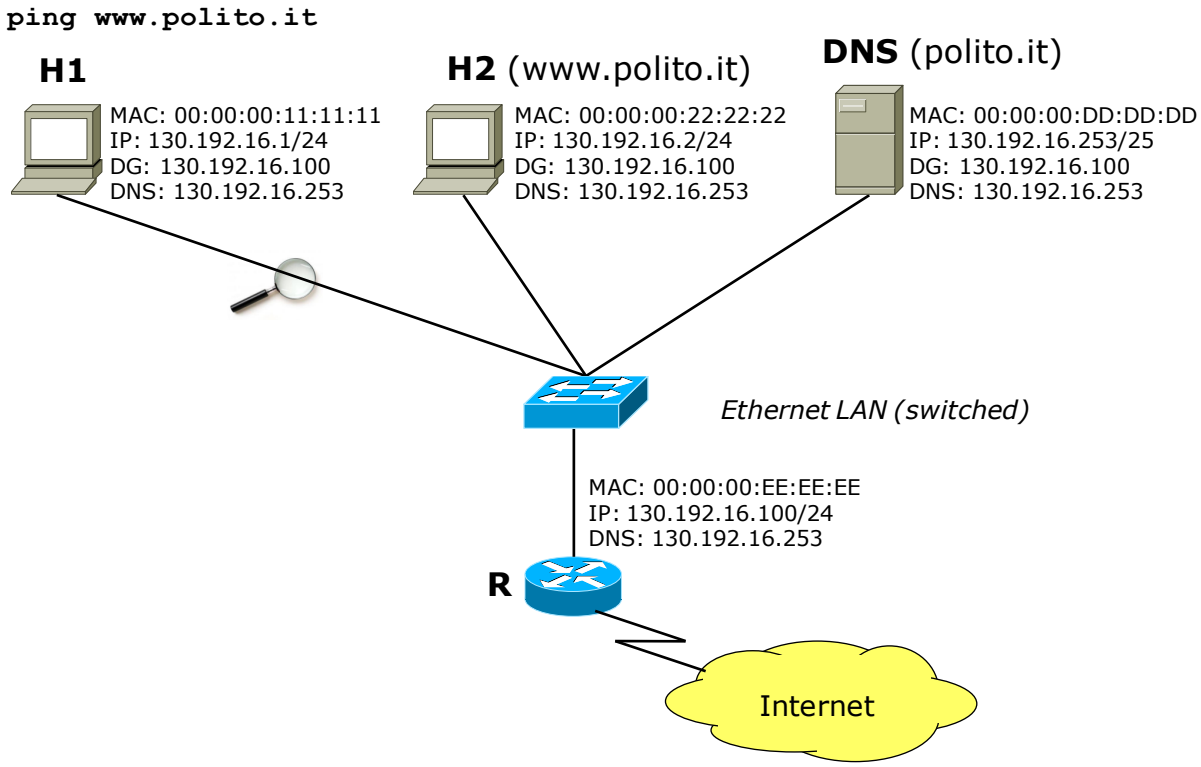
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.14. Exercise 14

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

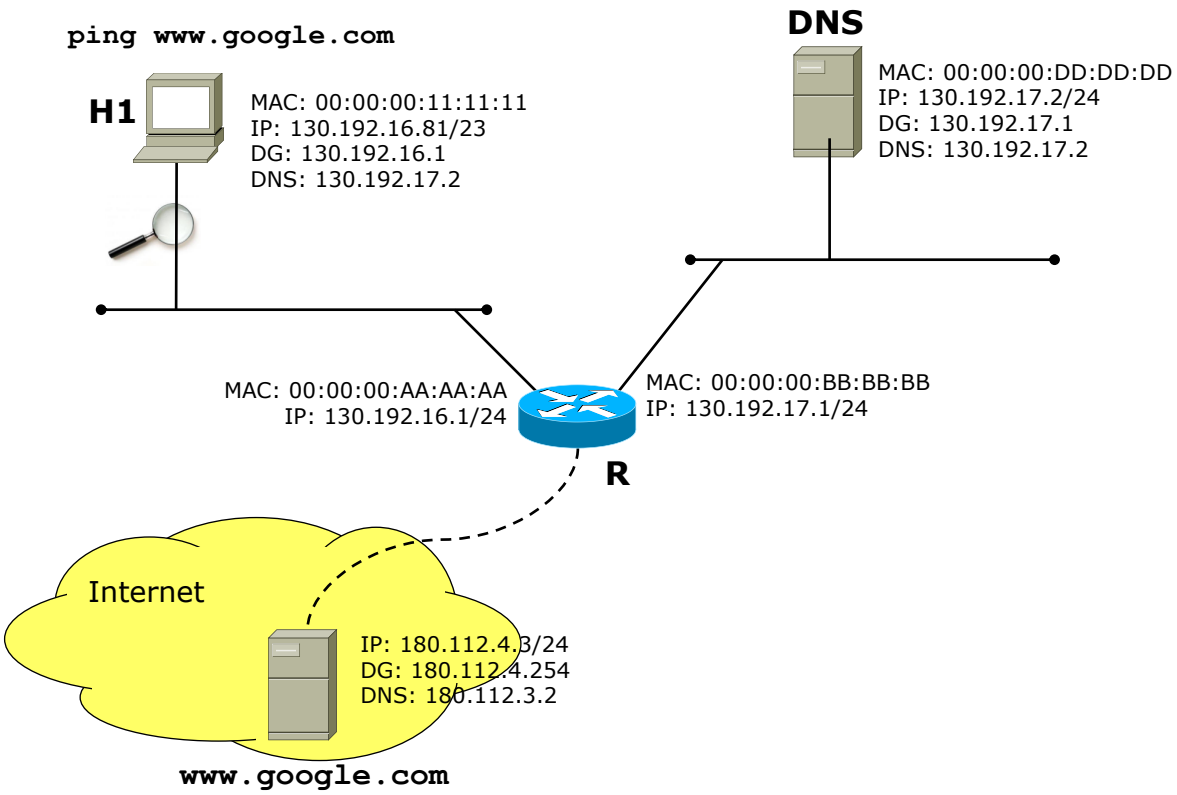
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



2.15. Exercise 15

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.google.com`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)

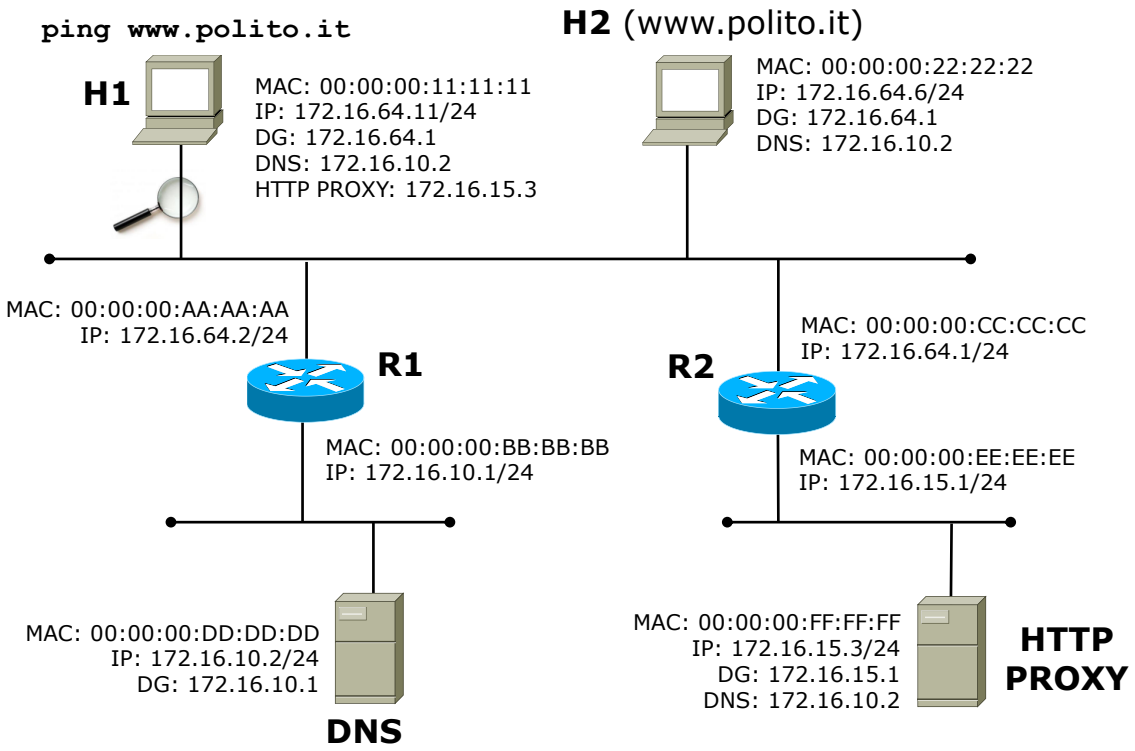


3. Application-layer traffic analysis

3.1. Exercise 16

Referring to the network topology depicted below, let us suppose that the owner of host H1 types the command "`ping www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

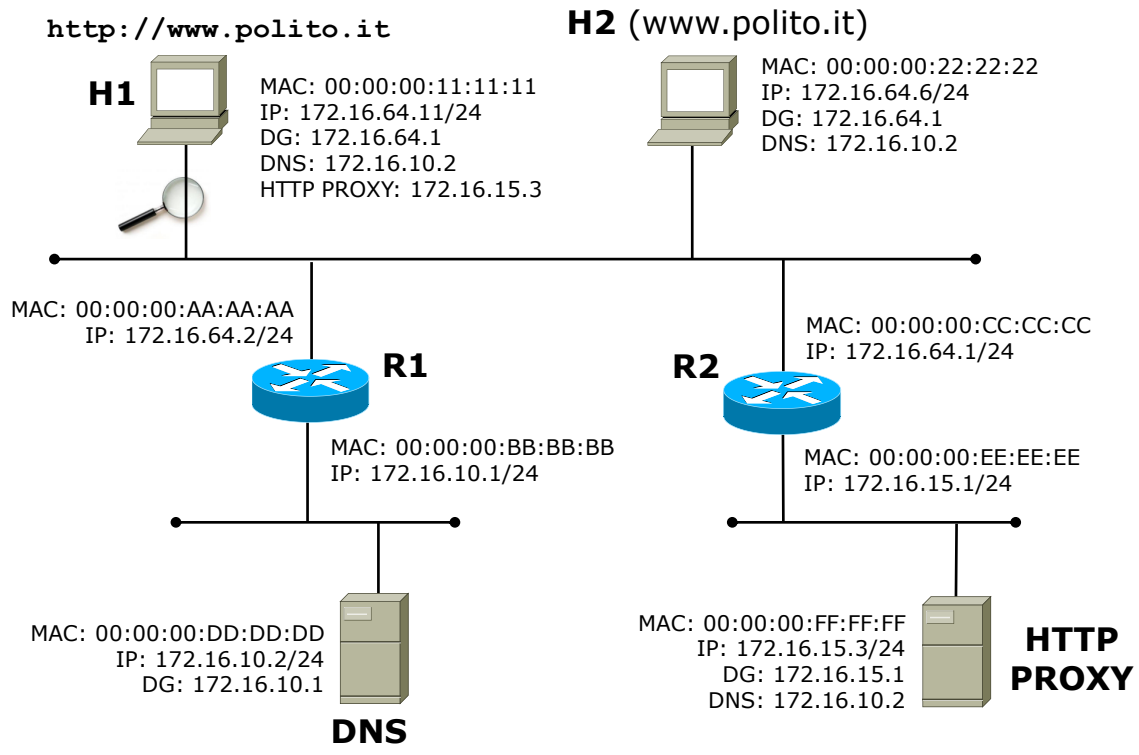
- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The cache on the HTTP proxy contains all the requested HTTP pages
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



3.2. Exercise 17

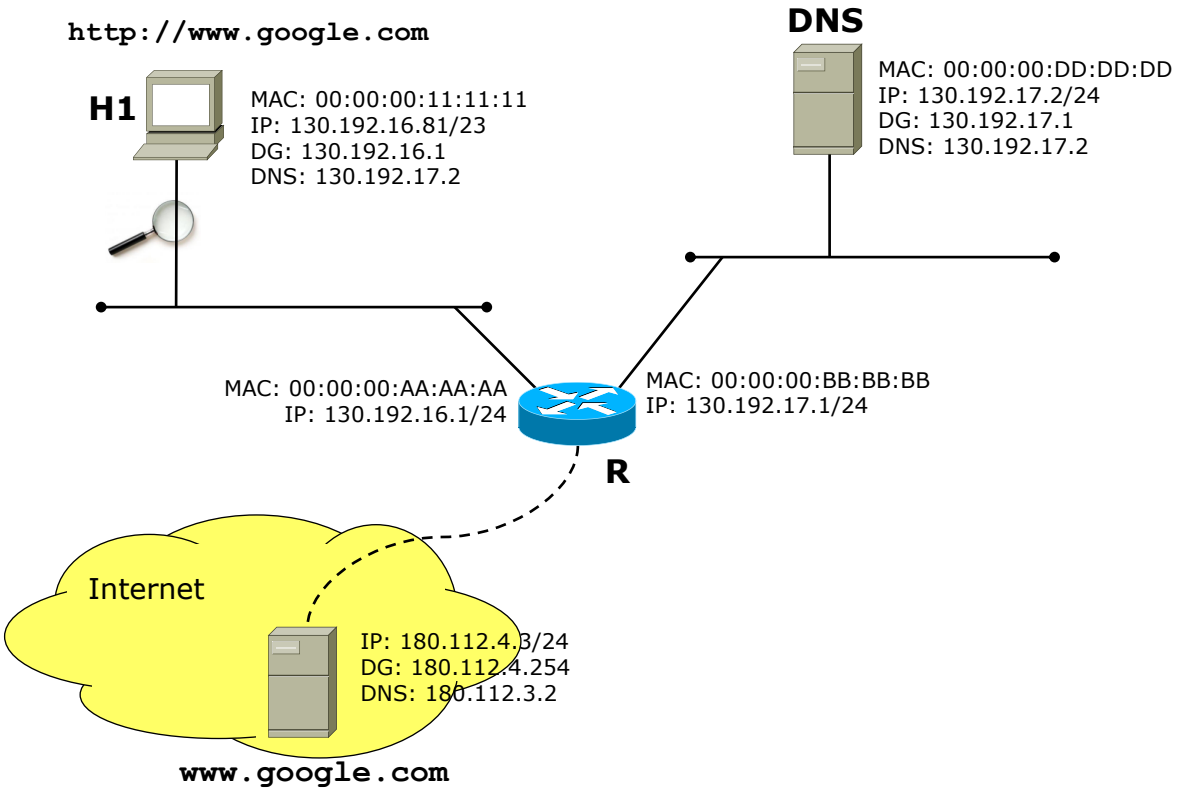
Referring to the network topology depicted below, let us suppose that the owner of host H1 opens a browser and types the URL "`http://www.polito.it`". Determine the number and the type of the frames captured by a sniffer located on the cable that connects host H1 to the LAN, supposing that:

- The ARP cache on all the devices is empty
- The DNS cache on all the clients is empty
- The cache on the HTTP proxy contains all the requested HTTP pages
- The DNS server is either authoritative for the domains involved or the information is already present in its cache (i.e. no interactions with additional DNS servers are required)
- Routers have the proper routes toward all the destinations and therefore they should be able to reach all the destinations present in the network (unless their Ethernet and/or IP configuration is incorrect)



3.3. Exercise 18

Referring to the network topology depicted below, let us suppose that the owner of host H1 opens a browser and types the URL "`http://www.polito.it`". Describe the possible errors that may have occurred into the network and that prevented the visualization of the page and, whenever possible, show the possible tools that can be used to diagnose these errors.



Part III.

Solutions

4. IP traffic analysis

4.1. Solution for exercise 1

Since the network is based on a shared bus, the sniffer will capture all the frames that will be transmitted over the LAN.

Both source (host H1) and destination (host H2) belong to the same IP network, therefore they are directly reachable without having to interact with the default router.

Since the IP address of the destination is known (it is specified by the user in the "ping" command), no interaction with the DNS is required. Therefore, both the router R and the DNS are useless in this exercise and might be omitted in the network topology.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
2	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.2 has MAC = 00:00:00:22:22:22
3	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
4	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply
5-10	Packets 3 and 4 are replicated 3 times			

4.2. Solution for exercise 2

In this exercise, hosts H1 and H2 belongs to different networks connected by a single router R.

Since we have two sniffing points, the answer will list the packet in the temporal order they are captured, indicating the sniffing point in which those have been seen.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	Sniffer	L2	L3	Highest layer protocol	Description
1	A	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.254 please reply with its MAC address
2	A	00:00:00:EE:EE:EE → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.254 has MAC = 00:00:00:EE:EE:EE
3	A	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 130.192.17.2	ICMP	ICMP Echo Request
4	B	00:00:00:CC:CC:CC → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.17.2 please reply with its MAC address
5	B	00:00:00:22:22:22 → 00:00:00:CC:CC:CC	—	ARP Reply	Host 130.192.17.2 has MAC = 00:00:00:22:22:22
6	B	00:00:00:CC:CC:CC → 00:00:00:22:22:22	130.192.16.1 → 130.192.17.2	ICMP	ICMP Echo Request
7	B	00:00:00:22:22:22 → 00:00:00:CC:CC:CC	130.192.17.2 → 130.192.16.1	ICMP	ICMP Echo Reply
8	A	00:00:00:EE:EE:EE → 00:00:00:11:11:11	130.192.17.2 → 130.192.16.1	ICMP	ICMP Echo Reply
11-22	Packets 3, 6, 7 and 8 are replicated 3 times				

4.3. Solution for exercise 3

This exercise is definitely similar to the previous one.

The difficulty of this exercise is in the algorithm used to populate the ARP cache. Depending on the Operating System (OS) present on the hosts, there are two possibilities when an ARP Request packet is received:

- some OSs store in the ARP caches an entry for each ARP request received (i.e. the couple MAC source - IP source);
- some other OSs store in the ARP cache an entry only when the ARP request relates to the host itself (i.e. if the IP destination address contained in the ARP Request packet matches the one configured on the host; in that case they store the couple MAC source - IP source).

In the former case, host H3 will already have the MAC address of host H1 in its cache, although host H1 does not have the MAC address of host H3. In this case, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
2	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.2 has MAC = 00:00:00:22:22:22
3	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
4	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply
5-10	Packets 3 and 4 are replicated 3 times			
11	00:00:00:33:33:33 → 00:00:00:11:11:11	130.192.16.3 → 130.192.16.1	ICMP	ICMP Echo Request
12	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.3 please reply with its MAC address[1]
13	00:00:00:33:33:33 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.3 has MAC = 00:00:00:33:33:33
14	00:00:00:11:11:11 → 00:00:00:33:33:33	130.192.16.1 → 130.192.16.3	ICMP	ICMP Echo Reply
15-20	Packets 11 and 14 are replicated 3 times			

In the latter case, neither host H1 nor host H3 know each other, and the ARP discovery phase must be started from scratch in order to be able to send the ICMP Echo Request packet. In this case, the

[1] It is worthy noticing that apparently host H1 already knows the MAC address of host H3. In fact, previous packet #11 is directed from H3 to H1 and hence H1 receives that frame that has the MAC source address equal to H3. However, IP packets are used to *refresh* the ARP cache and cannot *populate* it. This is the reason why host H1 has to issue an ARP Request to discover H3's MAC address.

frames captured by the sniffer are the following:

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
2	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.2 has MAC = 00:00:00:22:22:22
3	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
4	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply
5-10	Packets 3 and 4 are replicated 3 times			
11	00:00:00:33:33:33 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.1 please reply with its MAC address
12	00:00:00:11:11:11 → 00:00:00:33:33:33	—	ARP Reply	Host 130.192.16.1 has MAC = 00:00:00:11:11:11
13	00:00:00:33:33:33 → 00:00:00:11:11:11	130.192.16.3 → 130.192.16.1	ICMP	ICMP Echo Request
14	00:00:00:11:11:11 → 00:00:00:33:33:33	130.192.16.1 → 130.192.16.3	ICMP	ICMP Echo Reply
15-20	Packets 13 and 14 are replicated 3 times			

Most OSs adopt the second algorithm. The rational is that if the ARP Request is directed to the host itself, it is very likely that in the near future the host will be contacted from the requester and that it will have to send a reply back. In this case, the matching between the IP and the MAC addresses of the requester will probably used very soon. Vice versa, if an ARP Request is received (please note that ARP Requests are sent in broadcast and hence are always received) but it is not directed to the host itself, the probability that the couple MAC - IP address will be used by this host in the foreseeable future is very small. Therefore, most OSs tend to discard this information in order to reduce the size of their ARP cache.
In the following exercises we will suppose that the OSs involved adopt the second algorithm, which is much more common in practice.

4.4. Solution for exercise 4

The behavior is determined by the expiration of the ARP cache on host H1. ICMP Echo Request messages will be sent anyway by host H1, even if H2 is no longer active, until the ARP cache expires. Obviously, the ICMP Echo Request is no longer present in this case.

When the ARP cache on host H1 expires, this host can no longer send the Ethernet frame containing the ICMP Echo Request, because the MAC address of host H2 is now unknown. Therefore the Operating System will try to refresh that cache by sending the ARP Request, which obviously is not followed by any response since host H2 is no longer present. Host H1 will try this several times (some OSs launch the ARP resolution process up to 5 times), until this process aborts.

At this point the OS will return a general error to the user application, which will be terminated automatically; an error of "general failure" (depending on the specific OS) will be printed on the screen.

In these conditions, the sniffer will capture a trace that looks similar to the following:

N.	L2	L3	Highest layer protocol	Description
N+1	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
N+2	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply
...	H2 is disconnected from the network			
M+1	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
M+2	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
...	ARP cache on host H1 expires			
K+1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
K+2	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
...	ARP Request repeated several times (e.g., 5 times in some OS)			

4.5. Solution for exercise 5

Since the network is based on a shared bus, the sniffer will capture all the frames transmitted over the LAN.

The destination host is specified by its hostname; therefore host H1 needs to interact with the DNS. Since the source host and the DNS server belong to the same IP network, the name resolution phase does not require to transit through a router.

Finally, both the source (host H1) and the destination (www.polito.it, host H2) belong to the same IP network; hence also in this case they do not require to transit from a router. Therefore, the router R is useless in this exercise and might be omitted in the network topology.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.253 please reply with its MAC address
2	00:00:00:DD:DD:DD → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.253 has MAC = 00:00:00:DD:DD:DD
3	00:00:00:11:11:11 → 00:00:00:DD:DD:DD	130.192.16.1 → 130.192.16.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
4	00:00:00:DD:DD:DD → 00:00:00:11:11:11	130.192.16.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 130.192.16.2
5	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
6	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.2 has MAC = 00:00:00:22:22:22
7	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
8	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply
9-14	Packets 11 and 14 are replicated 3 times			

4.6. Solution for exercise 6

1. Being a shared Ethernet, the NIC of host H2 will receive all the frames that are transmitted on the network.
2. In case the NIC is configured in promiscuous mode, the card does not apply any hardware filter to the traffic and therefore all these frames are also received by the operating system.
3. Vice versa, in case the NIC operates according to the standard mode (not promiscuous), the card will filter all the unnecessary traffic out, presenting to the operating system only the traffic referred to MAC addresses that are of interest for that station. In other words, the operating system will receive only the traffic that is directed to the MAC address of host H2 (i.e., 00:00:00:22:22:22), or it has a broadcast MAC address (i.e., FF:FF:FF:FF:FF:FF)[2].

According to the three configurations, the OS will receive the following frames:

N.	Description	(1) Received by the NIC	(2) Received by the OS (when NIC in promiscuous mode)	(3) Received by the OS (when NIC in standard mode)
1	ARP Request (130.192.16.253)	YES	YES	YES
2	ARP Reply (130.192.16.253)	YES	YES	NO
3	DNS Query (H2)	YES	YES	NO
4	DNS Answer (H2)	YES	YES	NO
5	ARP Request (130.192.16.2)	YES	YES	YES
6	ARP Reply (130.192.16.253)	YES	YES	NO
7	ICMP Echo Request	YES	YES	YES
8	ICMP Echo Reply	YES	YES	NO

[2]No multicast groups are configured in this exercise.

4.7. Solution for exercise 7

This exercise is definitely similar to the previous one. However, since the network is switched, the sniffer will capture only the packets generated by host H1 and those forwarded by the switch on its port connected to host H1.

However, all the frames already presented in previous exercise are either generated by host H1, or must be received by host H1 as well (because they are either directed to host H1's MAC address or are broadcast frames). Therefore, the sniffer will capture exactly the same set of frames of the previous exercise.

4.8. Solution for exercise 8

Since the network is based on a shared bus, the sniffer will capture all the frames that will be transmitted over the LAN. It is worth reminding that the sniffer captures frames and it does not care about the possible IP address in the L3 envelope. Therefore, even if this exercise proposes two IP networks on the same LAN, all the frames on the LAN will be captured by the sniffer.

The destination host is specified by its hostname; therefore host H1 needs to interact with the DNS. Since the source host and the DNS server belong to different IP networks, the name resolution phase requires packets to transit through a router.

Instead, both the source (host H1) and the destination (www.polito.it, host H2) belong to the same IP network; hence also in this case they do not require to transit from a router.

Please note that the router is able to forward the packets to the destination (e.g. from host H1 to the DNS) because it has two IP addresses on its interfaces: the first one (130.192.16.254) belongs to the same IP network of host H1 (and therefore it can be used to talk, at the IP level, to host H1) while the second belongs to the same network of the DNS (and therefore it can be used to talk, at the IP level, to the DNS).

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.254 please reply with its MAC address
2	00:00:00:EE:EE:EE → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.254 has MAC = 00:00:00:EE:EE:EE
3	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 130.192.17.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
4	00:00:00:EE:EE:EE → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.17.253 please reply with its MAC address
5	00:00:00:DD:DD:DD → 00:00:00:EE:EE:EE	—	ARP Reply	Host 130.192.17.253 has MAC = 00:00:00:DD:DD:DD
6	00:00:00:EE:EE:EE → 00:00:00:DD:DD:DD	130.192.16.1 → 130.192.17.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
7	00:00:00:DD:DD:DD → 00:00:00:EE:EE:EE	130.192.17.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 130.192.16.2
8	00:00:00:EE:EE:EE → 00:00:00:11:11:11	130.192.17.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 130.192.16.2
9	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address

10	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.2 has MAC = 00:00:00:22:22:22
11	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
12	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply
13-18	Packets 11 and 12 are replicated 3 times			

4.9. Solution for exercise 9

Since the network is switched, the sniffer will capture only the frames generated by host H1 and those forwarded on the port of the switch connected to host H1.

The destination host is specified by its hostname; therefore host H1 needs to interact with the DNS. Since the source host and the DNS server belong to the same IP network, the name resolution phase does not require to transit through a router.

Instead, source (host H1) and destination (www.polito.it, host H2) hosts belong to different IP networks and therefore a router is required to communicate.

Please note that the packets forwarded by R2 toward the destination will not be captured by the sniffer. In fact, those packets are no longer transmitted on the LAN and therefore cannot be received by the sniffer that is on the LAN. This applies independently from the LAN technology used, i.e. either with a shared Ethernet or a switched Ethernet.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.253 please reply with its MAC address
2	00:00:00:DD:DD:DD → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.253 has MAC = 00:00:00:DD:DD:DD
3	00:00:00:11:11:11 → 00:00:00:DD:DD:DD	130.192.16.1 → 130.192.16.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
4	00:00:00:DD:DD:DD → 00:00:00:11:11:11	130.192.16.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 32.10.1.3
5	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.254 please reply with its MAC address
6	00:00:00:EE:EE:EE → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.254 has MAC = 00:00:00:EE:EE:EE
7	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 32.10.1.3	ICMP	ICMP Echo Request
8	00:00:00:EE:EE:EE → 00:00:00:11:11:11	32.10.1.3 → 130.192.16.1	ICMP	ICMP Echo Reply
9-14	Packets 7 and 8 are replicated 3 times			

4.10. Solution for exercise 10

Since the network is based on a shared bus, the sniffer will capture all the frames that will be generated on the LAN, even if there are two IP networks on it.

The destination host is specified by its hostname; therefore host H1 needs to interact with the DNS. Since the source host and the DNS server belong to different IP networks, the name resolution phase requires packets to transit through a router. The same applies to the communication between the source (host H1) and the destination (www.polito.it), which are on different IP networks as well and hence require a router.

Please note that the Default Gateway of host H1 is Router R1, which can send traffic directly to the DNS server (R1 has two IP addresses, the former in the same IP network of host H1 while the latter in the same IP network of the DNS). However, the Default Gateway of the DNS Server is R2, which does not have an IP address in the same network of host H1. Therefore, host H1 is not directly reachable from R2, which can then send the traffic to R1 (for instance, R1 and R2 have an IP address that belong to the same IP network and therefore can talk to each other) and from here to host H1. However, router R2 will recognize that there is a better next hop for reaching the host H1 from the DNS, which will be R1, and this is the reason of the ICMP Redirect message that can be seen in frame 10.

Please note that for the same reason, all IP the ICMP packets from host H1 toward the destination will travel from host H1 to R1 and then to R2. In fact, since host H1 and R2 belong to different IP networks, hence they are not allowed to communicate directly. In this case ICMP Redirect messages cannot be used to optimize the path from host H1 to the destination; in fact, ICMP Redirect messages can be used only to create "shortcuts" if both devices are in the same IP network.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.254 please reply with its MAC address
2	00:00:00:EE:EE:EE → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.254 has MAC = 00:00:00:EE:EE:EE
3	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 130.192.17.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
4	00:00:00:EE:EE:EE → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.17.253 please reply with its MAC address
5	00:00:00:DD:DD:DD → 00:00:00:EE:EE:EE	—	ARP Reply	Host 130.192.17.253 has MAC = 00:00:00:DD:DD:DD
6	00:00:00:EE:EE:EE → 00:00:00:DD:DD:DD-DD	130.192.16.1 → 130.192.17.253	DNS Query	Get the IP address corresponding to name "www.polito.it"

7	00:00:00:DD:DD:DD → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.17.254 please reply with its MAC address
8	00:00:00:CC:CC:CC → 00:00:00:DD:DD:DD	—	ARP Reply	Host 130.192.17.254 has MAC = 00:00:00:CC:CC:CC
9	00:00:00:DD:DD:DD → 00:00:00:CC:CC:CC	130.192.17.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 32.10.1.3
10	00:00:00:CC:CC:CC → 00:00:00:DD:DD:DD	130.192.17.254 → 130.192.17.253	ICMP Redirect	A better next hop is available for destination 130.192.16.1: please use 130.192.17.1
11	00:00:00:CC:CC:CC → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.17.1 please reply with its MAC address
12	00:00:00:EE:EE:EE → 00:00:00:CC:CC:CC	—	ARP Reply	Host 130.192.17.1 has MAC = 00:00:00:EE:EE:EE
13	00:00:00:CC:CC:CC → 00:00:00:EE:EE:EE	130.192.17.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 32.10.1.3
14	00- EE-EE-EE-EE-EE → 00:00:00:11:11:11	130.192.17.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 32.10.1.3
15	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 32.10.1.3	ICMP	ICMP Echo Request
16	00:00:00:EE:EE:EE → 00:00:00:CC:CC:CC	130.192.16.1 → 32.10.1.3	ICMP	ICMP Echo Request
17	00:00:00:CC:CC:CC → 00:00:00:EE:EE:EE	32.10.1.3 → 130.192.16.1	ICMP	ICMP Echo Reply
18	00:00:00:EE:EE:EE → 00:00:00:11:11:11	32.10.1.3 → 130.192.16.1	ICMP	ICMP Echo Reply
19-30	Packets 15-18 are replicated 3 times			

4.11. Solution for exercise 11

This exercise is definitely similar to the previous one. The difference is that router R1 (which is the DG for host H1) belongs only to the network 130.192.16.0/24 and it is not able to talk with router R2 on the LAN, because the latter belongs to the 130.192.17.0/24 network. Moreover, there are no alternative paths from R1 to R2 because R1 is connected to a private network, while R2 is connected to the public Internet and these (the private network and the Internet) are not connected together.

Hence host H1 cannot resolve the destination hostname because the router R1, which receives the DNS Request from host H1, is unable to forward the packets to a destination belonging to network 130.192.17.0/24, which is the network the DNS belongs to.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.254 please reply with its MAC address
2	00:00:00:EE:EE:EE → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.254 has MAC = 00:00:00:EE:EE:EE
3	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 130.192.17.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
4	00:00:00:EE:EE:EE → 00:00:00:11:11:11	130.192.16.254 → 130.192.16.1	ICMP	ICMP Destination unreachable: Network unreachable

4.12. Solution for exercise 12

This exercise is definitely similar to the previous two. Also in this case routers R1 and R2 cannot talk directly, because they belong to different IP networks on the LAN. However, in this case R1 and R2 have a path that can be used to exchange traffic between themselves, since both are connected to the Internet. In fact, we can suppose that both routers are properly configured at the routing level. In this case, router R1 knows that there is a route toward the network of the DNS (130.192.17.0/24) though the Internet, which is an information it receives from its routing protocols (not part of this exercise). The same applies for router R2: it does not know the position of network 130.192.16.0/24, but it knows that it is reachable by forwarding the packets to its next hop on the Internet.

The resulting packet flow is the following:

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.254 please reply with its MAC address
2	00:00:00:EE:EE:EE → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.254 has MAC = 00:00:00:EE:EE:EE
3	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 130.192.17.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
This packet is forwarded onto the Internet and will reach R2 at some point. Then, R2 will forward it to the DNS server.				
4	00:00:00:CC:CC:CC → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.17.253 please reply with its MAC address
5	00:00:00:DD:DD:DD → 00:00:00:CC:CC:CC	—	ARP Reply	Host 130.192.17.253 has MAC = 00:00:00:DD:DD:DD
6	00:00:00:CC:CC:CC → 00:00:00:DD:DD:DD	130.192.16.1 → 130.192.17.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
7	00:00:00:DD:DD:DD → 00:00:00:CC:CC:CC	130.192.17.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 32.10.1.3
8	00:00:00:EE:EE:EE → 00:00:00:11:11:11	130.192.17.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 32.10.1.3
9	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	130.192.16.1 → 32.10.1.3	ICMP	ICMP Echo Request
10	00:00:00:EE:EE:EE → 00:00:00:11:11:11	32.10.1.3 → 130.192.16.1	ICMP	ICMP Echo Reply
11-16	Packets 9 and 10 are replicated 3 times			

4.13. Solution for exercise 13

Since the network is switched, the sniffer will capture only the frames generated by host H1 and those forwarded on the port of the switch connected to host H1.

The most important point of this exercise is that the DNS is mis-configured compared to the rest of the network, in that it has a /25 network while the other hosts have a /24 network. Therefore, the DNS server belongs to the 130.192.16.128/25 network, which does not include the IP addresses of host H1 and host H2. Vice versa, both host H1, host H2 and the router are configured in such a way that the belong to the same IP network of the DNS server. This has 2 consequences:

- Host H1 can interact with the DNS server directly at layer 2, without a router.
- The DNS server, believing host H1 is in a different IP network, forwards to its default gateway all the IP datagrams whose destination is host H1.

In other words, host H1 will try to reach the DNS server directly (through an ARP Request). The DNS server will reply to this ARP request because the Operating Systems usually does not check the if the IP address of the requester belongs to the same IP network of the host itself.

However, when the DNS server has to send the DNS Answer back to H1, it will detect that the IP destination address is within another network and therefore the packet has to be sent through a router. Since the MAC address of the router is still unknown the DNS server will issue an ARP Request toward the router, and only after than phase the DNS Answer message will be sent to the router, which will forward it to host H1 (after another round of ARP messages since the MAC address of host H1 is still unknown to the router.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.253 please reply with its MAC address
2	00:00:00:DD:DD:DD → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.253 has MAC = 00:00:00:DD:DD:DD
3	00:00:00:11:11:11 → 00:00:00:DD:DD:DD	130.192.16.1 → 130.192.16.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
4	00:00:00:DD:DD:DD → FF:FF:FF:FF:FF:FF	—	ARP Request[3]	Who has IP=130.192.16.254 please reply with its MAC address
5	00:00:00:EE:EE:EE → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.1 please reply with its MAC address

[3]Please note that this frame is followed by the ARP Reply coming from the router and the DNS Answer sent by the DNS to its default gateway. However, those two frames are not captured by the sniffer because they are not forwarded by the switch on the port where host H1 is located.

6	00:00:00:11:11:11 → 00:00:00:EE:EE:EE	—	ARP Reply	Host 130.192.16.1 has MAC = 00:00:00:11:11:11
7	00:00:00:EE:EE:EE → 00:00:00:11:11:11	130.192.16.253 → 130.192.16.1	DNS Answer	Host "www.polito.it" has IP= 130.192.16.2
8	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.2 please reply with its MAC address
9	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.2 has MAC = 00:00:00:22:22:22
10	00:00:00:11:11:11 → 00:00:00:22:22:22	130.192.16.1 → 130.192.16.2	ICMP	ICMP Echo Request
11	00:00:00:22:22:22 → 00:00:00:11:11:11	130.192.16.2 → 130.192.16.1	ICMP	ICMP Echo Reply
12-17	Packets 10 and 11 are replicated 3 times			

4.14. Solution for exercise 14

This exercise is definitely similar to the previous one. However, in this case the default gateway configured in the DNS belongs to a different IP network, and therefore the DNS cannot reach its router. Therefore, the packet trace will be interrupted as soon as the DNS has to send a frame to its default gateway.

However, since the DNS is encapsulated in the (unreliable) UDP protocol, the source host cannot know whether the DNS Query was lost (e.g., because of network problems) or the DNS Answer was lost. Hence the source host will keep sending the DNS Query for a given number of times (e.g., 5 times on some operating systems), until it aborts.

In these conditions, the sniffer will capture the following frames:

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=130.192.16.253 please reply with its MAC address
2	00:00:00:DD:DD:DD → 00:00:00:11:11:11	—	ARP Reply	Host 130.192.16.253 has MAC = 00:00:00:DD:DD:DD
3	00:00:00:11:11:11 → 00:00:00:DD:DD:DD	130.192.16.1 → 130.192.16.253	DNS Query	Get the IP address corresponding to name "www.polito.it"
4-8	Packet 3 is replicated N times (e.g., 5 times)			

4.15. Solution for exercise 15

Host H1 has a netmask (/23) that includes also the IP address of the DNS server in the same IP network of the host itself. Therefore, host H1 will send an ARP Request on the network asking for the corresponding MAC address.

Unfortunately, it appears that host H1 is poorly configured and that its netmask is incorrect. Therefore, its ARP Request will never reach the destination, and therefore host H1 is unable to resolve that name.

The ARP Request is re-sent many times by the operating system (the exact number depends on the operating system) in order to overcome possible losses in the network, until an error is issued to the application.

In these conditions, the sniffer will capture the following frames:

N.	L3	Highest layer protocol	Description
1	—	ARP Request	Who has IP=130.192.17.2 please reply with its MAC address
2 - ...	The previous frame is repeated N times in order to overcome possible losses in the network		

5. Application-layer traffic analysis

5.1. Solution for exercise 16

This exercise is very similar to the previous ones; while apparently it involves an application-level protocol (HTTP through the HTTP proxy function), in fact this is not true. For instance, the configuration of the host H1 includes also an HTTP proxy, but the command typed on the host itself is a simple "`ping`", which does not involve the HTTP protocol at all.

In fact, the "`ping`" command will generate the usual ICMP Echo Request packets, which are completely unrelated from the HTTP proxy functionality. Therefore, the HTTP Proxy configuration is there only to confuse the student, but it has no effects at all on the exercise.

In these conditions, the sniffer will capture the following frames (considering that a Windows host generates 4 ICMP Echo Request packets):

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=172.16.64.1 please reply with its MAC address
2	00:00:00:CC:CC:CC → 00:00:00:11:11:11	—	ARP Reply	Host 172.16.64.1 has MAC = 00:00:00:CC:CC:CC
3	00:00:00:11:11:11 → 00:00:00:CC:CC:CC	172.16.64.11 → 172.16.10.2	DNS Query	Get the IP address corresponding to name "www.polito.it"
4	00:00:00:CC:CC:CC → 00:00:00:11:11:11	172.16.64.1 → 172.16.64.11	ICMP Redirect	A better next hop is available for destination 172.16.10.2: please use 172.16.64.2
5	00:00:00:CC:CC:CC → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=172.16.64.2 please reply with its MAC address
6	00:00:00:AA:AA:AA → 00:00:00:CC:CC:CC	—	ARP Reply	Host 172.16.64.2 has MAC = 00:00:00:AA:AA:AA
7	00:00:00:CC:CC:CC → 00:00:00:AA:AA:AA	172.16.64.11 → 172.16.10.2	DNS Query	Get the IP address corresponding to name "www.polito.it"
8	00:00:00:AA:AA:AA → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=172.16.64.11 please reply with its MAC address

9	00:00:00:11:11:11 → 00:00:00:AA:AA:AA	—	ARP Reply	Host 172.16.64.11 has MAC = 00:00:00:11:11:11
10	00:00:00:AA:AA:AA → 00:00:00:11:11:11	172.16.10.2 → 172.16.64.11	DNS Answer	Host "www.polito.it" has IP=172.16.64.6
11	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=172.16.64.6 please reply with its MAC address
12	00:00:00:22:22:22 → 00:00:00:11:11:11	—	ARP Reply	Host 172.16.64.6 has MAC = 00:00:00:22:22:22
13	00:00:00:11:11:11 → 00:00:00:22:22:22	172.16.64.11 → 172.16.64.6	ICMP	ICMP Echo Request
14	00:00:00:22:22:22 → 00:00:00:11:11:11	172.16.64.6 → 172.16.64.11	ICMP	ICMP Echo Reply
15-20	Frames 13 and 14 are replicated 3 times			

5.2. Solution for exercise 17

Host H1 requests an HTTP page through the HTTP Proxy. Since host H1 is statically configured with the IP address of the HTTP Proxy, it can generate immediately an HTTP GET message toward it.

On order to complete this step, the only additional frames requested are the ARP Request toward its default gateway R2 in order to know its MAC address, and the TCP 3-way handshake in order to establish the TCP connection. After those packets, the HTTP GET message can be sent directly to the HTTP Proxy.

In these conditions, the sniffer will capture the following frames:

N.	L2	L3	Highest layer protocol	Description
1	00:00:00:11:11:11 → FF:FF:FF:FF:FF:FF	—	ARP Request	Who has IP=172.16.64.1 please reply with its MAC address
2	00:00:00:CC:CC:CC → 00:00:00:11:11:11	—	ARP Reply	Host 172.16.64.1 has MAC = 00:00:00:CC:CC:CC
3	00:00:00:11:11:11 → 00:00:00:CC:CC:CC	172.16.64.11 → 172.16.15.3	TCP SYN	First packet of the TCP 3-way handshake
4	00:00:00:CC:CC:CC → 00:00:00:11:11:11	172.16.15.3 → 172.16.64.11	TCP SYN-ACK	Second packet of the TCP 3-way handshake
5	00:00:00:11:11:11 → 00:00:00:CC:CC:CC	172.16.64.11 → 172.16.15.3	TCP ACK	Third packet of the TCP 3-way handshake
6	00:00:00:11:11:11 → 00:00:00:CC:CC:CC	172.16.64.11 → 172.16.15.3	HTTP GET	Host: www.polito.it
7	00:00:00:CC:CC:CC → 00:00:00:11:11:11	172.16.15.3 → 172.16.64.11	TCP ACK	TCP acknowledgement[1]
8	00:00:00:CC:CC:CC → 00:00:00:11:11:11	172.16.15.3 → 172.16.64.11	HTTP Response	200 OK
9	00:00:00:11:11:11 → 00:00:00:CC:CC:CC	172.16.64.11 → 172.16.15.3	TCP ACK	TCP acknowledgement
10 - ...	N frames that transport the requested HTTP page			

The number of HTTP packets exchanged between the two hosts is not reported here because it depends on the size of the requested HTTP object.

[1] Please note that most of the current implementations explicitely generate a TCP acknowledgement packet even if the ACK flag could be transported in piggyback mode in the following data packet flowing in the opposite direction. Obviously, the same reason applies to packet #9.

5.3. Solution for exercise 18

From the network topology shown in the picture, the impossibility to retrieve the requested HTTP page is due to a misconfiguration of the netmask on the requesting host, which believes the DNS is reachable through a direct Ethernet frame, while this is not true. However, this is clear because we have the full network topology (and the related configuration); however, the user on host H1 does not have all this data and has to do some additional investigations in order to determine the cause of the error.

In general, the impossibility to retrieve a web page is often due to a network error. It may be a DNS error (e.g., wrong name, wrong configuration of the DNS system, wrong configuration of the DNS server), a network problem (e.g., impossibility to reach the default gateway from some of the hosts involved in the transaction, some faults on the path toward the host H1 and the destination, etc), a misconfigured host (e.g., a netmask error, a wrong address for the default gateway, etc), or even an application error (e.g. the HTTP server is not active on the target machine). Some of those errors can be diagnosed with the proper "`ping`" or "`traceroute`" commands, which test the availability of the path, although not all the errors can be detected by having only the control of the host H1. For instance, the fact that the DNS server may be unable to reach its default gateway because of a wrong netmask, cannot be detected from host H1.

In our case, a network manager that activates a sniffer on host H1 will easily detect the problem. In fact, being H1 on a shared network (i.e., the sniffer can intercept the traffic of all the hosts on the LAN) we can easily see that all the other hosts will send the traffic for the destinations outside network 130.19.16.0/24 to the default gateway. Another useful test can be to scan the entire network 130.192.16.0/23 with the "`ping`" command, and we will see that hosts whose addresses begin with 130.192.16.* are usually reachable, while all hosts whose addresses begin with 130.192.17.* appear unreachable[2].

At this point it is easy to conclude that probably the netmask on host H1 is wrong and that a /24 netmask should be used instead.

[2]We suppose that each LAN segment includes more hosts than the ones depicted in the network topology.