ECE 4309

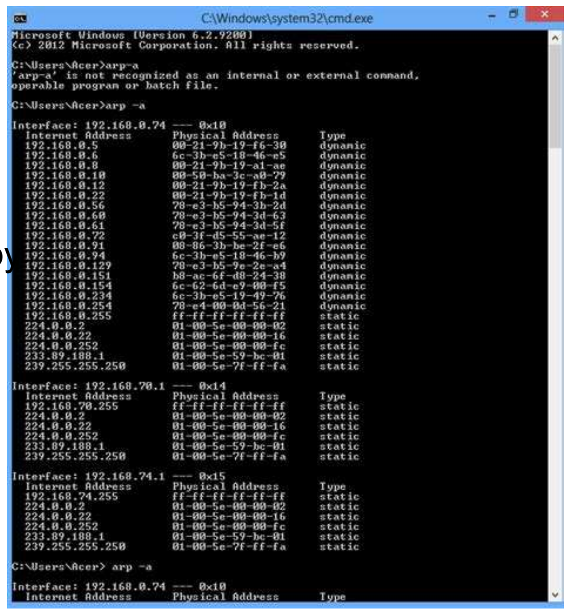
Basics of computer networks and systems - Appendix

Dr. Valerio Formicola

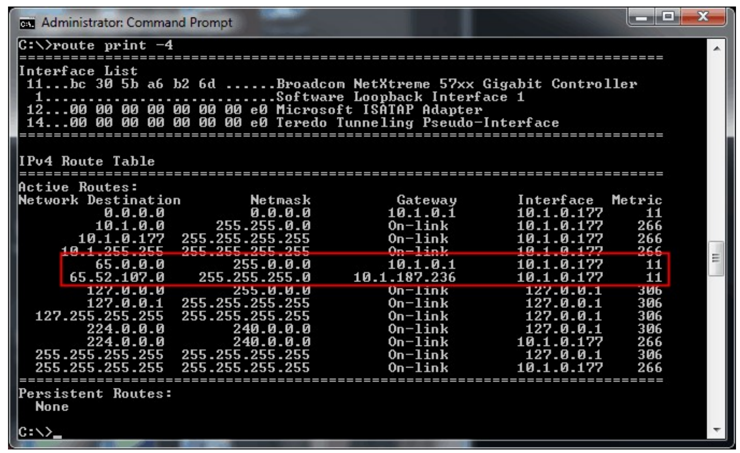
ARP table and Routing Table

- Each Node has an ARP table and a Routing Table: one for each network IP address it has.
- Including the Router, which usually has many interfaces, hence many ARP tables (aggregated) and Routing Tables (aggregated)!

ARP table:
Contains all the Physical addresses (MAC addresses)
learned during the time by a node associated
to their IP addresses



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Acer>arp-a
'arp-a' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Acer>arp -a

Interface: 192.168.0.74 --- 0x10
  Internet Address      Physical Address      Type
  192.168.0.5           00-21-9b-19-f6-30     dynamic
  192.168.0.6           6c-3b-e5-18-46-e5     dynamic
  192.168.0.8           00-21-9b-19-a1-ae     dynamic
  192.168.0.10          00-50-ba-3c-a0-79     dynamic
  192.168.0.12          00-21-9b-19-fb-2a     dynamic
  192.168.0.22          00-21-9b-19-fb-1d     dynamic
  192.168.0.56          78-e3-b5-94-3b-2d     dynamic
  192.168.0.60          78-e3-b5-94-3d-63     dynamic
  192.168.0.61          78-e3-b5-94-3d-5f     dynamic
  192.168.0.72          c0-3f-d5-55-ae-12     dynamic
  192.168.0.91          08-86-3b-be-2f-e6     dynamic
  192.168.0.94          6c-3b-e5-18-46-b9     dynamic
  192.168.0.129         78-e3-b5-9e-2e-a4     dynamic
  192.168.0.151         b8-ac-6f-d8-24-38     dynamic
  192.168.0.154         6c-62-6d-e9-00-f5     dynamic
  192.168.0.234         6c-3b-e5-19-49-76     dynamic
  192.168.0.254         78-e4-00-0d-56-21     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  233.89.188.1          01-00-5e-59-bc-01     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.70.1 --- 0x14
  Internet Address      Physical Address      Type
  192.168.70.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  233.89.188.1          01-00-5e-59-bc-01     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.74.1 --- 0x15
  Internet Address      Physical Address      Type
  192.168.74.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  233.89.188.1          01-00-5e-59-bc-01     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

C:\Users\Acer> arp -a

Interface: 192.168.0.74 --- 0x10
  Internet Address      Physical Address      Type
```



```
Administrator: Command Prompt
C:\>route print -4
===========================================================================
Interface List
 11...bc 30 5b a6 b2 6d ......Broadcom NetXtreme 57xx Gigabit Controller
  1...........................Software Loopback Interface 1
 12...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
 14...00 00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0         10.1.0.1      10.1.0.177     11
         10.1.0.0      255.255.0.0         On-link       10.1.0.177    266
       10.1.0.177  255.255.255.255         On-link       10.1.0.177    266
     10.1.255.255  255.255.255.255         On-link       10.1.0.177    266
         65.0.0.0        255.0.0.0         10.1.0.1      10.1.0.177     11
      65.52.107.0    255.255.255.0     10.1.187.236      10.1.0.177     11
        127.0.0.0        255.0.0.0         On-link        127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link        127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link        127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link        127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link       10.1.0.177    266
  255.255.255.255  255.255.255.255         On-link        127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link       10.1.0.177    266
===========================================================================
Persistent Routes:
  None

C:\>_
```

Routing table:
Contains all the associations
IP network addresses with Default Gateway (i.e., the router most often)
to be used for sending out the messages
to that network

Examples of LAN to LAN communications at Network and Data Link layers

Example of Eth. LANs – local communication (1/4)

- Suppose a user on host WK1 (workstation 1) wants to use a SSH client to connect to a SSH server on host WK2
  - Also, suppose user on WK1 knows the IP address of host WK2
  - To communicate WK1 generates some Ethernet *frames* over the cables; the frames contain IP *packets,* which transport TCP *segments*, which contain *messages/commands* from the SSH client (i.e., the app. running in WK1) to the SSH server (i.e., the app. running in WK2)

User ssh command in WK1 to connect to ssh in WK2

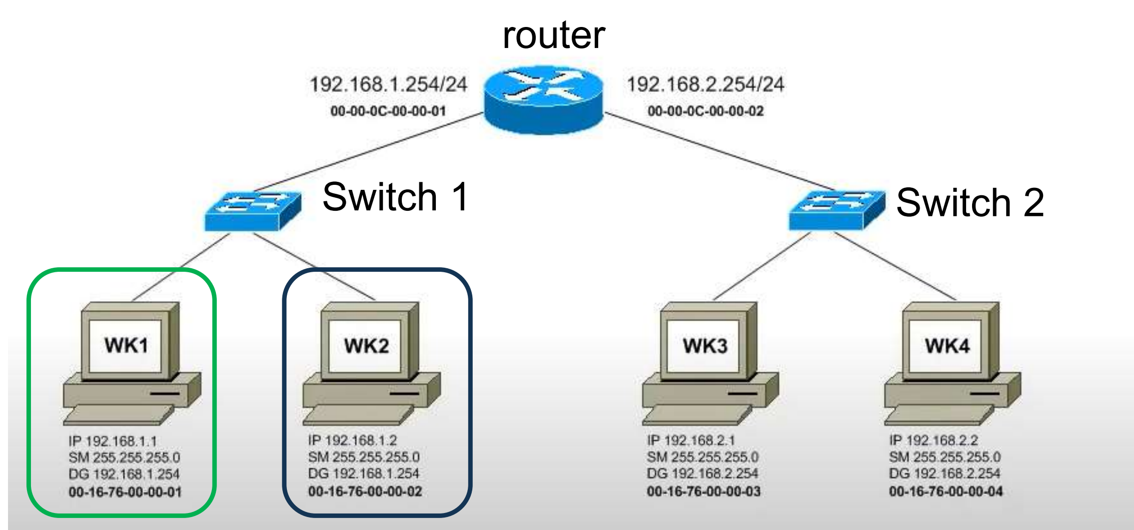
userX@192.168.1.1:~$ *ssh userX@192.168.1.2*

SSH app messages

TCP segments

IP packets

Ethernet frame

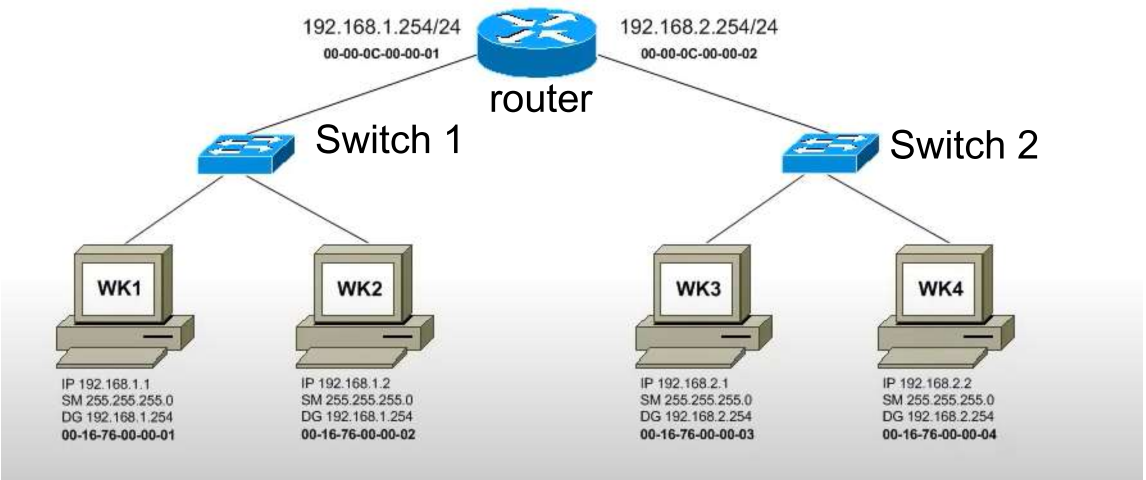


Example extended from video: https://www.youtube.com/watch?v=XP61HtbGPbA

Example of Eth. LANs – local communication (2/4)

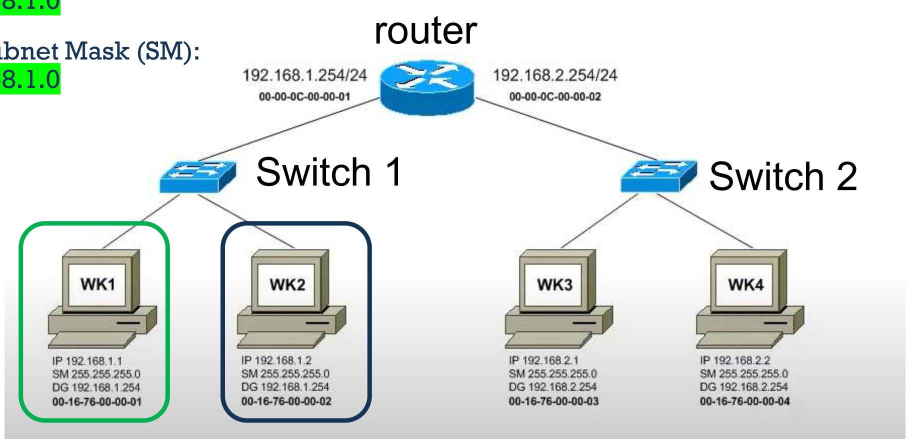
1. Since user WK1 knows the IP address of WK2, client SSH application has no need to use a DNS to find the IP address of WK2 (i.e., there is no need to resolve a URL)
2. When user executes the SSH command, SSH starts to generate data transformed into TCP segments (T-PDUs) that need to reach WK2. WK2 is waiting for incoming messages on the default TCP port 22
   - This happens because SSH application uses TCP as a transport layer, hance WK1 and WK2 use a TCP connection to exchange messages
3. TCP messages - generated by the SSH client, - at this point, need to be transported throuhg the physical network and reach somehow node WK2

Question: How the packets in WK1 know the way to reach WK2?
I.e., how to find the path from WK1 to WK2?



Example of Eth. LANs – local communication (3/4)

4. WK1 first calculates the BITWISE_AND of source IP address (its own, IP_WK1) and the Subnet Mask (SM):
   IP_WK1 *BITWISE_AND* WK1_SubNet_Mask => 192.16.1.1 AND 255.255.255.0 = 192.168.1.0
5. Then, WK1 calculates the BITWISE AND of destination IP address (IP_WK2) and the Subnet Mask (SM):
   IP_WK2 *BITWISE_AND* WK1_SubNet_Mask => 192.16.1.2 AND 255.255.255.0 = 192.168.1.0
6. WK1 compares the results and finds out they are the same!
   This means that destination WK2 is on the same subnetwork of WK1
7. At this point, WK1 knows that it can just send a packet on the same physical medium because WK2 is physically connected to WK1 (through *switch 1*, which might also be connected to many other computers on the same subnet).
   (Remember, a LAN is a shared medium)
8. Hence, WK1 creates the entire packet, with data, TCP headers, IP headers and Ethernet headers:



"Full" packet from WK1 to WK2

SSH Data: 01110101…
(whatever is the content of ssh command)

TCP Header:
src port: 2929, dst: port 22,
+ other TCP fields

IP Header:
Ip-address source = 192.168.1.1
Ip-address destination = 192.168.1.2
+ other IP fields

Ethernet Header:
MAC-source = 00-16-76-00-00-01
MAC-destination = ???,
+ other MAC layer fields

Example of Eth. LANs – local communication (4/4)

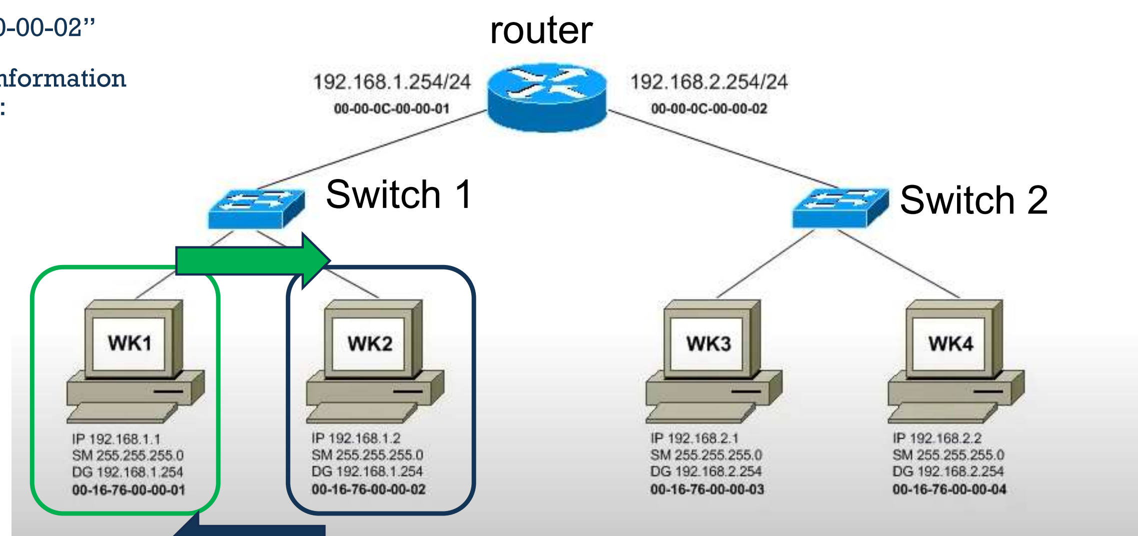
9. However, WK1 realizes that it doesn't know the MAC address of destination node (WK2) and needs to find it.
   Solution: ARP request to all machines physically connected to the same subnetwork (i.e., on Switch 1).
   WK1 will ask: "Which node has IP address 192.168.1.2?"
   WK2 will reply: "It's me, and I have physical address 00-16-76-00-00-02"

10. Finally, WK1 can complete the packet and frame with required information (here I reported only the part of address updated in the packet):
    Ip-address source = 192.168.1.1
    MAC-source = 00-16-76-00-00-01
    Ip-address destination = 192.168.1.2
    MAC-destination = 00-16-76-00-00-02

11. Finally, the frame can travel on the physical network of Switch 1.
    It will be received by all the hosts connected to Switch 1.
    However, only WK2 will recognize the destination MAC address and will collect the packet.

12. The packet received by WK2 can now dissected by WK2 to get the application layer data (upward process: IP-> TCP-> SSH application).

13. Later on, WK2 has to answer and create new TCP segments from WK2 to WK1. Since WK2 knows the MAC address of WK1 from the first communication, it will simply reuse it to reply WK1



Ip-address source = 192.168.1.2
MAC- source = 00-16-76-00-00-02
Ip-address destination = 192.168.1.1
MAC-destination = 00-16-76-00-00-01

+ other fields from all layers: App, TCP, IP, Eth

Packet from WK2 to WK1

Example of Eth. LANs – external communication (1/4)

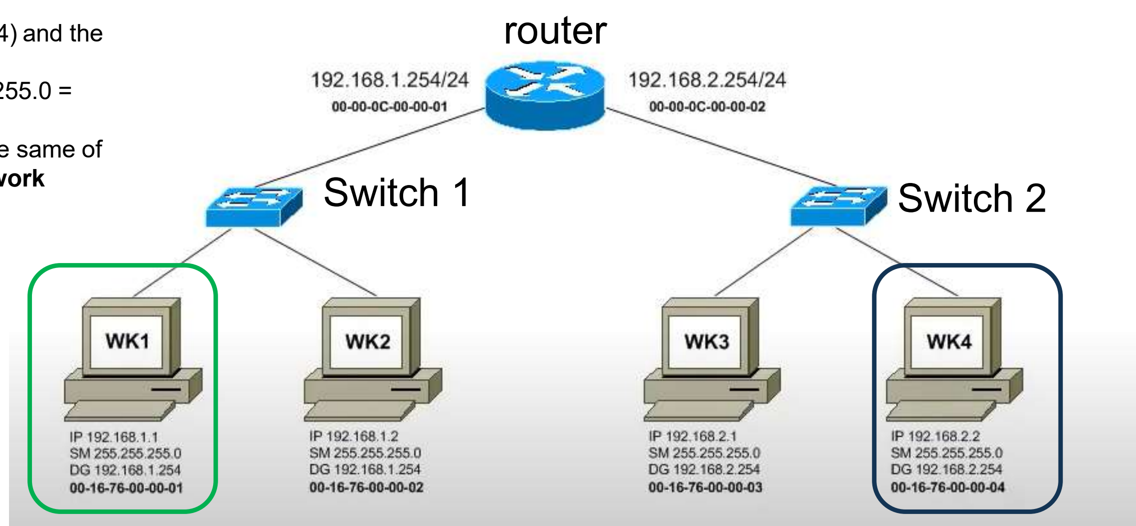
What if the SSH client in WK1 wants to talk to SSH server running in WK4 on a different network?

userX@192.168.1.1:~$ *ssh userX@192.168.1.4*

1. WK1 first calculates the BITWISE_AND of source IP address (its own, IP_WK1) and the Subnet Mask (SM):
   IP_WK1 BITWISE_AND WK1_SubNet_Mask => 192.16.1.1 AND 255.255.255.0 = 192.168.1.0
2. Then, WK1 calculates the BITWISE AND of destination IP address (IP_WK4) and the Subnet Mask (SM):
   IP_WK4 BITWISE_AND WK1_SubNet_Mask => 192.16.2.2 AND 255.255.255.0 = 192.168.2.0
3. WK1 compares the two products and finds out that the second one is not the same of the first one, hence the two hosts WK1 and WK4 are **not on the same network**

Question: Will it make sense for WK1 to know the MAC address of WK4?
Answer: No, because still WK4 is not physically connected to WK1 and so WK1 cannot send a message on its own local network, even after obtaining WK4 MAC_address

Question 2: So, given that WK4 is not physically reachable directly from WK1, what WK1 can do?



Example of Eth. LANs – external communication (2/3)

4. Each host (including WK1) has something called **ROUTING TABLE (**different for each host). It contains information about where to send each packet belonging to a specific subnetwork, especially does not directly connected to the same. If the subnetwork of the destination node is **not** in the Routing Table, then the host will send the packet to the **Default Gateway** (DG, in the figure)**.**
   - **Usually, the Default Gateway in a Subnetwork is also the Router connected to the Subnetwork itself.**
   - So, the Default Gateway for WK1 is 192.169.1.254 (i.e., router IP address on the subnetwork of WK1)
5. Since WK1 has the Router as a Default Gateway, the packet from WK1 will use the WK2 IP address and the Default Gateway MAC address for the destination. Hence, the packet created by WK1 will look like something like this

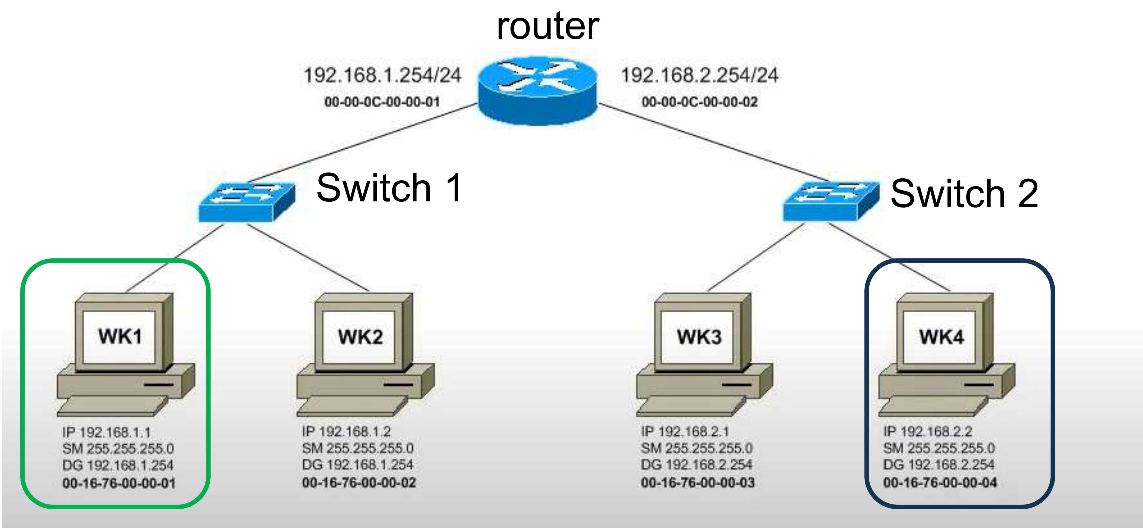
"Full" packet from WK1 to WK2

SSH Data: 01110101…
(whatever is the content of ssh command)

TCP Header:
src port: 2929, dst: port 22,
+ other TCP fields

IP Header:
Ip-address source = 192.168.1.1
Ip-address destination = 192.168.2.2
+ other IP fields

Ethernet Header:
MAC-source = 00-16-76-00-00-01
MAC-destination = 00-00-0C-00-00-01,
+ other MAC layer fields

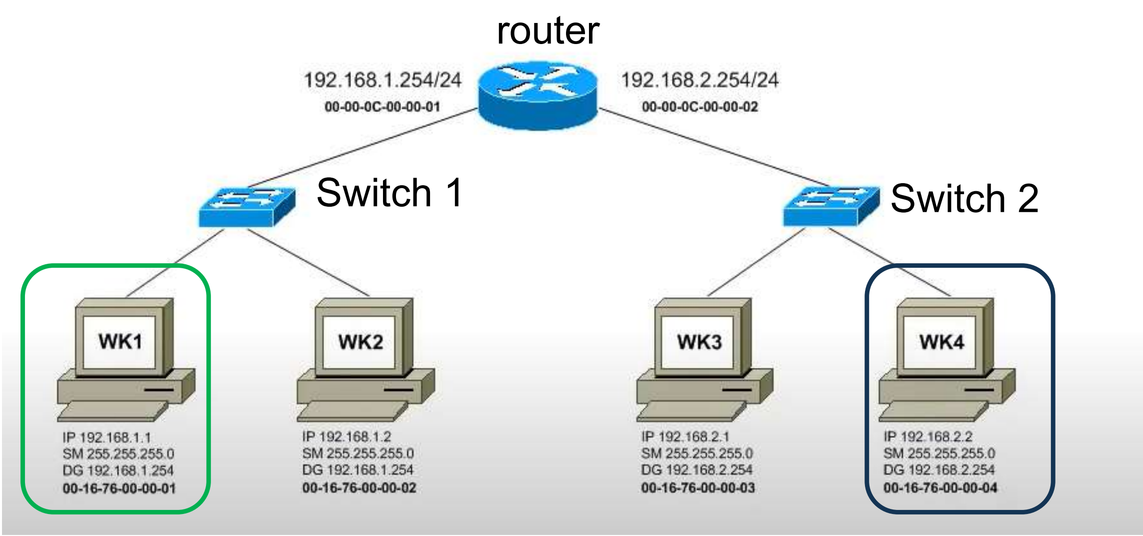


Example of Eth. LANs – external communication (3/3)

6. Hence, since the packet sent by WK1 has the WK4 IP address as a destination but the router MAC address, it will be the router to receive it (on the Ethernet interface 00-00-0C-00-00-01).
7. The router knows that it’s not a packet for him, so the router decides to forward the packet to the actual destination, which is WK4 on network 192.168.2.0/24.
    - Indeed, the router again applies the BitMask to check what is the destination subnet, and it finds out that the destination IP address is on the other subnetwork.
8. To send the packet, the router now needs to talk to the other network. At the beginning, the router doesn’t know the MAC address of WK4 yet; hence it can complete the packet to forward.

...
IP Header:
Ip-address source = 192.168.1.1
Ip-address destination = 192.168.2.2
+ other IP fields

Ethernet Header:
MAC-source = 00-16-76-00-00-01
MAC-destination = ?,
+ other MAC layer fields



9. To proceed, the router has to resolve the MAC address for WK4 IP address: Router will ask: “Which physical card has IP address 192.168.2.2?” WK4 will reply: “It’s me, I have physical address 00-16-76-00-00-04”
10. At this point, the router can complete the packet with the final destination MAC address:

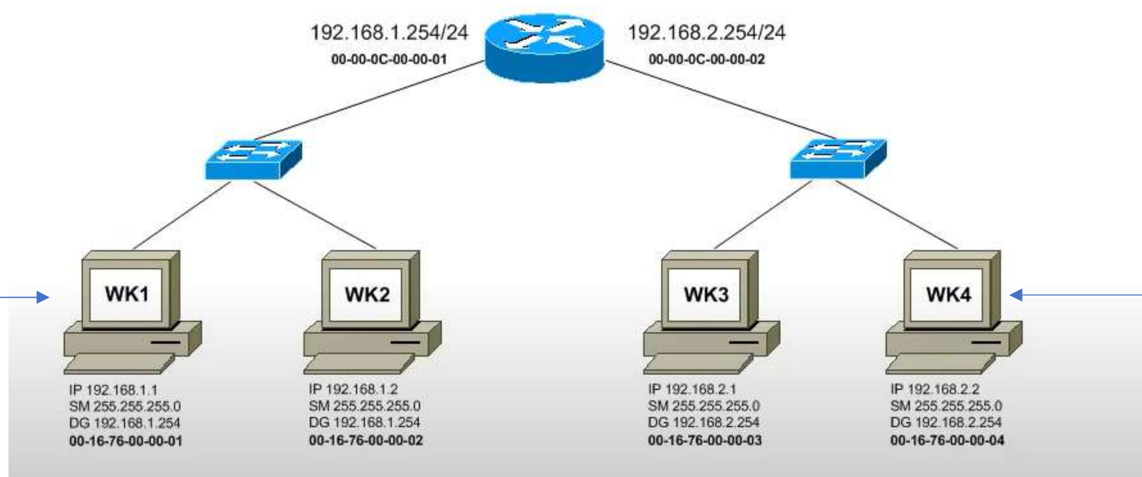
...
IP Header:
Ip-address source = 192.168.1.1
Ip-address destination = 192.168.2.2
+ other IP fields

Ethernet Header:
MAC-source = 00-16-76-00-00-01
MAC-destination = 00-16-76-00-00-04,
+ other MAC layer fields

Example with DNS

- Suppose host WK1 has to send a request to access a web-site server hosted WK4
- It's the same process of case two (external communication) but with a DNS server that needs to resolve the IP address of the Web Server.
- Usually, the Default Gateway is also hosting a small DSN server or, if not, it will connect all the hosts to an external DNS resolver

Web Browser
running here,
user wants access to
http://www.something.com



(Web) server containing
the web page
http://www.something.com