

# Practical Exercises in Computer Networks

## Experiments with the IP forwarding algorithm(WIP)

© 2013, José María Foces Morán

IP routers operate at the layer 3, the Network layer which functions consist of conveying IP packets in the internetwork. In former exercises we covered layer 2 communications in TCP/IP networks, now we transition to the internetwork and reproduce similar experiments to illustrate the forwarding function of IP routers. Other included exercises are oriented towards our understanding of other infrastructure and control-plane protocols such as ARP, DHCP and ICMP.

### Understanding IP forwarding

In the lectures, after introducing the IP service model, the IP packet format and its main constituent concepts, we need to discover the networking equipment that implements those IP functionalities, to that end we now want to devise a testbed internetwork for experimenting with IP, its forwarding algorithm and addressing.

**Basic requirements for the testbed internetwork.** We need an internetwork that allows us to probe traffic in any of its networks such that we check that the IP forwarding algorithm holds. The IP forwarding algorithm that we want to check is that explained in section 3.2.4 of the textbook; for convenience, we reproduce here that algorithm:

```
if (NetworkNum of destination = my NetworkNum) then
  deliver packet to destination directly
else
  deliver packet to default router
```

Fig. 1. Basic IP Forwarding Algorithm (IPFA)  
From Computer Networks Ed.5, ©2012 MKP, Prof. Peterson and Davie, pg.217

This form of the algorithm is the simplest, it assumes that the stations (hosts) used have a unique NIC installed and configured and that their *forwarding tables* contain a single *default router*. For the time being, that suffices for us. Fig. 2 captures these initial requirements.

This building block allows us to test IP traffic from Local Station 1.1 to Local Station 1.2 which clearly resembles the experiments carried out in the previous *Practical Exercise on Computer Networks* (IP forwarding in directly connected stations and ARP) and connects perfectly with the first paragraph of the IPFA mentioned above (Fig. 1): If the IP packet's destination address has a network prefix (Network number) that equals my network prefix, I will send that packet directly to the station since it is directly connected to me, *i.e.* we both belong to the same network. When two stations belong to the same network, we say they are directly connected even though they might be connected by means of several Ethernet switches (Recall, in this case we have a single broadcast domain, distributed along several network switches in an extended LAN). Let's work this experiment now.

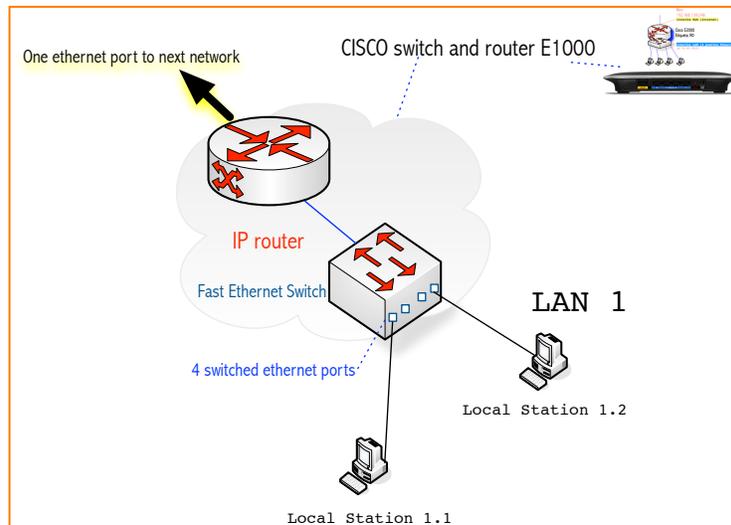


Fig. 2. Basic testbed internetwork building block for this exercise

### Experiment 1: IPFA when (NetworkNum of Destination is) == (my NetworkNum)

1. Set the IP configuration of your Cisco router according to Appendix 1 and your LAN number assigned (This number will serve as prefix for referencing the local stations in this document)
2. Connect two stations to the router's Ethernet switch, both must use DHCP to fetch their IP configuration
3. Run Wireshark in each station
4. Send a ping from your Local Station 1.1 (LS1) to your Local Station 1.2 (LS2)
5. Capture the resulting trace in both stations and confirm that the ICMP/IP packet from LS1 to LS2 travelled encapsulated in an Ethernet frame whose destination MAC is that of LS2 and whose source MAC is that of LS1
6. Flip the roles of LS1 and LS2 and perform the same check as that in step 5, confirming the same results but with the MACs swapped

As expressed in Fig. 2, the IPFA, when the IP packet is destined towards a station that belongs to a network prefix other than theirs, proceeds to convey it to its default router. The default router is directly connected to a Local Station, always, therefore, transmitting a frame from LS1 to LS2 is absolutely equivalent to transmitting a frame from LS1 to its default router.

Fig. 3 illustrates that a router interface to a LAN behaves exactly as though it were a common station LAN connection (NIC)<sup>1</sup>. If LS1 needs to send a frame to LS2, first, it finds out its MAC address by using ARP, then fabricates a new frame that contains its MAC as source and that of LS2 as destination. Now, if for whatever reason, LS1 needs to send a frame to the Router Interface, it proceeds as explained here: it finds out the router's interface MAC by using ARP, then, fabricates the frame with its own MAC as source and the router's interface MAC as destination.

<sup>1</sup> Actually a Linux or Open-BSD station can host a full fledged IP router if we install the appropriate hardware, software and establish its forwarding and routing information correctly.

Within a LAN that has a router connected, any of its stations sends Ethernet frames to the router interface by using exactly the same procedure that is used when the stations communicate between them

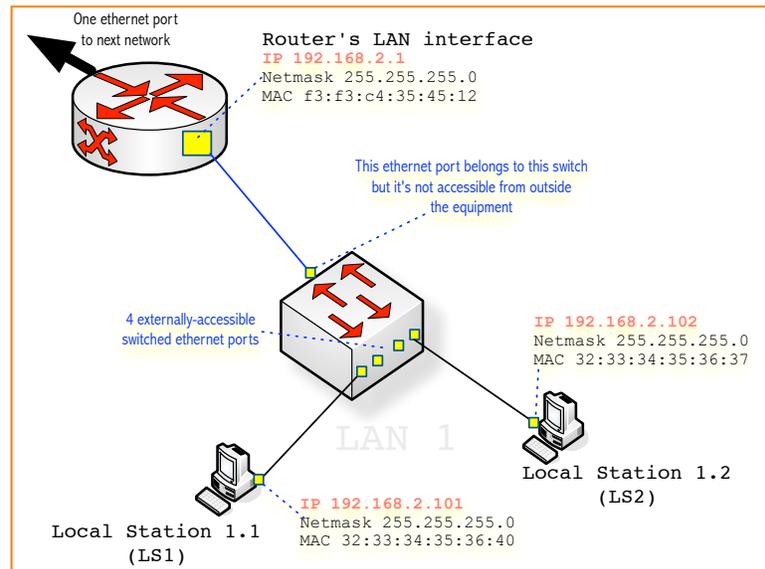


Fig. 3. Layer-2 equivalence of LS1, LS2 and the Router Interface within LAN1

Why have we just explained the layer-2 communication between an LS with the router interface? When we want to communicate with a remote station (RS, for the purposes of this Practical Exercise), for example, if you want to contact the web server at paloalto.unileon.es (Its IP address is 193.146.101.46), you absolutely need the intervention of your router, it will hand your paloalto-directed packets to the Internet. The question is, how? The answer is in the *else* clause of our basic IPFA in Fig. 1, it says “in case the packet is for a network prefix different from mine, I will send the packet to my default router”. Before checking this part of the IPFA, let’s devise a modification of experiment 1 to check the layer-2 equivalence of router interface and that of any station belonging to the same network.

## Experiment 2: Layer-2 equivalence of router interface and a station’s

1. Make sure your router has the correct configuration per appendix 1
2. Connect one station (LS1) to the router’s Ethernet switch, it must use DHCP to fetch its IP configuration
3. Run Wireshark in the station
4. Send a ping from LS1 to the router interface (RI) in LAN1
5. Capture the resulting trace in LS1 stations and confirm that the ICMP/IP packet from LS1 to RI travelled encapsulated in an Ethernet frame whose destination MAC is that of RI and whose source MAC is that of LS1

Now, we would like to check the *else* part of the IPFA mentioned above. Let’s devise another simple experiment similar to experiments 1 and 2, however, in this case, a single network will not suffice, obviously, if

we want to check how communications in the internetwork function we will have to design an internetwork.

## Understanding IP forwarding in the internetwork

We can design a simple internetwork by using the simple network building block developed in the previous section, let's start by gaining some insight on functions fulfilled by our simple IP router in fig. 4 next.

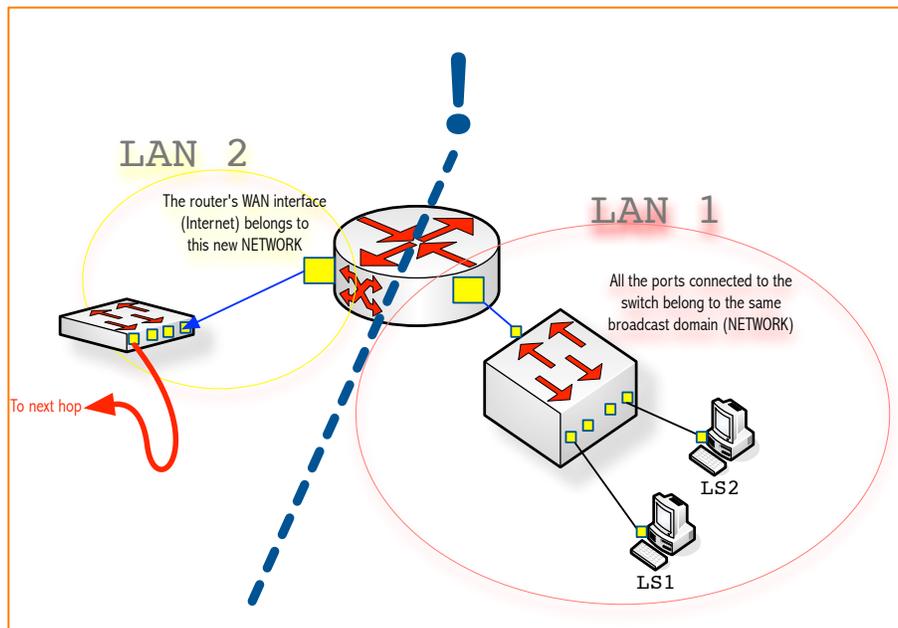


Fig. 4. A router with two NICs creates two networks

The IP router is a layer-3 equipment that interconnects networks, in the case of Fig. 4, the router connects two LANs (LAN1 and LAN2), each supported by a switch which permits the connection of further stations. Each of the networks *created* by the router is obviously a broadcast domain totally separated from others. If we connect stations to LAN2, they will communicate among themselves by using the procedure explained before, and those stations will need an IP address which network prefix will be the same of LAN2. We now want to extend this *network building block* so we turn it into an internetwork.

Focus your attention on the switch on the left, can you see that we can connect a brand new router to it (That is, its WAN interface -please, see legend of LAN2), exactly in the same manner as that of fig. 4, that way, we will obtain LAN3 and other LANs that might be needed. The following network diagram in Fig. 5 represents our testbed internetwork, which has been designed by applying the scheme just explained. In the real laboratory exercise, you will have a few more networks, though.

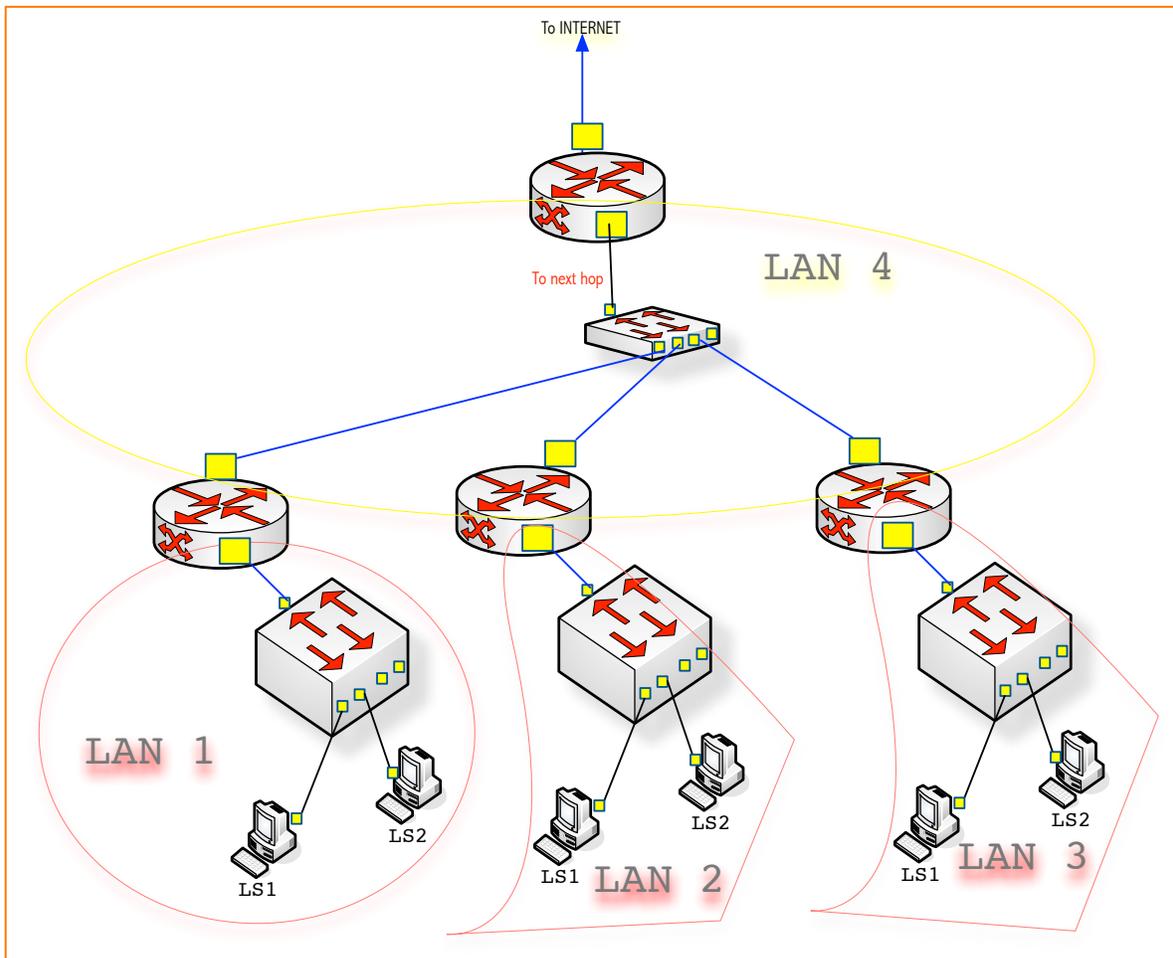


Fig. 5. A testbed internetwork

### Experiment 3: IPFA when (NetworkNum of Destination) $\neq$ (my NetworkNum)

This part of the IPFA states that when an IP host sends an IP packet, if the packet's destination IP address' prefix number is different from its network number (Or network prefix), then, that packet is encapsulated in an Ethernet frame destined to its default router. The router will receive the frame mentioned, will deencapsulate the IP packet, will analyze the its destination address and, looking up the forwarding table, it will send it to some other *directly connected* router or, in case the destination address' network prefix is one of its directly connected networks, will send the packet to the destination host encapsulated in an appropriately addressed Ethernet frame.

1. Set the IP configuration of your Cisco router according to Appendix 1 and your LAN number assigned (This number will serve as prefix for referencing the local stations in this document). Take into account that the real internetwork to be built may contain more LANs than appear on Fig. 5, nevertheless, all LANs in the diagram are equivalent to each other except that at the upper router aggregation point (LAN 4)
2. Connect one station (LS) to your router's Ethernet switch, it must use DHCP to fetch its IP configuration, connect another station (RS, in this case a Remote Station) to any other LAN of your choosing other than your own assigned LAN

3. Run Wireshark in both stations
4. Send a ping from your Local Station (LS) to your Remote Station (RS)
5. Capture the resulting trace in both stations and justify the source and destination MACs that are included in the ping frames from LS to RS (And back). You may have to take further traces at other probing points in the internetwork.

**Exercise 1.** Extend experiment 1 such that we can check that the Ethernet frame from LS1 to LS2 does not cross the router of your assigned LAN. Hint: For capturing *all* the outbound traffic from a router, you can, for example, connect its WAN port (**Internet**) to a HUB and connect to it your Remote Station and run Wireshark.

**Exercise 2.** We already explained that when the IP packet from LS is destined towards a station that belongs to a network prefix other than theirs (RS in this case), it proceeds by conveying it to its default router *in the hope* that it will manage to ultimately deliver the packet to its destination however complex the intervening internetwork is. Now you will have to check that this explanation holds, to that end, carefully analyze the traces obtained by Wireshark in LS and RS and try to compose an argument that supports or refutes that explanation.

## Appendix 1: Router configurations for each network

Aggregated space base address: 192.168.100.0/24

- ⇒ 24 bits of network prefix
- ⇒ 8 bits of IP addressing space
- ⇒ (24 + 8 = 32)

Number of subnets necessary:  $8 = 2^3$

- ⇒ 3 bits for subnet
- ⇒ IP addressing space = subnet space + host space
- ⇒ 8 bits = 3 bits + 5 bits

### **SUBNET # 0**

192.168.100.0/(24+3) = 192.168.100.0/27

Size:  $32 - 27 = 5$ , size =  $2^5 = 32$

Prefix number = 192.168.100.0

First usable address = 192.168.100.0 + 1 =

192.168.100.1

Last usable address = 192.168.100.0 + (Size - 2) =

192.168.100.30

Broadcast address = 192.168.100.0 + (Size - 1) =

192.168.100.31

### **SUBNET # 1**

Size:  $32 - 27 = 5$ , size =  $2^5 = 32$

Prefix number = previous broadcast + 1 = 192.168.100.32

First usable address = 192.168.100.32 + 1 =

192.168.100.33  
Last usable address = 192.168.100.32 + (Size - 2) =  
192.168.100.62  
Broadcast address = 192.168.100.32 + (Size - 1) =  
192.168.100.63

**SUBNET # 2**

Size: size =  $2^5 = 32$

Prefix number = previous broadcast + 1 = \_\_\_\_\_

First usable address = 192.168.100.\_\_\_ + 1 =

Last usable address = 192.168.100.\_\_\_ + (Size - 2) =

192.168.100.\_\_\_

Broadcast address = 192.168.100.\_\_\_ + (Size - 1) =

192.168.100.\_\_\_

**SUBNET # 3**

**SUBNET # 4**

**SUBNET # 5**

**SUBNET # 6**

**SUBNET # 7**

**SUBNET # 8**