TCP/IP Computer Networks

Laboratory guides — sessions 1

- *Linux network commands*

- *Session 1: Introduction and Revisions*

TCP/IP Networks (Fall 2014)
José Legatheaux Martins
João Magalhães

Dep. Informatics
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

Introduction and revisions

TCP/IP Computer Networks Lab 1

Goals

Learn how to diagnose the data-link and network layers of the computer.

Learn how to capture network traffic and to recognize different protocols.

Know how to measure transit time in different scenarios and compare it to the theoretical expected ones.

Report

Use this guide to take notes during the lab class. If asked by the instructor, write a report on your most relevant findings and try to explain them. The report should have around 5 pages (double spaced, 11 dots) and must be delivered in the class (lab session) that follows the last class (lab session).

Initial Inspection

Use application *ifconfig* to examine your PC network configuration.

Which interfaces exist in your PC? _____

What is the physical address of the active interface? _____

What is the network address of the active interface? _____

How many computers can exist in your network? _____

Web Access

Capture the network traffic generated by a Web access. You should observe several protocols at work (e.g., DNS, IP, TCP, HTTP). Then observe captured packets and explain the process sequentially.

A. Start the traffic capture application

```
tcpdump -w capture1.pcap -i eth?
```

B. Verify the route to reach the address www.abola.pt:

```
traceroute www.abola.pt
```

C. Establish an end-to-end connection with a WWW server on port 80 with the telnet command:

```
telnet www.abola.pt 80
```

D. On the telnet window execute the following command:

```
GET / HTTP/1.0 (press Return twice)
```

E. Stop the capturing traffic and examine the output with Wireshark

```
wireshark capture1.pcap
```

apply the following *Display Filter*:

```
bootp or arp or dns or http
```

Detail the observed process by identifying **all network elements** and **network protocols** ARP, TCP, DNS, e HTTP.

Setting-up a Network

Setup a network as shown in figure (see last page). Each computer has two Ethernet interfaces: (1) one wireless connected to the wireless network (with dynamic addresses sharing some prefix and setup by DHCP) and (2) the other in the benches wired network. The second is accessed via the wired Ethernet interface with name eth? or en1 …. (use ifconfig to discover it).

Manual Network Address

Use command ifconfig to assign an IP address to **interface connected to the bench hubs or switches**:

```
ifconfig eth? 192.168.1.? netmask 255.255.255.0 broadcast 192.168.1.255
```

at the end of the document you will find the required addresses and other parameters.

ARP Tables

Command arp will allow you to look into the computers ARP tables. ARP tables also record interface identifiers associated with IP addresses. Is it strictly necessary?

Observe exchanged ARP messages and analyze the contents of ARP frames. Print an ARP exchange and explain what is going on. Do the same with the ICMP traffic generated by ping.

Routing Tables: Network Masks

(This exercise will require the cooperation of several student groups)

Change the north side computers netmasks interfaces to 255.255.255.240 (/28). Not all machines will be able to communicate any more. Explain why.

Routing Tables: Network Addresses

(This exercise will require the cooperation of several student groups)

Examine the routing table of your PC:

```
route -n
```

What is the network address of your Default Gateway? _____

Use addresses with the IP prefix 192.168.0.0/24 in the north side benches computers, addresses with the IP prefix 192.168.1.0/24 in the south side benches computers and addresses with the IP prefix 192.168.2.0/24 in the center benches computers. Try to ping machines with one prefix from machines with the same or a different prefix. Start capturing traffic and observe the exchanged packets. Explain what you have observed.

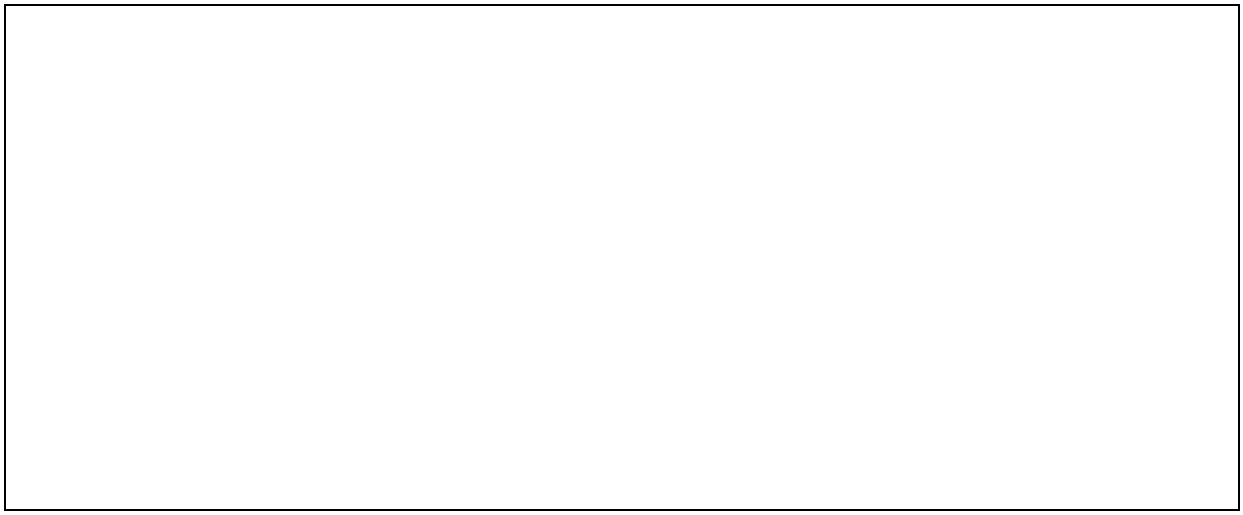
Suggest some method to allow computers with interfaces using addresses of different IP prefixes to communicate directly through the lab hubs / switches interfaces.

Measuring the Network Latency

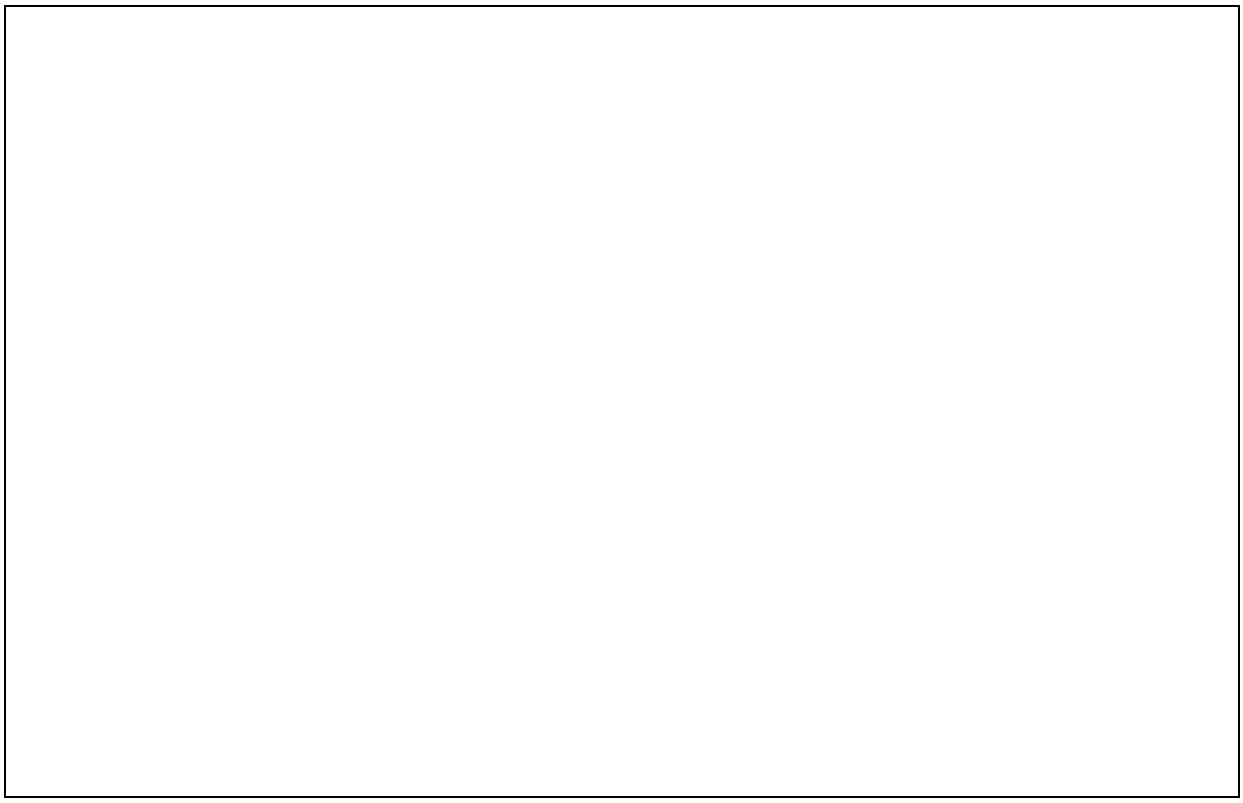
Use program ping to measure latency when computers communicate through both networks (wireless and benches networks) with and without background traffic. Run command

```
ping –s SIZE
```

to change the size of the ICMP packets used. Heavy background traffic can be easily generated by pinging broadcast addresses with option –f (flood). Compute analytically transit time in the different scenarios and compare these values with the measured ones (only in the light traffic scenario).



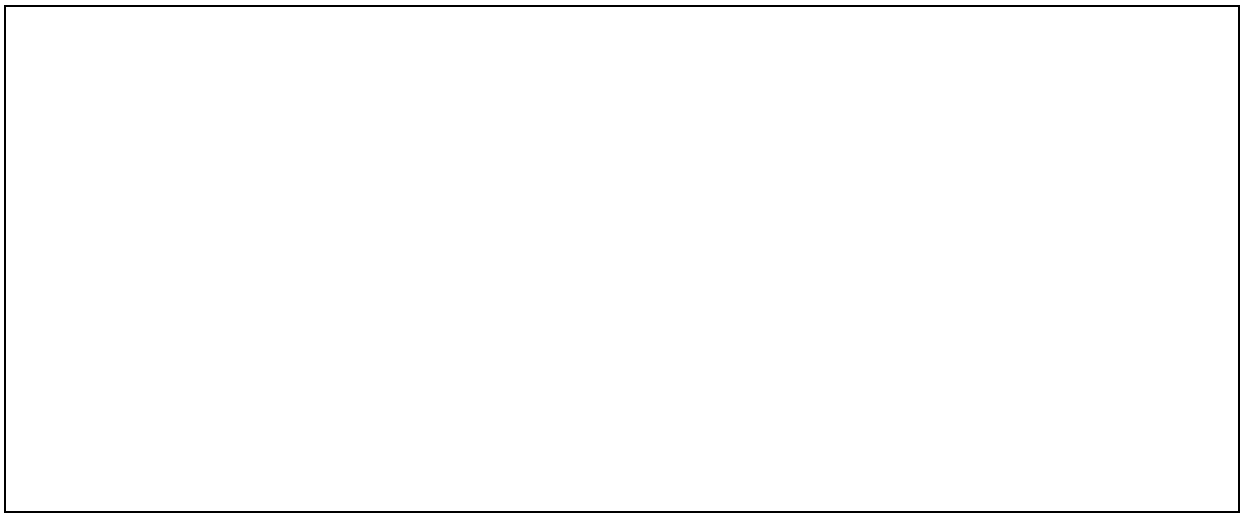
Try to figure out where packets queues will be located in the two scenarios (wireless network and benches network).



Measuring the Network End-to-End Capacity

Use program iperf to measure network end-to-end performance when computers communicate through both networks (wireless and lab networks) with and without background traffic (in the case of the wireless network, using broadcast background traffic will collapse the network, do not try it!). Run the following commands in two different computers, using the hubs network and the switches network:

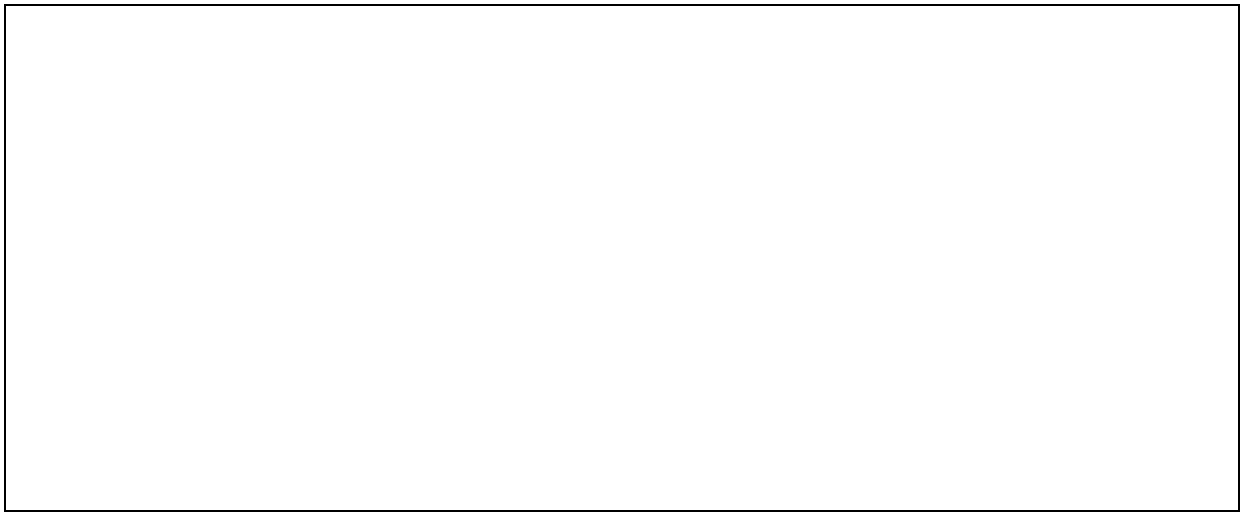
```
iperf –s  -i 5 –t 360       // TCP server
iperf –c 192.168.1.?  -i 5 –t 360       // TCP client; replace
                                         // ? with the address of the server
```



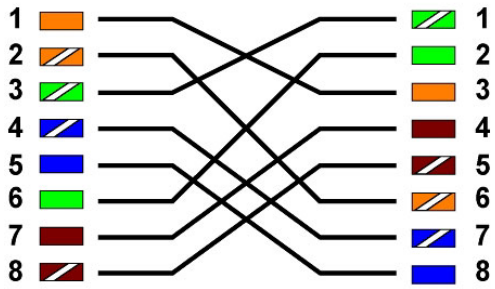
Do the same exercise using UDP:

```
iperf –s  -u -i 5 –t 360       // UDP server
iperf –c 192.168.1.? -u -i 5 –t 360  -b 10m
       // UDP client; replace ? with the address of the server
       // -b 10m stands for 10 Mbps maximum packet rate
```

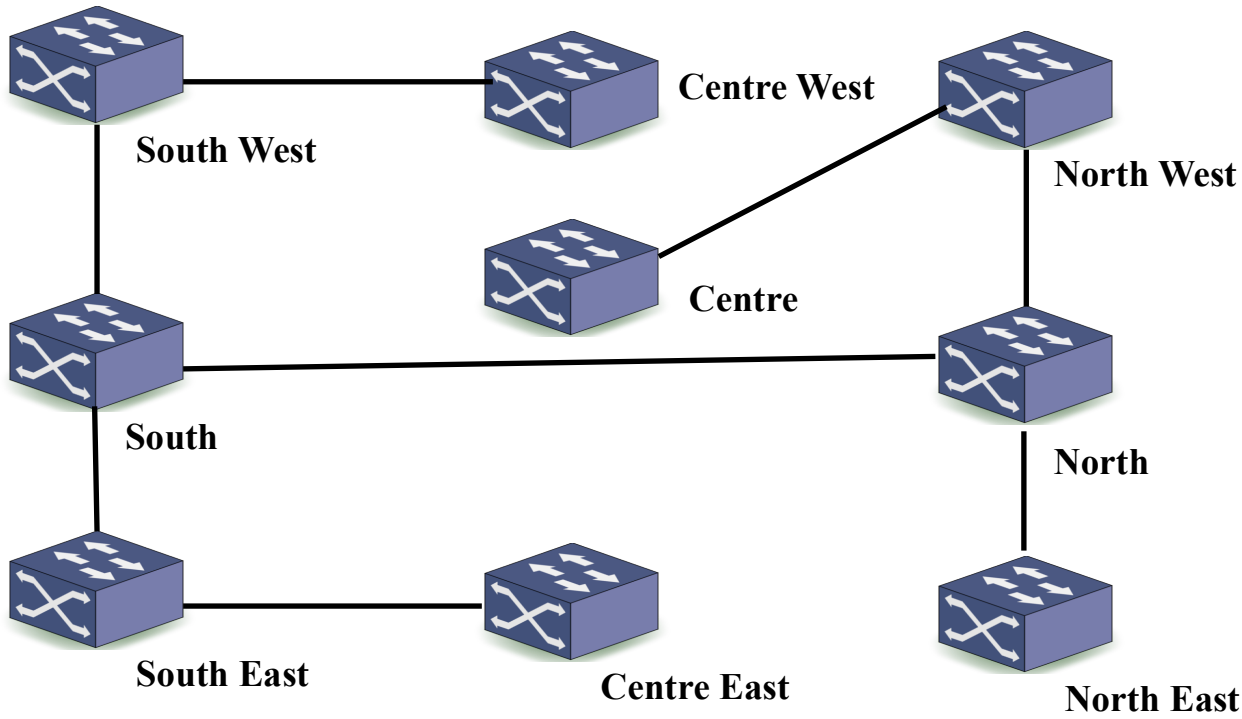
Report what you can observe.



Ethernet Cables

RJ-45 Straight Cable	RJ45 Crossover Cable
	

Network Diagrams



IP addresses

SW: 192.168.1.40, 41, 42, … /24	**CW: 192.168.1.90, 91, 92, …/24**	**NW: 192.168.1.10,11, 12, … /24**
S: 192.168.1.50, 51, 52, … /24	**C: 192.168.1.70, 71, 72, …/24**	**N: 192.168.1.20,21, 22, … /24**
SE: 192.168.1.60, 61, 62, … /24	**CW: 192.168.1.80, 81, 82, …/24**	**NE: 192.168.1.30,31, 32, … /24**

Linux network commands

arp

```
arp  –a          test:  -d delete –s inserts
```

arping

```
arping
```

dig

```
dig ns.di.fct.unl.pt
```

wireshark

```
wireshark –r traballho1.3.txt &
```

ifconfig

```
ifconfig eth1 192.168.0.11 [netmask 255.255.255.0]
ifconfig eth1 netmask 255.255.255.240
ifconfig -a
ifconfig eth1 down
```

iperf

```
iperf –s [ -w 130k ] [ -i 5 ]
iperf –c 192.168.0.11  [ -w 130k ] [ -t 120 ]

iperf –s –u [-l 60k ] [-i 5 ]
iperf –c 192.168.0.11 –u [-l 60k] [-t 120][-b 10m ]
```

minicom

Start with: **minicom –s**
and then: **minicom**

modprobe

```
modprobe 3c59x  (to activate a 3COM 3c59x Ethernet controller)
```

netstat

```
netstat –an

netstat –i eth0
netstat –r
netstat –g ....
netstat –s ....
```

ping

```
ping –n [ -f ]  [ -s packet-size ] ip-address
```

The packet sent has packet-size + 8 + 20 + 20 bytes approximately. By default 56 + 8 + 20 + 20 = 104 bytes

route

```
route –n
route add default gw 10.200.0.1
route delete default gw 10.200.0.1
route add –net 192.168.0.0/16 gw 192.168.100.1
route add –net 224.0.0.0/8 dev eth1
```

screen

```
screen /dev/uc.usb      (is a modern replacement of minicom)
```

script

```
script –a trabalho1.txt
```

ssh

```
ssh host-name  [ ou  ip-address ]
```

traceroute

```
traceroute –n ip-address
```

tcpdump

```
tcpdump –enx –w traballho1.3.txt
tcpdump host ipaddr
tcpdump –i eth1 igmp
tcpdump –i eth1 ip multicast
tcpdump –i eth1 tcp
tcpdump –i eth1 udp
tcpdump host ipaddr
tcpdump host ipaddr-1 and ipaddr-2
tcpdump udp port 53
tcpdump –n –nn ip-addr-1 and ipaddr-2
tcpdump –x –s 120 ip proto 89
tcpdump –x –s 70 host ip-addr-1 and ( ipaddr-2 or
            ipaddr-3)
tcpdump –x –s 70 host ip-addr-1 and  not ipaddr-2
```

ttcp

```
ttcp –s [t|r] v [u] –f m –l 1500 –n 5000 [ 8000 ]
       [ receiver-ip-addr ]
```