Fondamenti di Internet: *“Aspetti pratici delle reti di telecomunicazioni ”*

University of Rome "Tor Vergata"
Department of Electronics Engineering
NETWORKING GROUP

http://netgroup.uniroma2.it

Donato Battaglino
donato.battaglino [at] uniroma2.it

Lorenzo Bracciale
lorenzo.bracciale [at] uniroma2.it

First Part

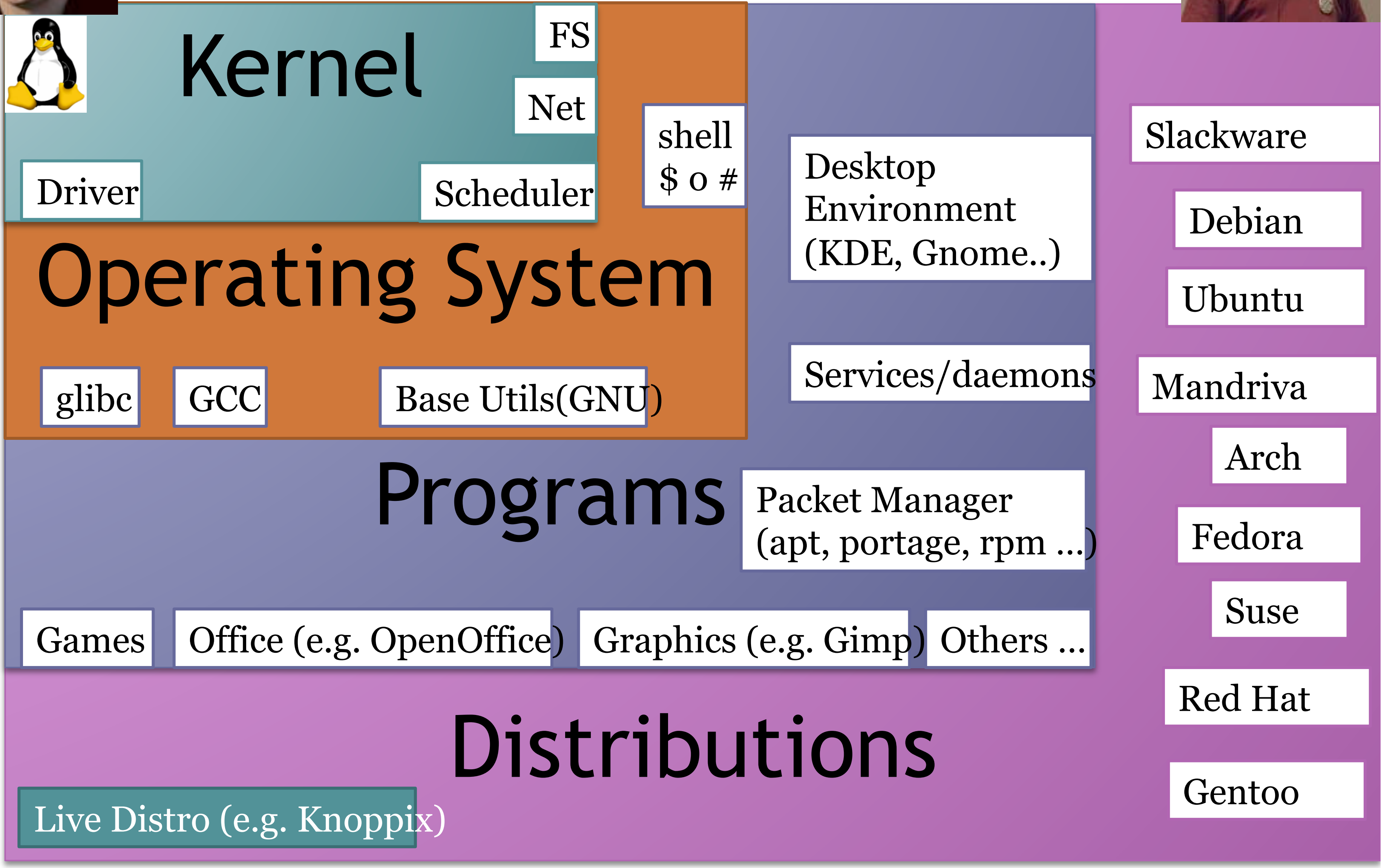
Outline

- First part:
  - ✓ Linux for dummies
  - ✓ Emulation of computer networks: Netkit
- Second part:
  - ✓ Configuration of network interfaces: IProute2
  - ✓ Analyzing of network traffic: Wireshark & Tcpdump
  - ✓ Examples of network traffic: Scapy, Ping

Why Linux ?

- Open Source Philosophy: no secret for the user
- Widely used in networks
  - ✓Routers
  - ✓Embedded systems
  - ✓Servers
- Free distribution
- Large support and documentation
- …and “User Mode Linux”

Linux in a nutshell



Kernel

FS

Net

Driver

Scheduler

shell
$ o #

Operating System

glibc

GCC

Base Utils(GNU)

Desktop Environment (KDE, Gnome..)

Services/daemons

Programs

Packet Manager (apt, portage, rpm ...)

Games

Office (e.g. OpenOffice)

Graphics (e.g. Gimp)

Others ...

Distributions

Live Distro (e.g. Knoppix)

Slackware

Debian

Ubuntu

Mandriva

Arch

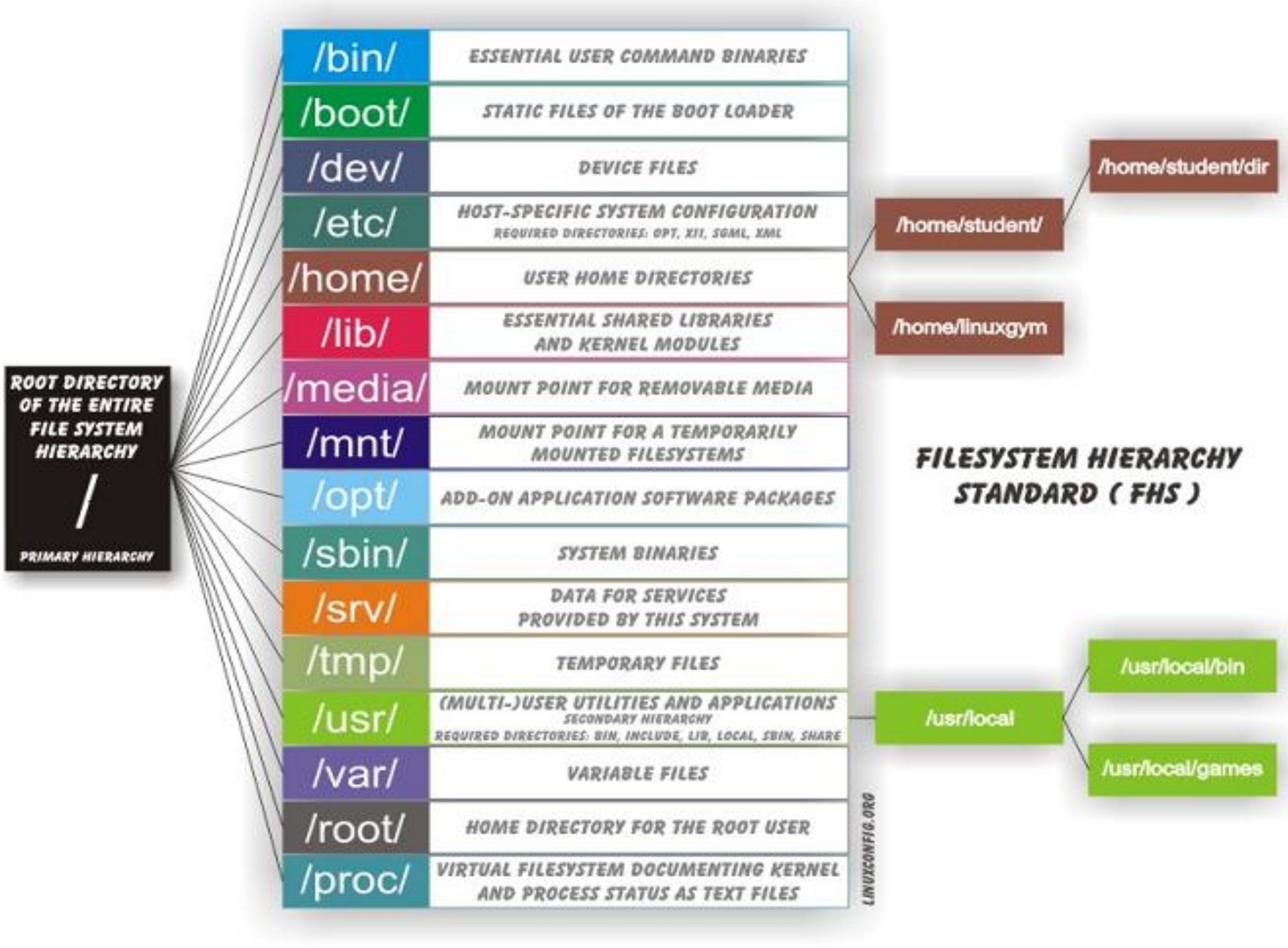
Fedora

Suse

Red Hat

Gentoo

Linux (typical) directory tree



/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

Usefull shell commands

ls	“List” file/directory contained in the current directory.Usefull options: “–al”
cd <DIRECTORY>	Change current Directory . Usefull: “cd ..”
mkdir <NEW_DIRECTORY>	Create a new directory
rm <FILE>	Remove file. Usefull options “-rf” to remove also directory
touch <FILE>	Create a new, empty, file
nano <FILE>	edit a file (or create if it does not exist)
cp <ORIG> <DEST>	copy a file. Usefull options “-r” to copy also directory
mv <ORIG> <DEST>	move or rename a file/directory

Support

The Magic Word

man

Documentation

Linux User Guide

http://www.pluto.it/files/ildp/guide/GuidaUtente/index.html

Distro Specific guides (e.g. http://help.ubuntu-it.org/)

Linux User Group

http://www.torlug.org

Download the netkit live distribution:

http://wiki.netkit.org/index.php/Download_Official

Linux and Networking

User Space

System calls:Socket

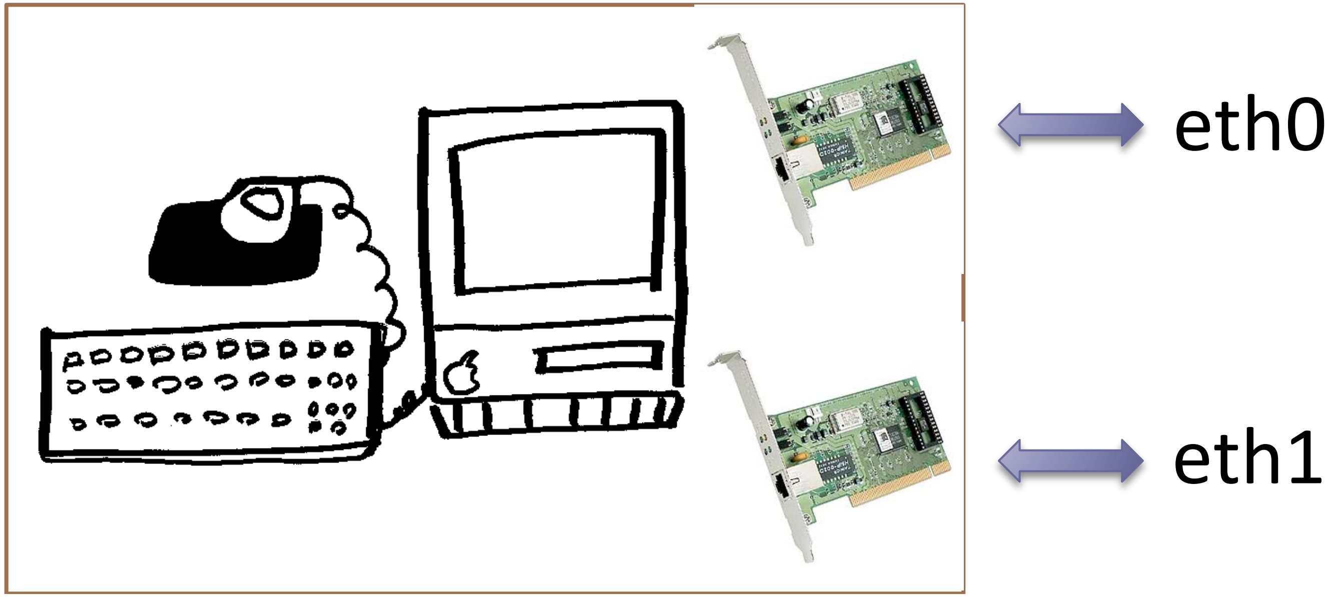
Kernel Space

Drivers

Hardware

Application
Transport
Network
Data Link
Physical

Interface Mapping



Outline

- First part:
  - ✓ Linux for dummies
  - ✓ Emulation of computer networks: Netkit
- Second part:
  - ✓ Configuration of network interfaces: IProute2
  - ✓ Analyzing of network traffic: Wireshark & Tcpdump
  - ✓ Examples of network traffic: Scapy, Ping

Simulation vs. Emulation

Emulation and simulation systems put at user's disposal a virtual environment that can be exploited for tests, experiments, measures

SIMULATION

simulation systems aim at reproducing the performanceof a real-life system (latency time, packet loss, etc.)

- e.g.: ns, omnet++, …

EMULATION

emulation systems aim at accurately reproducing the functionalitiesof a real-life system (configurations, architectures, protocols), with limited attention to performance

References: http://www.netkit.org/netkit-labs/netkit_introduction/netkit-introduction.pdf

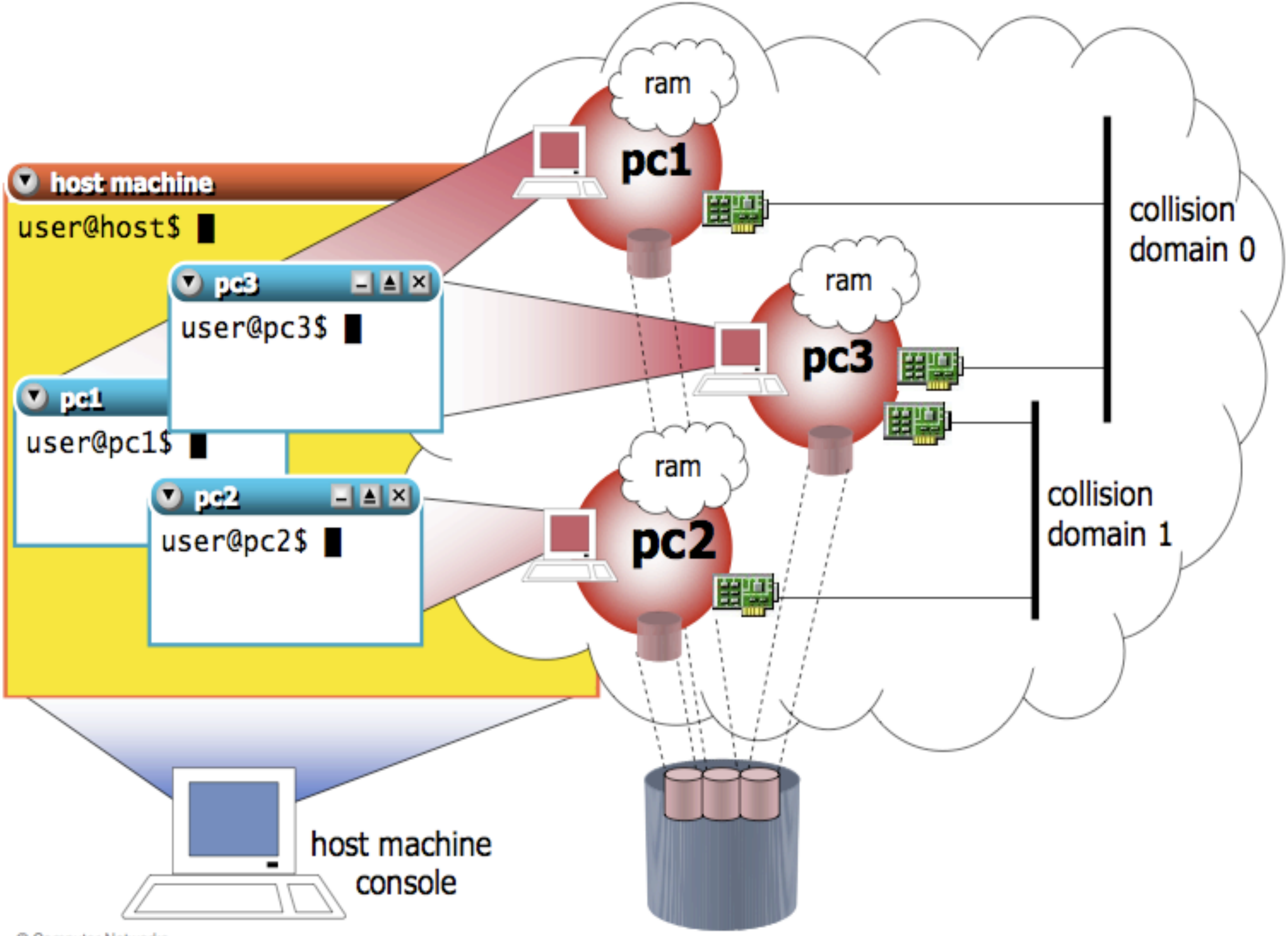
Netkit

wiki.netkit.org

”The poor man's system to experiment computer networking”

WHAT’S NETKIT:

- a system for emulating computer networks
- based on uml (user-mode linux)
  - ✓ user-mode linux is a linux kernel (inner part of the linux os) that can
    be executed as a user process on a standard linux box
  - ✓ a user-mode linux process is also called virtual machine (vm), while the
    linux box that hosts a virtual machine is called host machine (host)
- each emulated network device is a virtual linux box
- note: the linux os is shipped with software supporting most of the network protocols
  - ✓ hence, any linux machine can be configured to act as a
    bridge/switch or as a router



```
user@host$ █
```

```
user@pc3$ █
```

```
user@pc1$ █
```

```
user@pc2$ █
```

Netkit Virtual Machine Commands

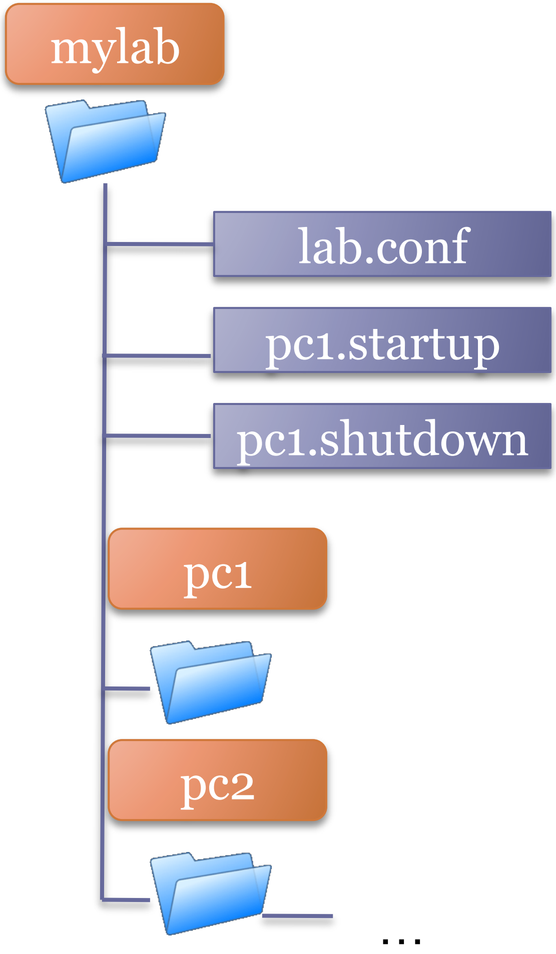
vstart: starts a new virtual machine
vlist: lists currently running virtual machines
vconfig: attaches network interfaces to running vms
vhalt: gracefully halts a virtual machine
vcrash: causes a virtual machine to crash
vclean: “panic command”to clean up all netkit processes (including vms) and configuration settings on the host machine

More Info about netkit commands? **man vstart** or visit *wiki.netkit.org*

Setting-up a Netkit Lab

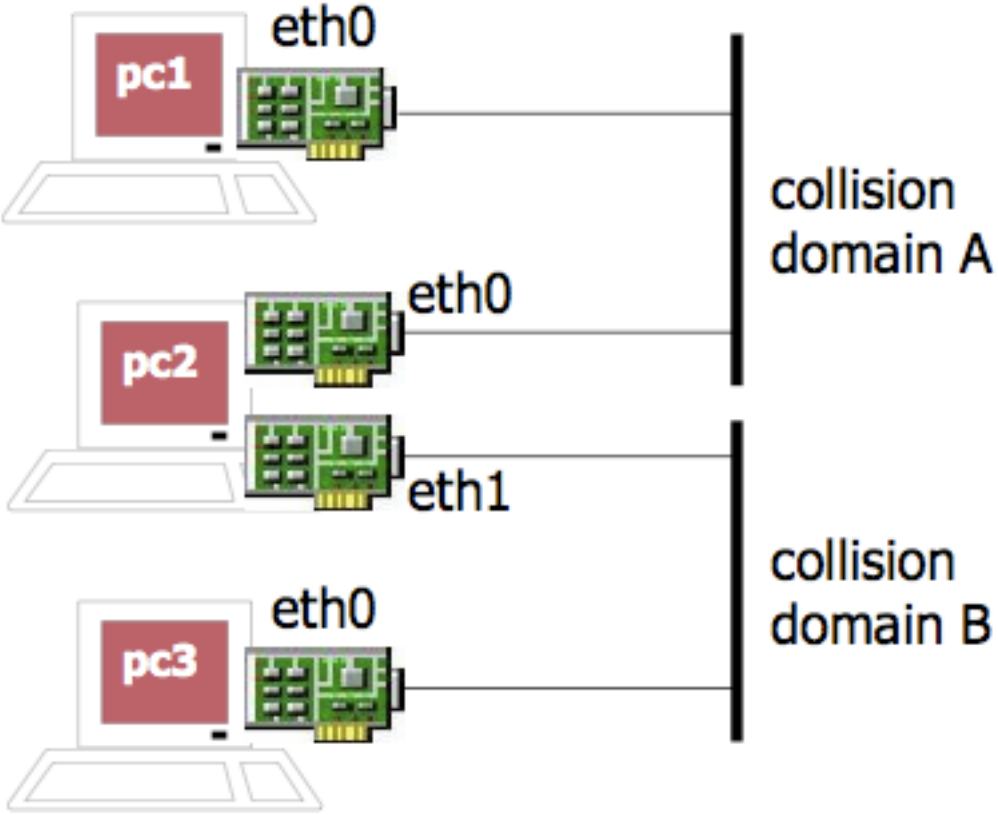
Netkit Lab: automatize multiple virtual machine startup. To create a lab we need:

- a lab configuration file describing the network topology (lab.conf)
- a set of subdirectories that contain the configuration settings for each virtual machine
- [optionally] .startup and .shutdown files that describe actions performed by virtual machines when they are started or halted
- [optionally] a lab.depfile describing dependency relationships on the startup order of virtual machines



Example of lab file: mylab.conf

```
pc1[0]=A
pc2[0]=A
pc2[1]=B
pc2[mem]=256
pc3[0]=B
```



Netkit Lab Commands

lstart: starts a netkit lab
lhalt: gracefully halts all vms of a lab
lcrash: causes all the vms of a lab to crash
lclean: removes temporary files from a lab directory
linfo: provides information about a lab without starting it
ltest: allows to run tests to check that the lab is working properly

As for the case of virtual machine, for help type **man lstart** or visit *wiki.netkit.org*

File Exchange and Internet Connection

the directory **/hosthome** inside a virtual machine directly points to the home directory of the current user on the real host

The directory **/hostlab** is shared inside a lab

vstart can automatically configure tunnels (“tap interfaces”) by which a virtual machine can access an external network

File .startup

1. For each VM is possible to create a “VM_NAME”.startup file
2. VM_NAME is the name of the virtual machine
   Example: pc1 → pc1.startup
3. The commands written in the .startup file will be exectued at the end of the boot precess

TAP interface

1. A VM interface can be attached to a “tap” instead of a collidsion domain
2. In this case, a virtual (tap) interface will be created on the host machine and will be connected to the specific VM interface

Example (in lab.conf)

```
pc1[0]=tap,10.0.0.1,10.0.0.2
```

In this case the eth0 interface on pc1 will be connected to a tap on the host machine with address 10.0.0.1 while eth0 on pc1 will be configured automatically with address 10.0.0.2

Second Part

Outline

- First part:
  - ✓ Linux for dummies
  - ✓ Emulation of computer networks: Netkit
- Second part:
  - ✓ Configuration of network interfaces: IProute2
  - ✓ Analyzing of network traffic: Wireshark & Tcpdump
  - ✓ Examples of network traffic: Scapy, Ping

IPRoute2

Iproute2 is a collection of utilities for controlling TCP / IP networking and traffic control in Linux. [1]

Implemented on almost any linux system with kernel > 2.2.X

Mangle almost any network stuff:

- Interface setting up and drop down
- Routing Tables
- Arp cache
- And more…
  - Multiple routing tables
  - Policy Routing
  - Tunnels
  - NAT
  - IPsec policy and associations

Replace “old” (but still used) commands: route, arp, ifconfig

[1] http://www.linuxfoundation.org/en/Net:Iproute2

Simple and basic guide: http://www.commedia.it/ccontavalli/docs-it/ip/ip4dummies.pdf

IProute2: data link settings of a network interface

IProute2 command to manage data link settings

ip link <command>

how to? # ip link help

```
ip link help
Usage: ip link set DEVICE { up | down |
                         arp { on | off } |
                         dynamic { on | off } |
                         multicast { on | off } |
                         allmulticast { on | off } |
                         promisc { on | off } |
                         trailers { on | off } |
                         txqueuelen PACKETS |
                         name NEWNAME |
                         address LLADDR | broadcast LLADDR |
                         mtu MTU }
       ip link show [ DEVICE ]
```

IProute2: ip link <…>, a basic usage

In a UNIX-like system, a network interface can have different names. Under Linux-based operating systems: eth0, eth1, eth2, etc. Generally **ethX**

Activating a network interface

```
# ip link set eth0 up
```

De-activating a network interface

```
# ip link set eth0 down
```

Changing the data link address (mac address)

```
# ip link set eth0 address xx:xx:xx:xx:xx:xx
```

Showing data link attributes of system network interfaces

```
# ip link show
```

IProute2: network (IP) settings of a network interface

IProute2 command to manage network (IP, Internet Protocol) settings

```
# ip address <command>
```

how to? # ip address help

```
ip address help
Usage: ip addr {add|del} IFADDR dev STRING
       ip addr {show|flush} [ dev STRING ] [ scope SCOPE-ID ]
                            [ to PREFIX ] [ FLAG-LIST ] [ label PATTERN ]
IFADDR := PREFIX | ADDR peer PREFIX
          [ broadcast ADDR ] [ anycast ADDR ]
          [ label STRING ] [ scope SCOPE-ID ]
SCOPE-ID := [ host | link | global | NUMBER ]
FLAG-LIST := [ FLAG-LIST ] FLAG
FLAG  := [ permanent | dynamic | secondary | primary |
           tentative | deprecated ]
```

We will manage it better in the next lectures

Outline

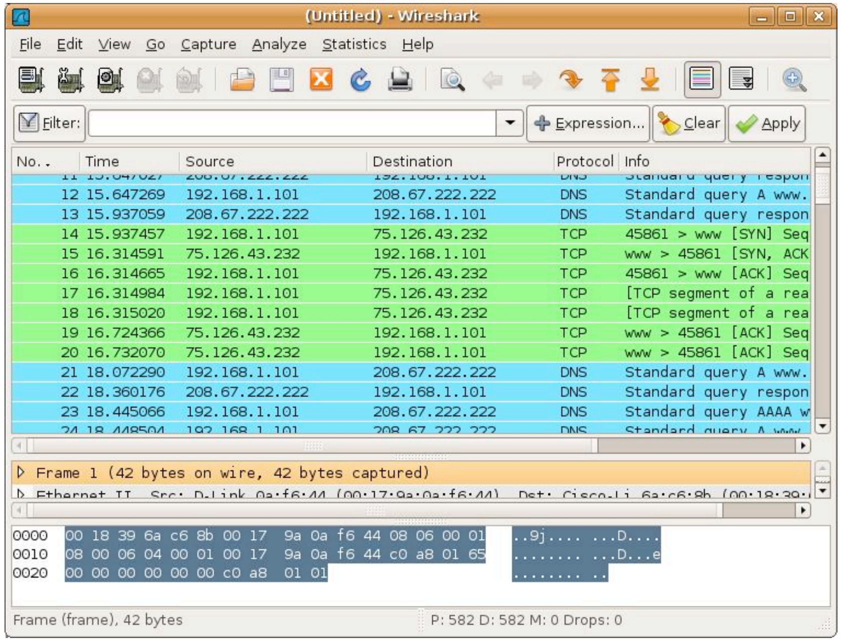
- First part:
  - ✓ Linux for dummies
  - ✓ Emulation of computer networks: Netkit
- Second part:
  - ✓ Configuration of network interfaces: IProute2
  - ✓ Analyzing of network traffic: Wireshark & Tcpdump
  - ✓ Examples of network traffic: Scapy, Ping

Wireshark: “the” network analyzer

www.wireshark.org

Wireshark is the world's foremost network protocol analyzer, and is the de facto (and often de jure) standard across many industries and educational institutions.

Wireshark development thrives thanks to the contributions of networking experts across the globe. It is the continuation of a project that started in 1998 (Ethereal).



No.	Time	Source	Destination	Protocol	Info
11	15.647027	208.67.222.222	192.168.1.101	DNS	Standard query respon
12	15.647269	192.168.1.101	208.67.222.222	DNS	Standard query A www.
13	15.937059	208.67.222.222	192.168.1.101	DNS	Standard query respon
14	15.937457	192.168.1.101	75.126.43.232	TCP	45861 > www [SYN] Seq
15	16.314591	75.126.43.232	192.168.1.101	TCP	www > 45861 [SYN, ACK
16	16.314665	192.168.1.101	75.126.43.232	TCP	45861 > www [ACK] Seq
17	16.314984	192.168.1.101	75.126.43.232	TCP	[TCP segment of a rea
18	16.315020	192.168.1.101	75.126.43.232	TCP	[TCP segment of a rea
19	16.724366	75.126.43.232	192.168.1.101	TCP	www > 45861 [ACK] Seq
20	16.732070	75.126.43.232	192.168.1.101	TCP	www > 45861 [ACK] Seq
21	18.072290	192.168.1.101	208.67.222.222	DNS	Standard query A www.
22	18.360176	208.67.222.222	192.168.1.101	DNS	Standard query respon
23	18.445066	192.168.1.101	208.67.222.222	DNS	Standard query AAAA w
24	18.448504	192.168.1.101	208.67.222.222	DNS	Standard query A www.

```
▷ Frame 1 (42 bytes on wire, 42 bytes captured)
▷ Ethernet II, Src: D-Link_0a:f6:44 (00:17:9a:0a:f6:44), Dst: Cisco-Li_6a:c6:8b (00:18:39:
```

```
0000  00 18 39 6a c6 8b 00 17  9a 0a f6 44 08 06 00 01   ..9j.... ...D....
0010  08 00 06 04 00 01 00 17  9a 0a f6 44 c0 a8 01 65   ........ ...D...e
0020  00 00 00 00 00 00 c0 a8  01 01                     ........ ..
```

Frame (frame), 42 bytes

P: 582 D: 582 M: 0 Drops: 0

* Deep inspection of hundreds of protocols, with more being added all the time

* Live capture and offline analysis

* Standard three-pane packet browser

* Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others

* Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility

* The most powerful display filters in the industry

* Rich VoIP analysis

* Read/write many different capture file formats

* Capture files compressed with gzip can be decompressed on the fly

* Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB,
,Token Ring, Frame Relay, FDDI, and others (depending on your platfrom)

* Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2 (having the keys ☺)

* Coloring rules can be applied to the packet list for quick, intuitive analysis

* Output can be exported to XML, PostScript®, CSV, or plain text

Tcpdump: command line network analyzer

tcpdump/libpcap

```
tcpdump [ -adeflnNOpqRStuvxX ] [ -c count ]
              [ -C file_size ] [ -F file ]
              [ -i interface ] [ -m module ] [ -r file ]
              [ -s snaplen ] [ -T type ] [ -w file ]
              [ -E algo:secret ] [ expression ]
```

man tcpdump

Why using tcpdump? Netkit hasn't a graphical environment, using tcpdump to capture and Wireshark to display pkts

Outline

- First part:
  - ✓ Linux for dummies
  - ✓ Emulation of computer networks: Netkit
- Second part:
  - ✓ Configuration of network interfaces: IProute2
  - ✓ Analyzing of network traffic: Wireshark & Tcpdump
  - ✓ Examples of network traffic: Scapy, Ping

COMMUNICATING

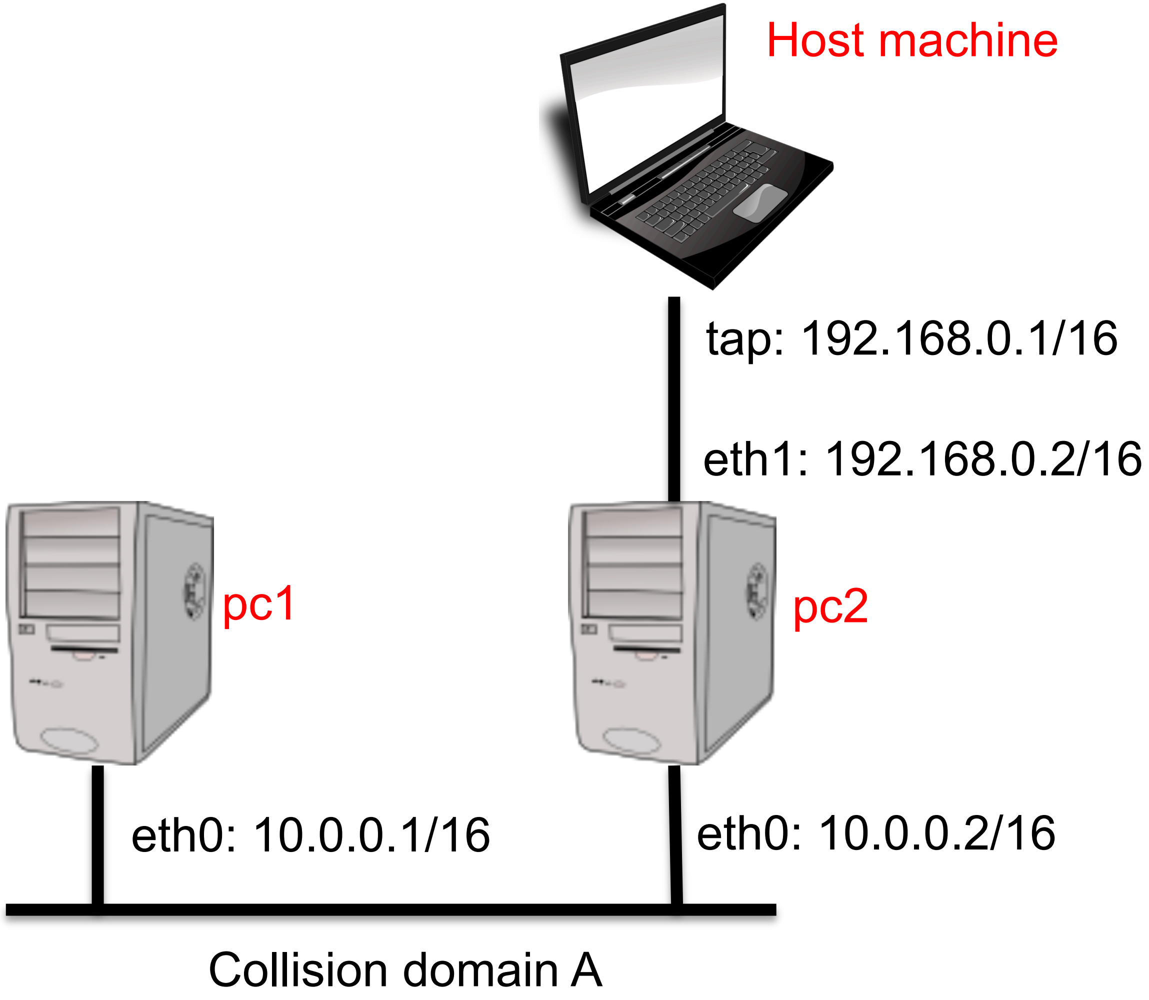
- What is SCAPY? Scapy is a powerful interactive packet manipulation program. It is able to forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies, and much more.

http://www.secdev.org/projects/scapy/

-What is PING? Ping is a computer network tool used to test whether a particular host is reachable across an IP network; it is also used to self test the network interface card of the computer, or as a speed test. It works by sending ICMP “echo request” packets to the target host and listening for ICMP “echo response” replies.

man ping ☺

Example: Lab2



Host machine

tap: 192.168.0.1/16

eth1: 192.168.0.2/16

pc1

pc2

eth0: 10.0.0.1/16

eth0: 10.0.0.2/16

Collision domain A