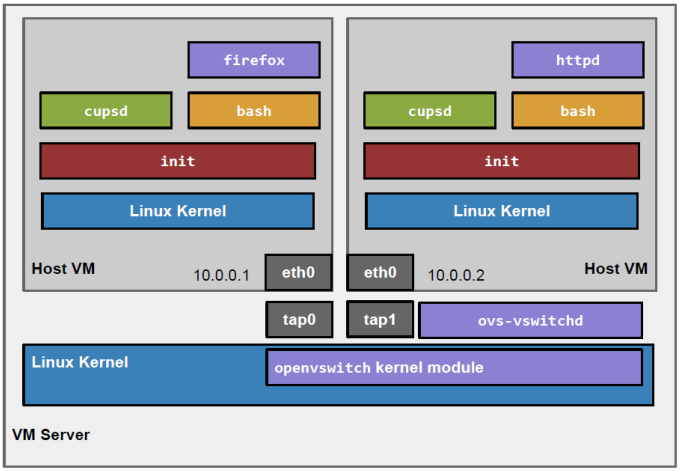
# Labo computernetwerken I – IP routing

Hosts kunnen wereldwijd met elkaar verbonden worden, ook al zitten ze niet in eenzelfde (local area) netwerk. Deze verbinding-van-netwerken gebeurt door het Internet Protocol – IP. Elk (local area) netwerk krijgt een unieke range van adressen, elke host binnen dit netwerk een (wereldwijd) uniek IP-adres. De verbinding tussen de netwerken gebeurt door middel van een router.

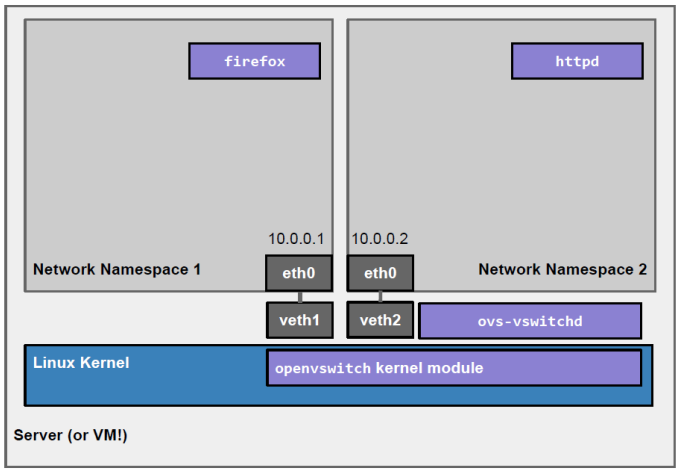
De organisatie van IP-ranges voor enkele kleine (bedrijfs)netwerken werken we uit in dit labo. Eveneens bestuderen we hoe IP-adressen op hosts ingesteld worden, en op routers. Het bijwerken van de routeringstabel is de volgende logische stap.

## Emulatie: mininet

We kunnen meerdere hosts en een (virtuele) router organiseren door meerdere VMs aan elkaar te schakelen in VMware of Virtualbox. Het nadeel van een dergelijke opstelling is de overhead die ontstaat op de eigen hardware: elke VM heeft een eigen hard disk, een eigen operating system nodig[1]:




Om een netwerk te kunnen emuleren, is al deze overhead an sich niet nodig: wat we willen, is een omgeving waarin verschillende netwerkkaarten kunnen bestaan op een toestel, en waar de nodige applicatielaag software actief kan zijn:




Mininet[2] is een voorbeeld van een dergelijke emulatiesoftware. Het kan op een Linux omgeving geïnstalleerd worden als package. Opdat iedereen dezelfde omgeving zou hebben, installeren we deze software in een VM – en daarbinnen heeft iedereen dan verschillende geëmuleerde hosts.
De VM zelf maakt geen deel uit van de set-up.

[1] Afbeeldingen van SigComm 2014, https://conferences.sigcomm.org/sigcomm/2014/
[2] http://mininet.org/

## Opstarten emulatieomgeving

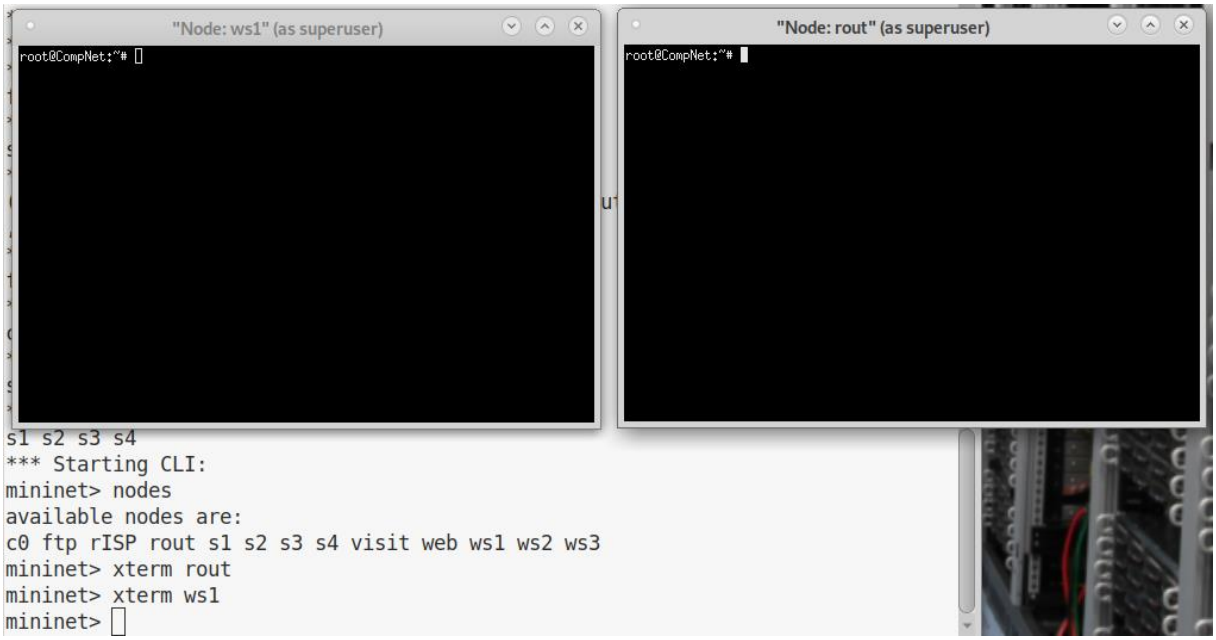
In dit labo emuleren we een volledig netwerk m.b.v. mininet. Deze software is reeds geïnstalleerd op jouw Ubuntu client VM[3]. Je kan de emulatieomgeving downloaden met **`wget`**, en opstarten met **`python`** :

```
student@linuxvm:~$ sudo –s ## (werk als root)
root@linuxvm:~# wget http://157.193.215.171/cnet_lab_IProuting.py
root@linuxvm:~# python cnet_lab_IProuting.py
```

Als de emulatieomgeving is opgestart, zie je in mininet alle ingeladen nodes:

```
*** Creating network
*** Adding controller
*** Adding hosts:
ftp rout visit web ws1 ws2 ws3
[ … ]
*** Waiting for switches to connect
s1 s2 s3 s4
*** Starting CLI:
mininet> nodes
available nodes are:
c0 ftp rISP rout s1 s2 s3 s4 visit web ws1 ws2 ws3
mininet>
```

Vanuit mininet kan je vervolgens een CLI voor elke node opstarten met **`xterm`**:




In deze xterm-cli omgevingen gaan we aan de slag voor deze labo-opgave

## Your address range

Voor dit labo werk je een configuratieverslag (in markdown of pdf) uit – er is geen test volgende week.

Afhankelijk van het nummer van jouw break-out room, start je jouw verslag met jouw eigen IP-adressen (geïndividualiseerd per student). Je downloadt jouw template (op je eigen laptop) via

`http://157.193.215.171/cnet/cnet1rout<XXXX>.md`

waar je <XXXX> vervangt door jouw studentenkaartnummer (e.g. 01605595).

[3] In een Debian based VM installeer je de software met, indien nodig:

```
student@linuxvm:~$ sudo apt install mininet
```

# Introduction

A small company wants to connect to the Internet, both to allow employees to access information on the Internet and to provide customers with information through a company website and FTP-server.

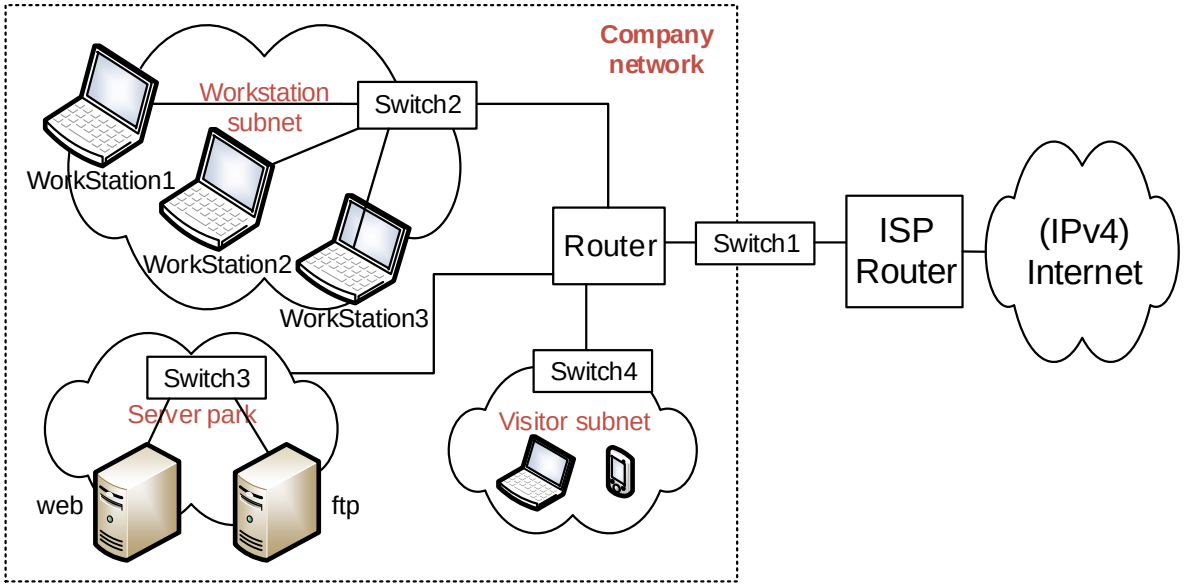



Figure 1: *Situation sketch*

The company has decided that in an initial stage, only three workstations will be connected, together with a newly acquired FTP server and webserver. Furthermore, a separate subnetwork for visitors should allow to connect a maximum of 12 laptop users. The IP-address of one of these nodes is known in advance.
For the physical connection to the Internet ("Uplink"), the company is allowed to send traffic directly to the ISP router.

# 1 Network design

The first task, before setting up the network, is the address space design. The different components will be discussed briefly:

**Workstations** The workstations are Linux based nodes, labelled ws1 to ws3. The IP adresses are already determined.

**Servers** The two servers are also Linux machines. Again, the IP addresses cannot be altered.

**Visitors** The IP-address of one of these machines is known in advance. The network should allow 12 IP host addresses.

**Router** The router is again a Linux machine. The uplink to the ISP has a fixed address; the other addresses are your calculation work.

**ISP Router** This router belongs to the ISP, the only thing given is the IP address of the interface you are connecting to. The ISP router is fully configured to reach the Internet.

### Tasks : design

The router has multiple interfaces: **`rout-eth1`** is connected to switch1; **`rout-eth2`** to switch2 ...etc. For each of the interfaces, you will need to work out the corresponding subnetwork.

The IP addresses for the hosts & servers in the network have been pre-defined; you can find them in your report template. First build up your network design, as you have done in the IP exercises.

*Please note*: When choosing IP addresses and netmasks in your design, follow these two rules:

1. Always choose the smallest subnets possible (e.g. what is the smallest network possible for a point-to-point connection to the ISP router?)?
2. Router interfaces should be assigned the lowest address in the picked subnet.

- Design the topology of your entire network - indicate how you will interconnect the components so that all workstations and servers are connected to the Internet.
  Calculate all details of your subnetworks (network address, broadcast, ... ).
  Build up an overview of your subnets in your lab report.

## 2 Set-up and Router Configuration

After designing the network, the router has to be configured correctly to let all computers have access to your LAN. First read the following paragraphs to learn which commands you can use to configure the router, and then proceed with the tasks.

Log in on the machine by setting up an *xterm* connection to the name of the machine you want to connect to.

| Machine | Name |
|---|---|
| Router | rout |
| Workstation 1 | ws1 |
| Workstation 2 | ws2 |
| Workstation 3 | ws3 |
| WebServer | web |
| FTP-server | ftp |
| Host in 12-host network | visit |

Table 1: *node names*

### 2.1 Configuring interfaces

To configure an (ethernet) interface, use the **`ip`** command. The syntax is as follows:

```
gr<X>@rout:~$ ip address show dev rout-eth<X>
gr<X>@rout:~$ ip a sh dev rout-eth<X> # same, abbreviated
gr<X>@rout:~$ sudo ip address add <ip-address>/<netmask> dev rout-eth<X>
gr<X>@rout:~$ sudo ip a a <ip-address>/<netmask> dev rout-eth<X> # abbr.
```

The first command displays the settings for the specified interface, the second is used to configure the interface (with the specified IP address, netmask). For more information, see the ip man-page:

```
gr<X>@rout:~$ man ip
```

## 2.2 Completing the routing table

The **ip a a** command automatically adapts the routing table. However, is your routing table complete now? Have a look, and adapt where needed. Hint: how does your router reach the Internet? To add or remove a routing entry from the routing table, use the **ip route** command. Syntax:

```
gr<X>@rout:~$ ip route
gr<X>@rout:~$ sudo ip route add <network address>/<netmask> via <gateway address>
gr<X>@rout:~$ sudo ip r a <network address>/<netmask> via <gateway address>  # abbr.
gr<X>@rout:~$ sudo ip route add default via <gateway address>
gr<X>@rout:~$ sudo ip route del <network address>/<netmask> via <gateway address>
gr<X>@rout:~$ sudo ip r d <network address>/<netmask> via <gateway address>  # abbr.
```

The first command displays all routing table entries, the second and third adds an entry for a network/netmask via a specified gateway or next hop, the fourth adds a default route and the fifth and sixth are used to remove entries. Take into account that **ip a a** also adds routing information: routes for directly connected networks are automatically added when the interface is configured. For more information, see the man-page.

# 3 Tasks : configure the network

## 3.1 Tasks : configuring the router

None of the nodes in your network have been configured. We will start with the company's router:

- Configure the router (interfaces & routing table), using the commands that are described above in this document. Complete the routing table, so your router uses the ISP to connect to the Internet.

- Test your uplink by checking the connectivity. First, check the config of the router. Can you also ping a server on the Internet?

## 3.2 Tasks: configuring the hosts

Now the router has been configured, you can use the same set of commands to configure the hosts of your network.

- Configure the interface of the hosts, the server and the visitor node.

- Then, check whether you can reach all the components of the set-up from the router using the ping command.

- Can a host ping the IP-address of the webserver? What do you need to add on the nodes to make this work?

## 3.3 Tasks: re-configuring the ISP router

Although you might have fully configured your router AND the nodes in your network, something is still missing as you are not able to reach the Internet from your hosts. Does your ISP know the network that you have been configuring?

- Check the routing table of the ISP router. Which network(s) are missing?
  Add the necessary information to the ISP router. You can aggregate all your subnets including the not-used networks into a single /24 network.

- Your workstations should now be able to ping 8.8.4.4.

- If you have configured your ISP router correct, the address 10.20.82.253 will match with two lines in your finished routing table. Explain which entry will match, and what general rule takes care of this.