Simple OSPF configuration

Prosta konfiguracja dynamicznego protokołu rutingu OSPF

- W ramach zajęć zostanie utworzona sieć składająca się z czterech maszyn wirtualnych
- Studenci nauczą się, w jaki sposób uruchomić kilka maszyn wirtualnych jednocześnie z wykorzystaniem oprogramowania **`kathara`**
- Maszyny wirtualne zostaną wyposażone w podstawową konfigurację, jeszcze przed ich uruchomieniem.

Simple OSPF configuration

Prosta konfiguracja dynamicznego protokołu rutingu OSPF

- W ramach zajęć należy:
  - Skonfigurować protokół OSPF działający w jednym obszarze (ang. ***area***)
  - Zrozumieć przeznaczenie interwałów ***hello-interval*** oraz ***dead-interval*** protokołu OSPF, obowiązujących w danym segmencie
  - Zrozumieć funkcje rutera wyróżnionego (ang. ***designated router***, **DR**)
  - Zrozumieć działanie parametrów protokołu OSPF: ***router-id***, ***priority***
  - Zrozumieć działanie protokołu OSFP w przypadku sieci rozgłoszeniowych (ang. *broadcast*) oraz typu punkt-punkt (ang. *point to point*)
  - Przechwycić i przeanalizować wymianę komunikatów protokołu OSPF

Simple OSPF configuration
Prosta konfiguracja dynamicznego protokołu rutingu OSPF

- **Głównym celem zajęć jest**
  - zapoznanie się z działaniem dynamicznego protokołu rutingu OSPF opartego na algorytmie stanu łącza (ang. *link state*)
  - nabycie umiejętności konfigurowania i analizowania protokołu ospf w celu zapewnienia osiągalności urządzeń sieciowych oraz diagnozowania przyczyn trudności w prawidłowym doborze trasy.

Preparing a kathara lab

Authors: G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini

Modified and extended for the purpose of the IP Networks lab

Simple OSPF configuration

Authors: K. Kosek-Szott, P. Pacyna, S. Szott

Introduction

preparing a lab

- a kathara lab is a set of preconfigured device containers that can be started and halted together
- we will learn how to set up a standard kathara lab that can be launched by using the **`lcommands`**
  - instructions based on https://github.com/KatharaFramework/Kathara-Labs/blob/master/001-kathara-introduction.pdf
- we will create a **simple OSPF** configuration

kathara labs using `lcommands`

- a standard kathara lab is a directory tree containing
  - a `lab.conf` file describing the network topology
  - a set of subdirectories, each containing the configuration settings for one virtual machine (a router or host)
  - `.startup` file that describes actions performed by virtual machines when they are started
- **Download `tar.gz` with a sample kathara lab**
- **Extract the files to a hard disk**
- **List the files in the lab directory**

`lab.conf`

- this file describes
  - the devices to be started
  - the topology of the network that interconnects the lab devices
- list of **`machine[arg]=value`** assignments
  - **`machine`** is the name of the device (e.g., **`pc1`**)
  - **`arg`** is an integer number (***`i`***) representing the **`eth`*`i`*** interface
  - **`value`** is the name of the collision domain to which the interface **`eth`*`i`*** should be attached

- **Note**: a collision domain represents a multi-access network such as, for example, an Ethernet switch.

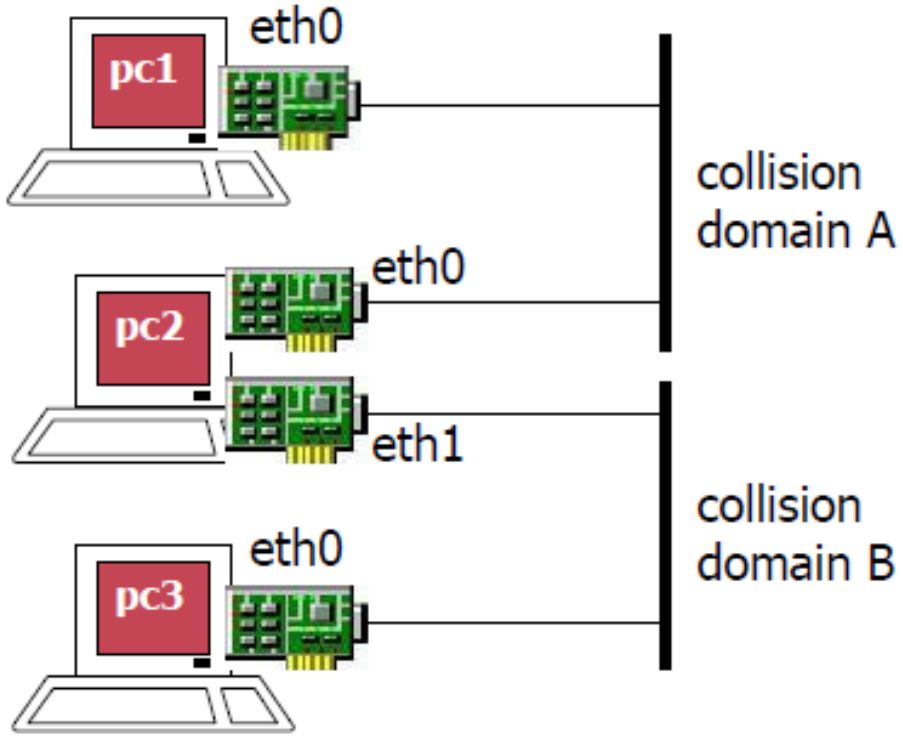
lab.conf

- example of **lab.conf**

```
pc1[0]=A

pc2[0]=A
pc2[1]=B


pc3[0]=B
```



Share files mirrored to the device

- There are two ways to share mirrored files:
  - the **`/shared`** directory inside a device directly points to the **`shared`** directory inside the lab
    - by default it is **ENABLED**, you can disable it in the settings
  - the **`/hosthome`** directory inside a device directly points to the **home directory** of the current user of the host
    - by default it is **DISABLED**, you can enable it in the settings

lab subdirectories

- kathara starts one virtual device for each subdirectory, with the host name set to the subdirectory name.

- contents of subdirectory **`device`** are copied into the root (**`/`**) of **`device`**'s filesystem
  - for example, **`device/foo/file.txt`** is copied to **`/foo/file.txt`** inside the **`device`**
  - this only happens the 1st time the **`device`** is started; in order to force the copying again, you have to remove the device container.

`startup` files

- shell scripts that are executed inside a device right after its startup
- a **`.startup`** file can be used, for example, to configure network interfaces and/or start network services
  - for example:

```
ifconfig eth0 10.0.0.1/24 up
/etc/init.d/quagga start
```

launching/stopping a lab

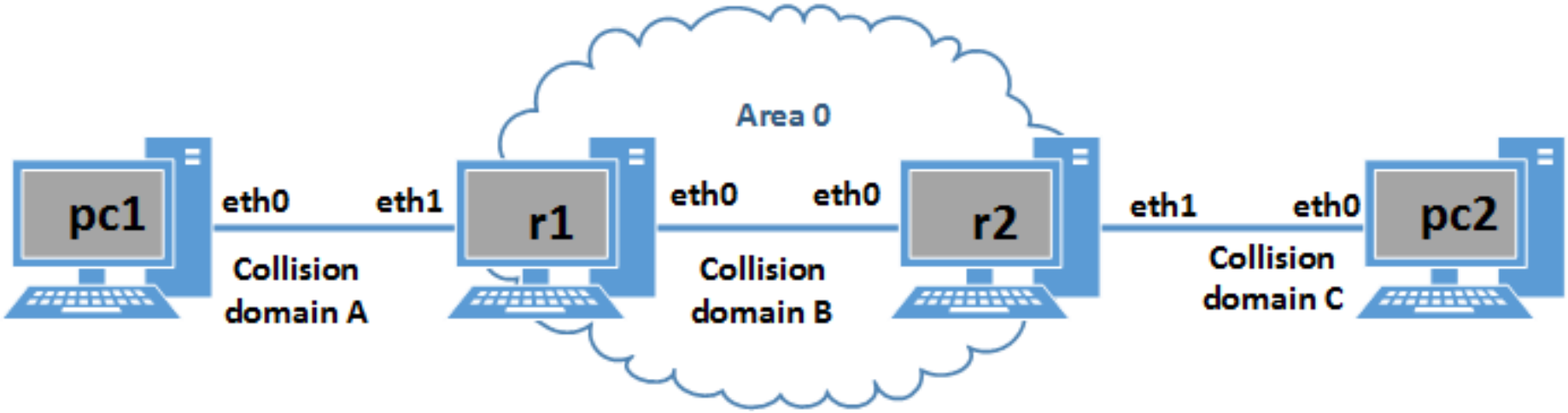
- **Open a terminal**
- Go to the lab directory (**`cd lab_directory`**)
- Launch a Kathara **`l-command`**
  - where **`lcommand`** is one of the following
    - **`Kathara lstart`**, to start the lab
    - **`Kathara lclean`**, to stop the lab
    - **`Kathara lrestart`**, to restart the lab

copyright notice

simple OSPF lab (DR election)

assign IP addresses

- Assign IP addresses to ALL interfaces on routers and PCs



Please note that the network IP address in Area 0 should include the LAB-ID assigned to you by the lab instructor, e.g., in the following way: <LAB-ID>.0.0.0/24

update `lab.conf`

- inside the lab directory update the **`lab.conf`** file and insert topology-defining entries for **`pc1`**, **`pc2`**, **`r1`**, and **`r2`**

- exemplary entry

```
pc1[0]="A"
```

create `*.startup` files

- Create **`startup`** files for all devices
  - in your lab directory you should have: **`r1.startup`**, **`r2.startup`**, **`pc1.startup`**, **`pc2.startup`**
- Configure the previously chosen IP address(es) for each device
- Exemplary entry

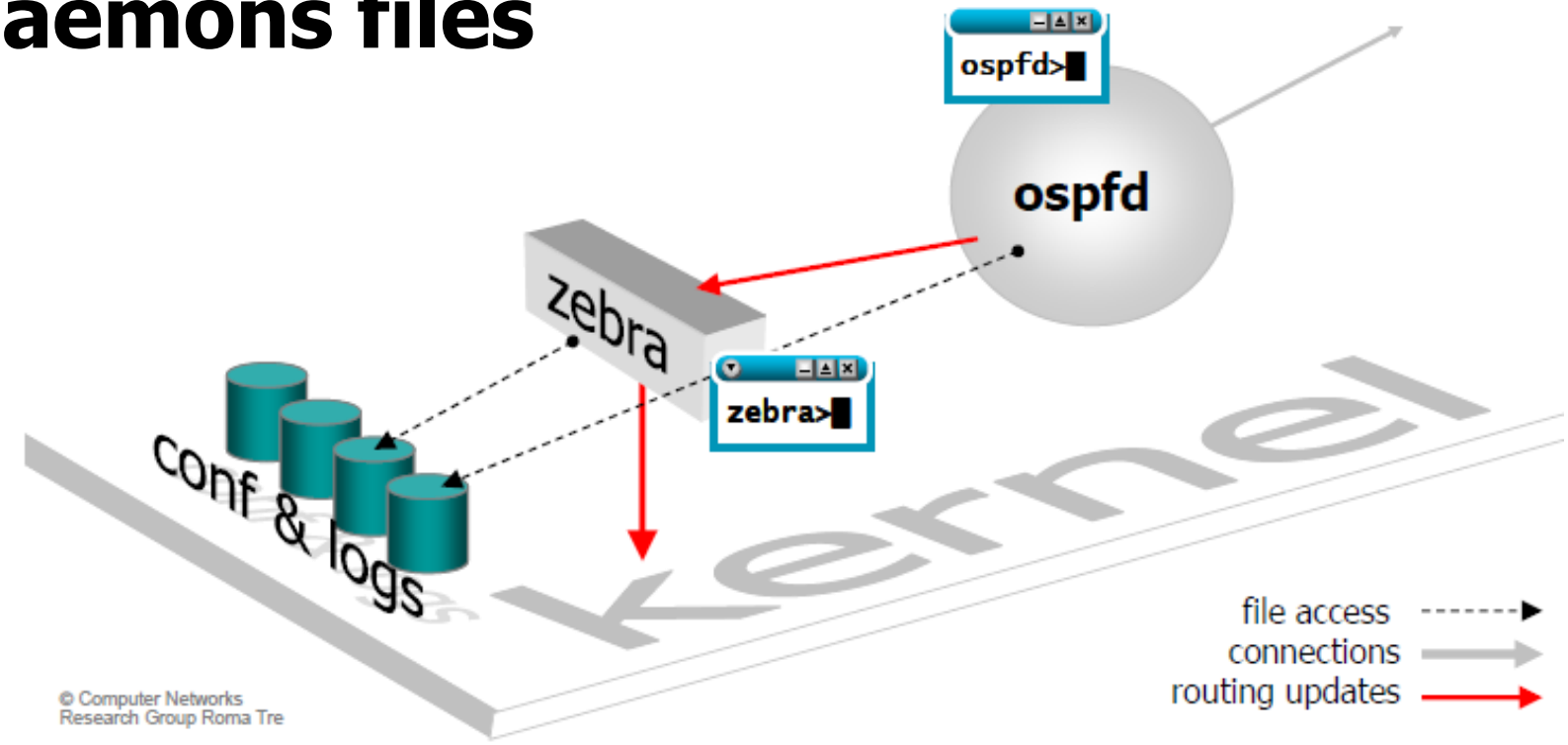
```
ifconfig eth0 10.10.10.1 up
```

create subdirectories for **`r1`**, **`r2`**, **`pc1`**, and **`pc2`**

- duplicate the **`router`** and **`pc`** subdirectories in your lab directory
  - as a result you should have two **`pc`** subdirectories and two **`router`** subdirectories
- set the names of your subdirectories to **`r1`**, **`r2`**, **`pc1`**, and **`pc2`**

modify `daemons` files in `r1` and `r2` subdirectories

- enable **`zebra`** and **`ospfd`** routing daemons for **`r1`** and **`r2`** by editing their **daemons files**



start the lab

- start the configured laboratory using the **`kathara lstart`** command in the lab directory
- if you did not configure **`quagga`** routing daemon to be started automatically, do it now
  - on **`r1`** and **`r2`** issue the following command

    `/etc/init.d/quagga start`
- set appropriate read/write properties to the /etc/quagga directory on both routers:
  - **`chmod a+rw /etc/quagga`**

OSPF configuration
`dummy0` interface

- on each virtual router, **telnet** to the **`zebra`** daemon and configure the special loopback address called **`dummy0`**

- **Hints**
  - **`> configure terminal, interface dummy0`**
  - enter the **`no shutdown`** command
  - enter a special loopback address which is different than the IP adderess configured on the **`eth`** interface
  - write changes to the **`zebra`** configuration file

Question 1: Why should we enable a special loopback address for OSPF?

OSPF configuration
OSPF intervals

- on each running virtual router, **telnet** to **`ospfd`** and change the **`hello-interval`** to 5 and the **`dead-interval`** to 20 on the **`eth0`** interface [quagga manual]

Note: these parameters are specific to an interface. They depend on the network type.

Question 2: What are these intervals used for? Why are we changing them to lower values? Should they have the same values on both sides of the link?

OSPF configuration

- on each virtual **router**, while connected to **`ospfd`**
  - create an OSPF routing process (**`router ospf`**)
    - configure the **`ospf router-id`**
      - set it to the IP address value assigned to **`eth0`**
    - define the IP addresses on which OSPF runs
      - **`network`** command

      HINT: don't forget to define the **`area ID`**
  - write changes to the **`ospfd`** configuration file using the **`write file`** command

OSPF routing information

- on each virtual router, check the running system information using different **`show ip ospf <…>`** commands

Question 3: Which router was chosen as the Designated Router? Why? What is the status of the other router ?

Reporting

- Please deliver the following items to the UPEL system using your account
  1. A photocopy or a screenshot showing the output of the following commands:
     - **`show ip ospf`** executed on router r2
     - **`show ip ospf neighbor`** executed on router r2

remote destination availability

- **`ping pc1`** from **`pc2`** and vice versa
- if the devices cannot reach each other – fix the problem

Question 4: How did you fix the problem?

Reporting

- Please deliver the following items to the UPEL system using your account
  1. A photocopy or a screenshot showing the output of the following commands
     - **`ping`** of pc1 executed on pc2.

OSPF: choosing the DR

- on the router that was chosen as the DR change its **`router-id`** to the one assigned to the **`dummy0`** interface

Question 5: Which router was chosen as the Designated Router? Can you explain why? What is the status of the other router?

OSPF: choosing the DR cont'd

- on the router which is currently the DR change the **`priority`** to zero
  - **Hint**: go to the interface configuration

Question 6: What is the state of that router now? Can you explain why?

OSPF: choosing the DR cont'd

- change the priority values on both routers so that they are different than zero. Choose them so that the router with the lower **`router-id`** has the higher **`priority`** value
- **Hint:** to see the result on **`r1`** and **`r2`** go to the **`ospf`** configuration and issue the commands
  - **`no network`** `<IP address>` **`area 0`**
  - **`network`** `<IP address>` **`area 0`**

Question 7: Why do we need to disable and then enable ospf? Which is more important factor when choosing the DR: **`router-id`** or **`priority`** ?

Playing with network type

- Change the network type of one link to `point-to-point`
  Question 8: Is a DR selected? Can you explain why?

- Change the network type back to `broadcast`.

- Add the third virtual router to the environment, also connected to the same broadcast domain. Enable OSPF. Observe how the DR election algorithm works for three routers.
  Question 9: Check the **`neighbour count`** and the **`adjacent neighbour count`** on each router. Explain these numbers.

Reporting

- Please deliver the following items to the UPEL system using your account
  1. Your lab directory with all introduced OSPF configurations for all the devices.

DR selection

- Create a new scenario with *three* routers connected in series.

Question 10: Can a router be a DR on one interface and, at the same time, not a DR on its another interface ?

sniffing the OSPF protocol **on BMA networks**

- use **`tcpdump`** to sniff on some router interface
- capture the conversation between routers
- move the file from **guest device** to **host machine**
- browse the file using **`wireshark`** on the host machine in the graphical mode
- useful commands:

```
tcpdump -i eth0 proto ospf -w logfile.cap
tcpdump -r logfile.cap proto ospf
tcpdump -v -r logfile.cap proto ospf
```

Sniffing the OSPF protocol **on BMA networks** cont'd

Question 11: What IP addresses are used during the message exchange?

Question 12. What OSPF message types are used during the conversation?

Question 13. What information is conveyed inside **OSPF LS Update** packets?

Question 14. How often are packets exchanged? What events can trigger an **Update**?