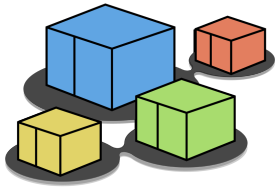
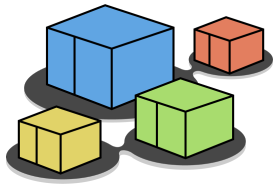


Kathará

# kathara lab

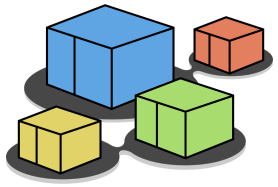
## FRRouting

| | |
|---|---|
| **Version** | 1.1 |
| **Author(s)** | L. Ariemma, T. Caiazzi, G. Di Battista, M. Patrignani, M. Scazzariello |
| **E-mail** | contact@kathara.org |
| **Web** | https://www.kathara.org/ |
| **Description** | experiences with FRR configurations and command line interface |



# copyright notice

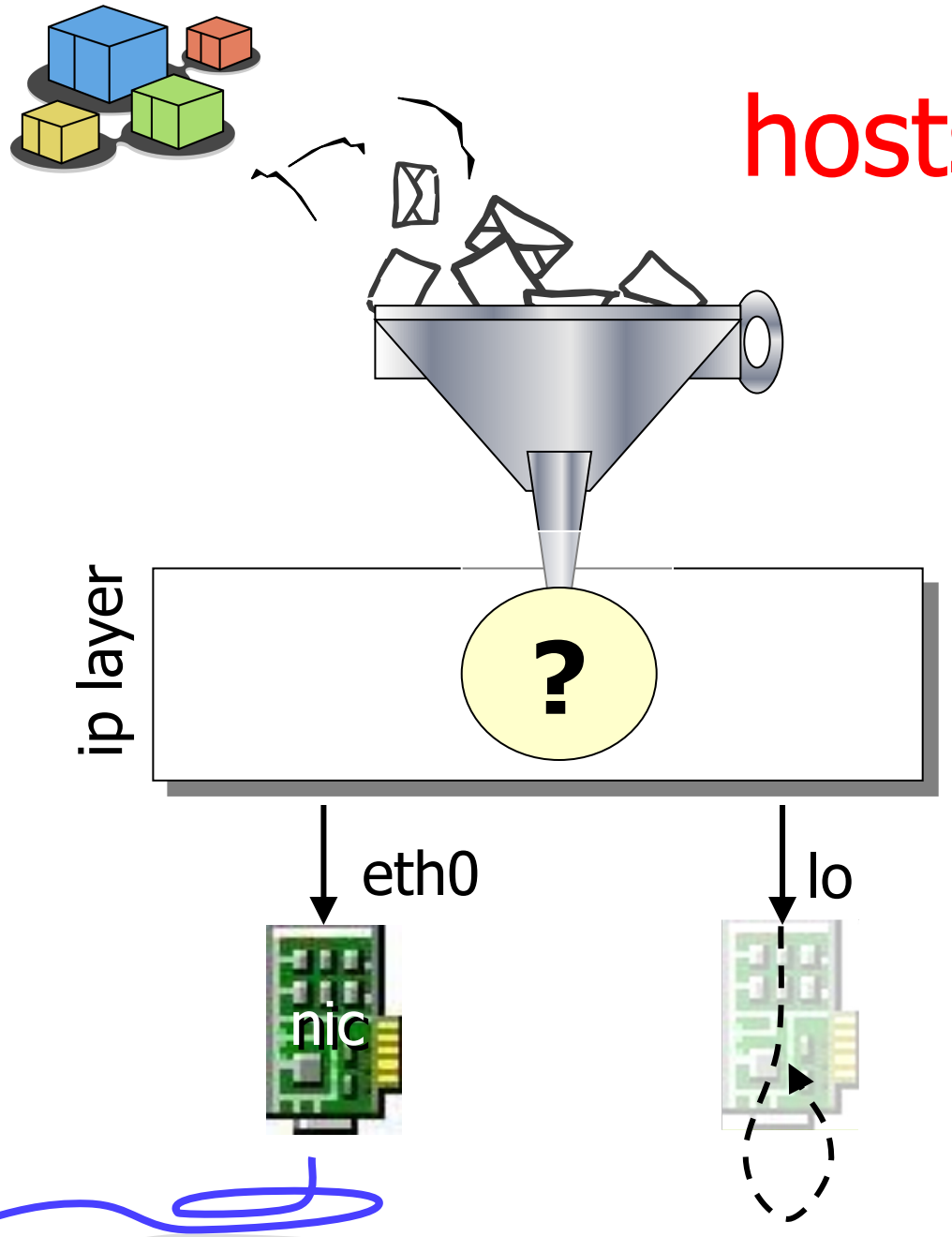
- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material “as is”, with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.



# preconditions

- for this lab we assume you have chosen “kathara/frr” as the default image of your Kathará installation
  - execute “kathara settings”
    - select “choose default image”
    - select “kathara/frr”
    - exit from the settings procedure

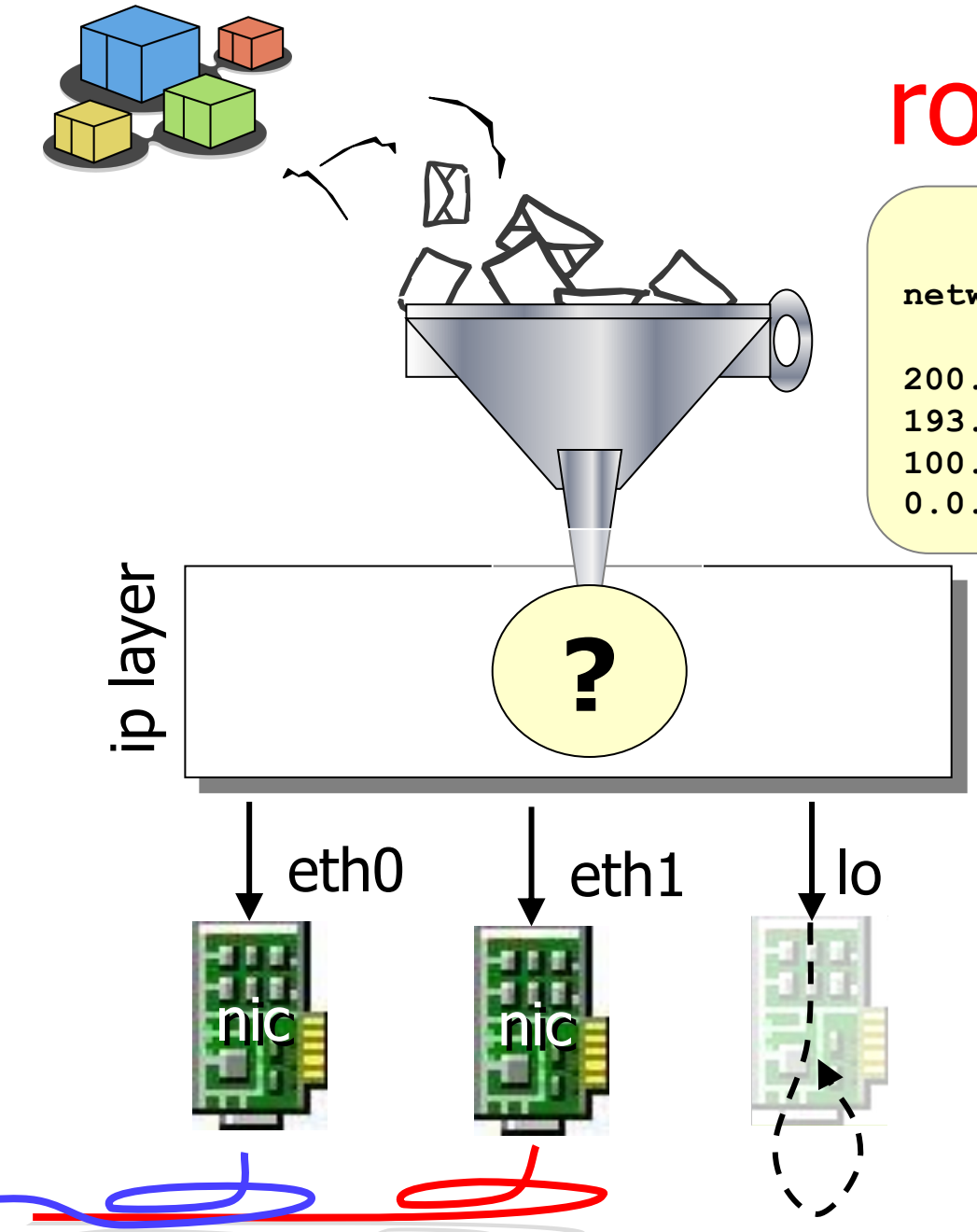
# hosts need routing




- each host with a network stack performs some elementary routing
- at the very least, the network stack may be used to access local services (e.g., Xorg)
- the host must decide when a packet needs to be sent to the network interface card (nic) and when it needs to be bounced to the loopback interface (lo)


© Computer Networks Research Group Roma Tre

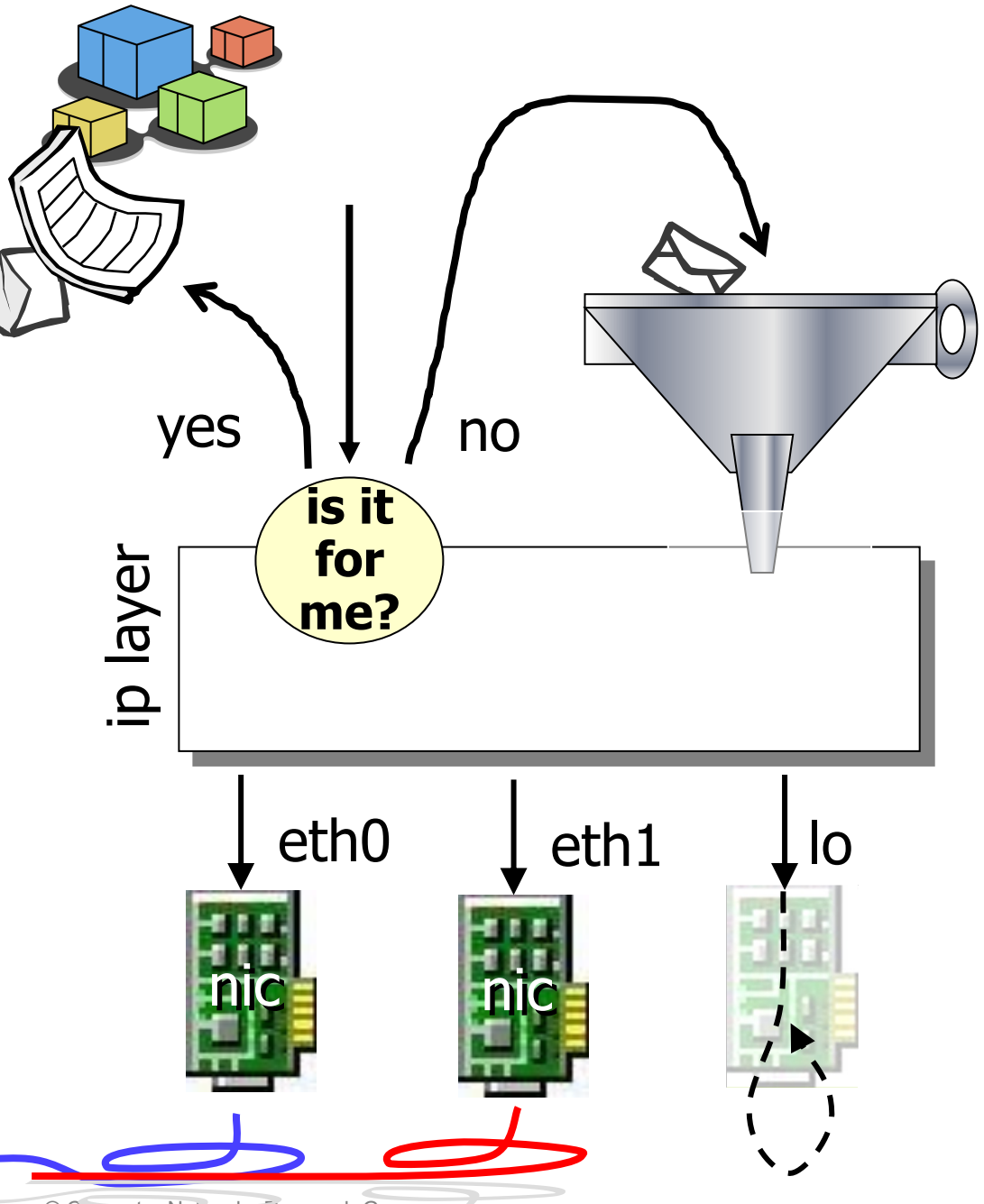
# routing table




**routing table**

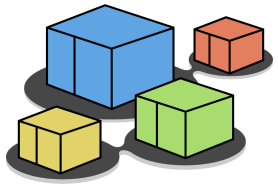
| network | netmask | nexthop | dev |
|---|---|---|---|
| 200.3.24.0 | 255.255.255.0 | 12.0.0.4 | eth1 |
| 193.2.0.0 | 255.255.248.0 | 11.0.0.2 | eth0 |
| 100.4.5.0 | 255.240.0.0 | 11.0.0.3 | eth0 |
| 0.0.0.0 | 0.0.0.0 | 11.0.0.2 | eth0 |

the ip layer uses a routing table to decide which is the interface the packet needs to be forwarded to




# routers

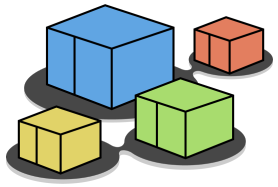
- a router (also called gateway or intermediate-system)
  - has more than one network interface card
  - feeds incoming ip packets (that are not for the router itself) back in the routing process
    - this operation is called relaying or forwarding



# routing protocols

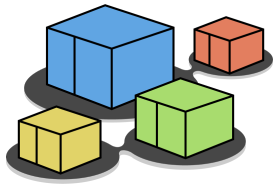
- routing protocols are used to automatically update routing tables, relieving administrators from the need to do it manually
- routers (i.e., devices that run routing protocols) in Kathará are virtual machines that run a specific piece of software that implements routing protocols

**FRRouting**



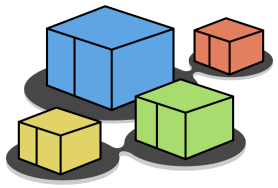
# The FRRouting Suite

a free and open-source Internet routing protocol suite for Linux and Unix platforms

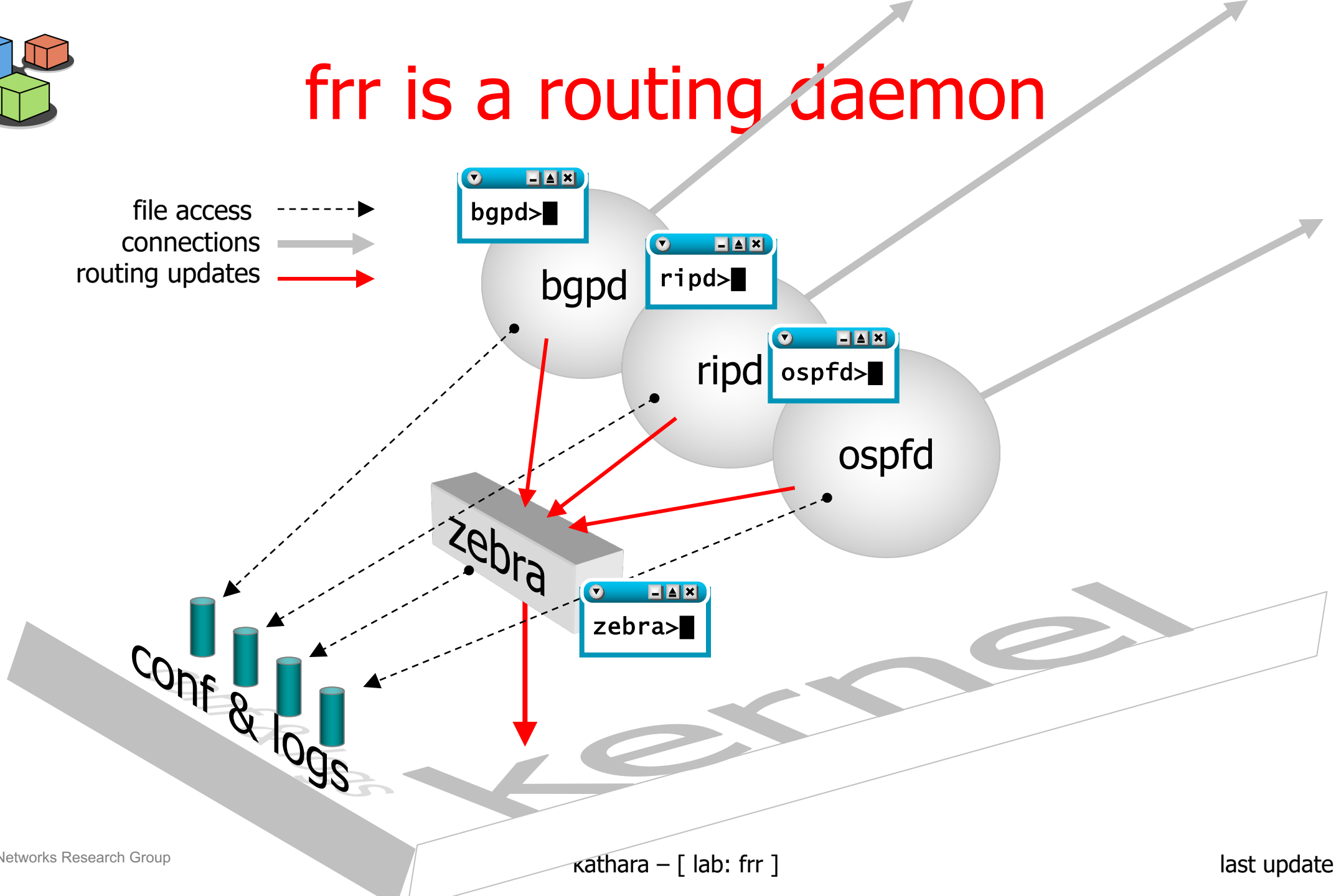


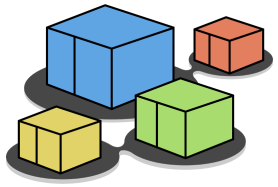
# about the FRRouting Suite

- FRR is a software that implements some routing protocols
  - rip (v1 and v2), ospf (v2 and v3), is-is
  - bgp
  - pim, ldp, bfd, babel, pbr, openfabric, vrrp, ...
- frr has its roots in the quagga project
  - quagga is no longer supported
- in turn, quagga was a community project that originated from (and superseded) the zebra project



# frr is a routing daemon

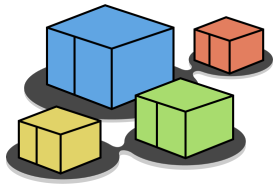




# sample `daemons` configuration file

```
root@r1:~$ cat /etc/frr/daemons
# This file tells the frr package which daemons to start.
#
# Sample configurations for these daemons can be found in
# /usr/share/doc/frr/examples/.
#
# ...
#
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=no
...
```

the rip daemon will be started

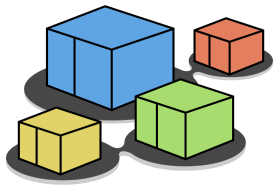


# sample `vtysh.conf` configuration file

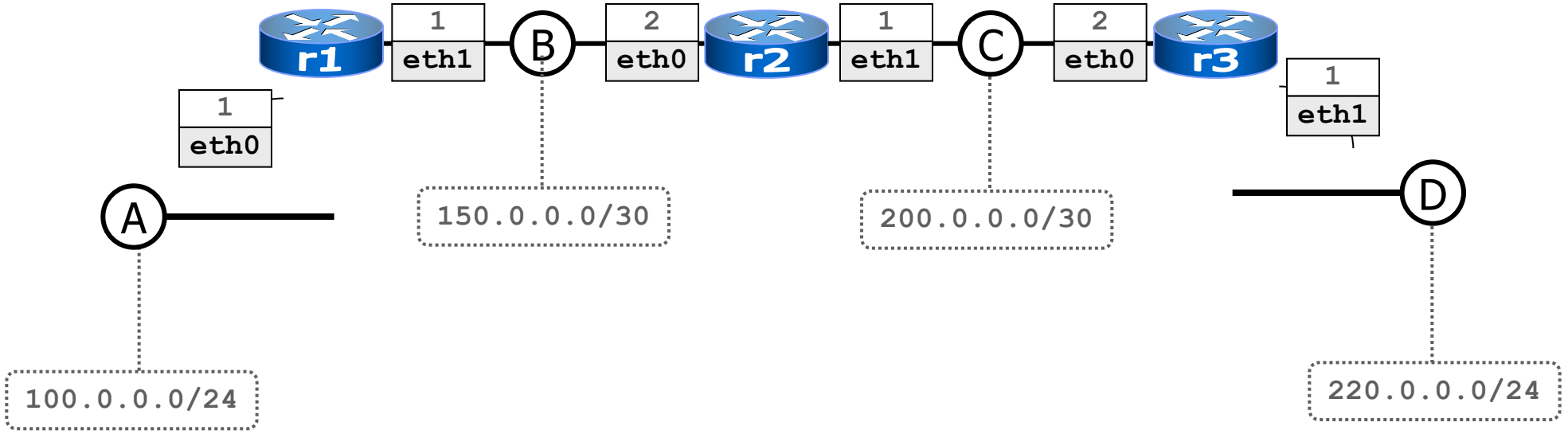
```
root@r1:~$ cat /etc/frr/vtysh.conf
no service integrated-vtysh-config

root@r1:~$
root@r1:~$
root@r1:~$
root@r1:~$
root@r1:~$
root@r1:~$ cat /etc/frr/vtysh.conf
service integrated-vtysh-config
```

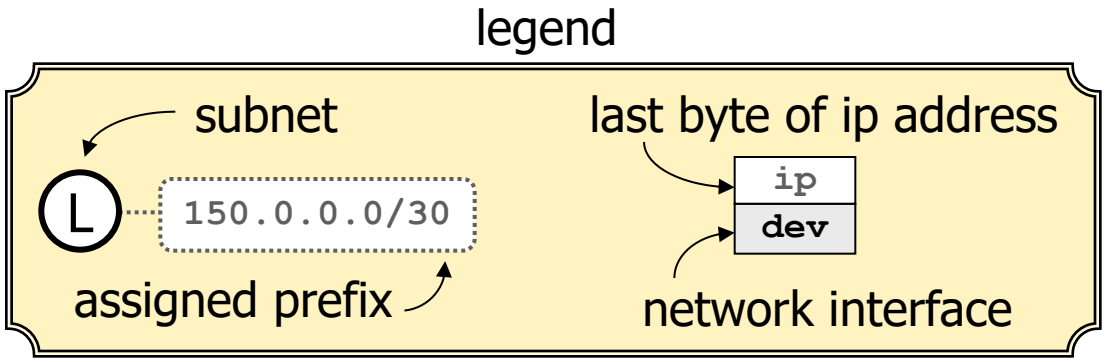
with this command each daemon will need its own configuration file (default)

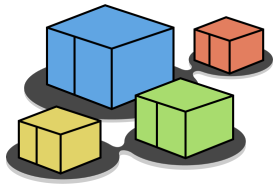
a single configuration `frr.conf` file for all daemons (recommended)



# a simple topology



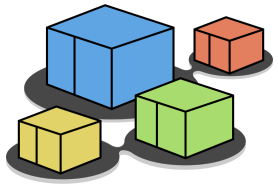





# launching the lab script

- the lab configuration is such that
  - three virtual hosts are created and connected to the right collision domains (virtual hubs)
  - for each virtual host
    - network interfaces are automatically configured
    - frr configuration files are updated
  - the frr routing daemon is automatically started

```
user@localhost:~$ cd kathara-lab_frr
user@localhost:~/kathara-lab_frr$ kathara lstart
```

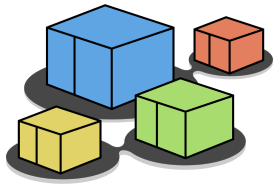


# automatically start frr

- to ease the startup of the lab is better to automatically start the frr
  - the main daemon and all the routing protocols daemons are automatically started with frr

**r1.startup**

```
systemctl start frr
```

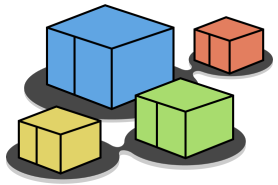


# a one-fits-all shell

- instead of having to connect to each single daemon, users can interact with frr by using a built-in shell, called **`vtysh`**

```
root@r1:~$ vtysh
Hello, this is FRRouting (version 9.0.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r1#
```

- the user is not prompted for a password
- all the commands to manage all the routing daemons (including zebra itself) are available in this shell

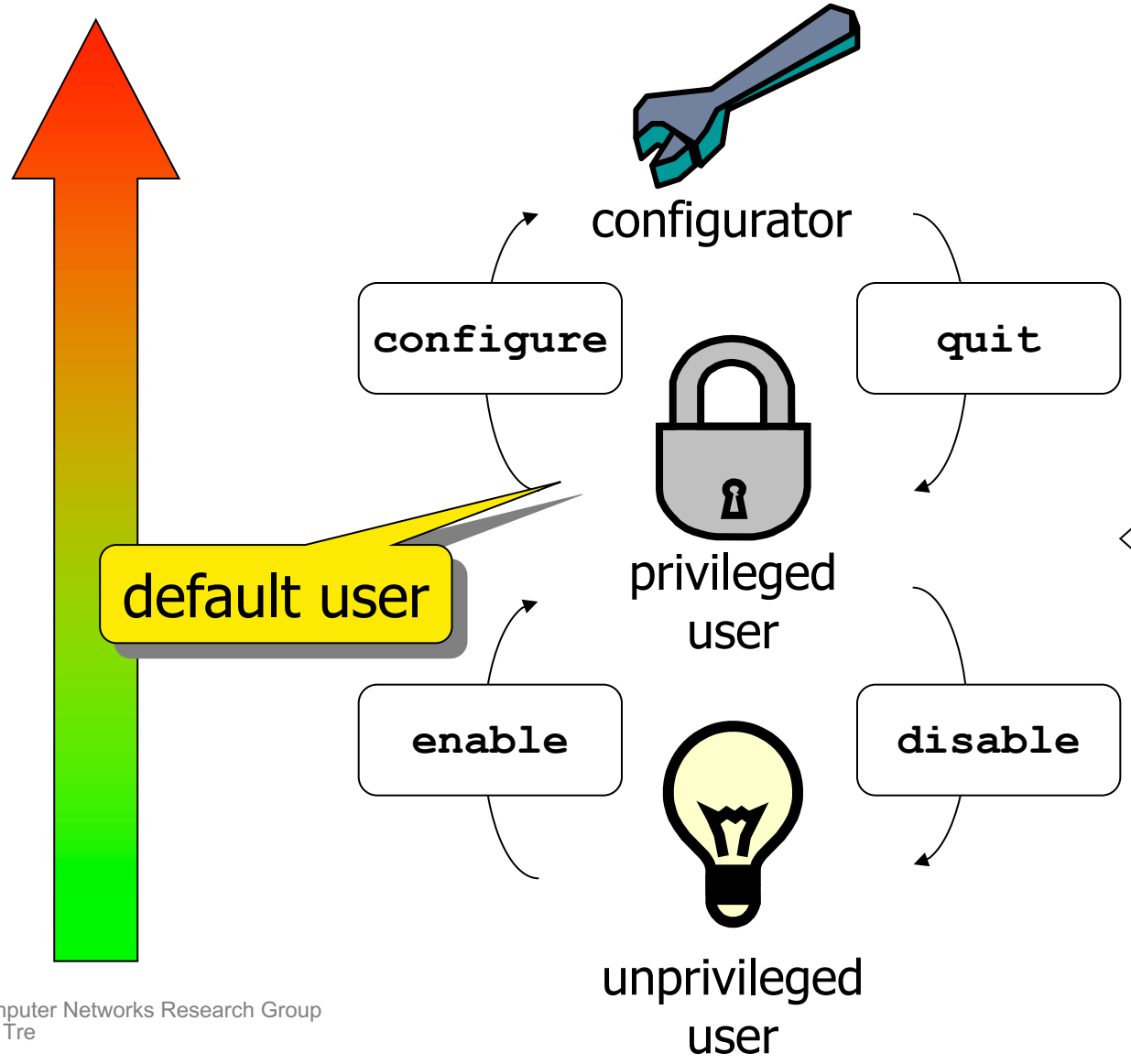


# available `vtysh` commands

- press '?' at the command prompt to retreive all commands available
  - here are displayed only the main ones

```
r1# ?
  configure   Configuration from vty interface
  enable      Turn on privileged mode command
  exit        Exit current mode and down to previous mode
  no          Negate a command or set its defaults
  ping        Send echo messages
  quit        Exit current mode and down to previous mode
  show        Show running system information
  terminal    Set terminal line parameters
  write       Write running configuration to memory, network, or terminal
```

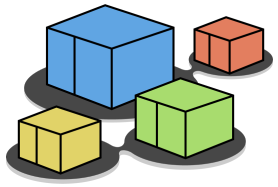
# privileges on a router




- performs persistent configurations affecting the router behavior

- sees critical information
- performs elementary configurations that do not affect router behavior (e.g., set time and date)
- commits changes

- sees noncritical information



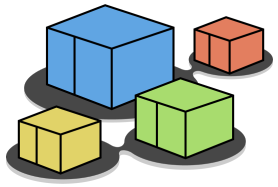
# inspecting the frr routing table

```
r1# show ip route
  Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
         I - IS-IS, B - BGP, E - EIGRP, N - NHRP, T - Table, v - VNC,
         V - VNC-Direct, A - Babel, F - PBR, f - OpenFabric,
         > - selected route, * - FIB route, q - queued, r - rejected, b - backup
         t - trapped, o - offload failure

R>* 100.0.0.0/24 [120/3] via 200.0.0.1, eth0, weight 1, 00:01:32
R>* 150.0.0.0/30 [120/2] via 200.0.0.1, eth0, weight 1, 00:01:32
C>* 200.0.0.0/30 is directly connected, eth0, 00:01:33
C>* 220.0.0.0/24 is directly connected, eth1, 00:01:33
```

- FIB entries from this table (marked with a '>') are injected into the kernel routing table

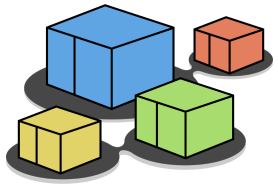


© Computer Networks Research Group Roma Tre



# inspecting the current configuration

```
r1# show running-config
Building configuration...

Current configuration:
!
frr version 9.0.1
frr defaults traditional
hostname r3
no ipv6 forwarding
no service integrated-vtysh-config
!
end
r1#
```



# some observations

- the frr configuration language is “operational” and not “declarative”
  - the configuration file is simply a list of configuration commands that would bring the router to its operative state
  - it is not an abstract description of how the router should behave according to the configurator
- you cannot remove a configuration command
  - if the command has a certain string (e.g., “ipv6 forwarding”) in order to remove it you should issue the same string preceded by “no” (e.g., “no ipv6 forwarding”)