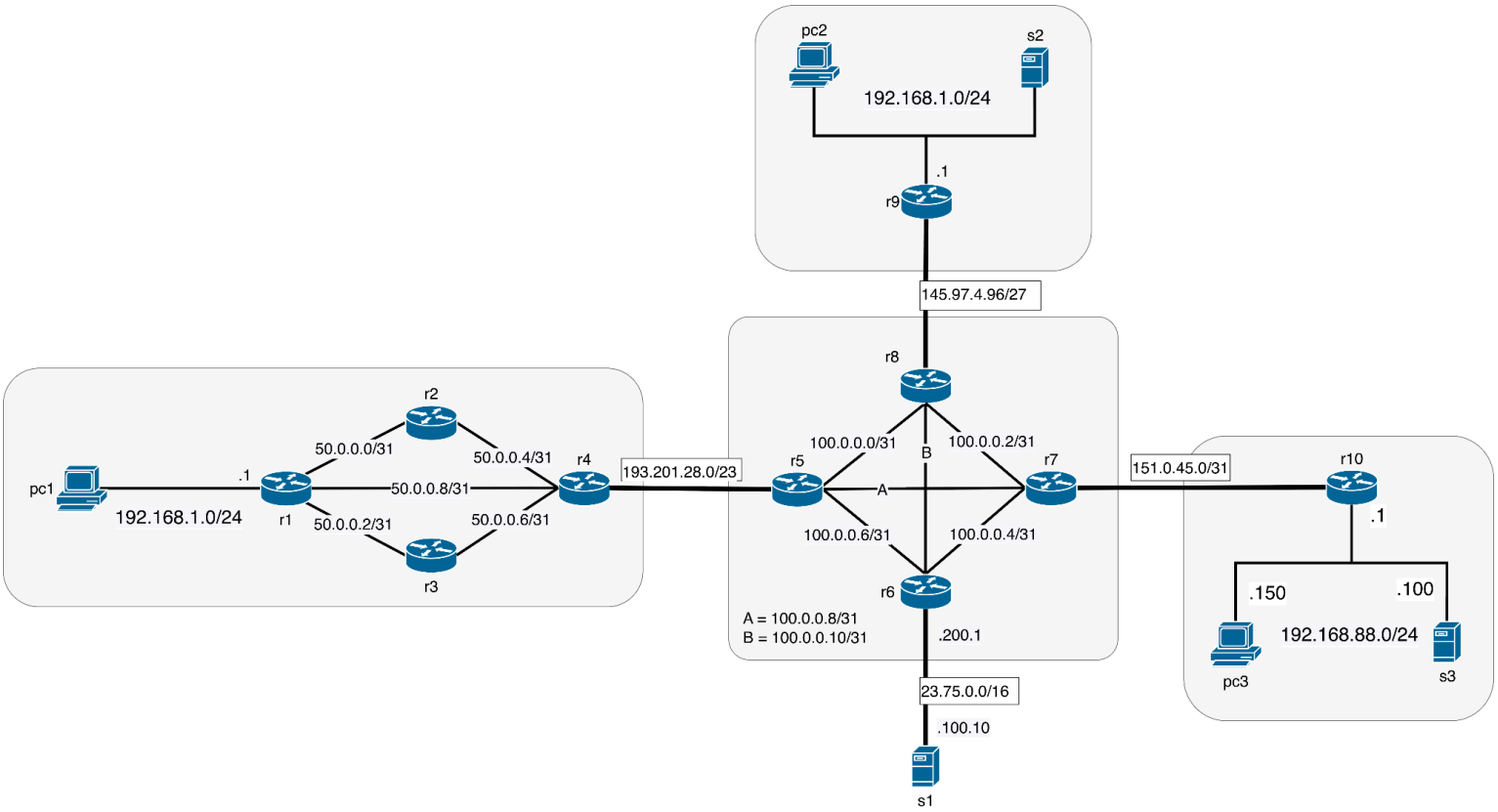
Network Infrastructures A.A. 2025 / 2026
Homework



Given the topology in figure, reproduce it in Kathara. You must use container names and addresses specified in the figure above. Container names should be all in lowercase.
The network is composed of 10 routers (r1, …, r10), 3 servers (s1, s2, s3) and 3 pc (pc1, pc2, pc3). For subnets connecting two routers, the addresses are assigned with the following rule: the router with the highest number takes the highest address and the other takes the lowest. The maximum points are **16** and are assigned as follows:

1. +0.5 points. Lab created with correct lab.conf and folders created correctly. Nodes pc2, s2 and the topmost interface of r9 are in the same collision domain. Nodes pc3, s3 and the bottom-most interface of r10 are in the same collision domain.

2. +1 point. Configure a DHCP server on r1 and r9 assigning addresses and default gateways to the clients in the private subnets 192.168.1.0/24. Assign to the remaining nodes static IP addresses via /etc/network/interfaces. Configure default gateway routing on pc3 and s3.
3. +1 point. On routers r{5,6,7,8}, configure OSPF using the same 0.0.0.0 area and a static route to reach 50.0.0.0/24 through r5. On the remaining routers, configure static routing so that any router is able to reach (e.g. to ping) all the interfaces with a non-private address belonging to every other router.
4. +0.5 points: Set up a NAT on r1, r9 and r10 for traffic exiting their private subnets.
5. +0.5 points: Set up a firewall on r1, r9 and r10 blocking all traffic directed to their private subnets unless it is initiated by the private subnets themselves.
6. +1 points: Set up a SSH server on s1 with a user "ubuntu", accessible via pubkey authentication from pc1. It is not mandatory to create the SSH key at startup.
7. +2 points: create a new Certification Authority (CA) having as Common Name (CN) and encryption password your University numeric id (a.k.a. your matricola). Generate a certificate for a server with CN "vpn_server" and for two clients with CN "vpn_client_1" and "vpn_client_2". It is not mandatory to create certificates and keys at startup.
8. +3 points: Set Up an OpenVPN server on s3 and an OpenVPN client on pc1 and pc2.
    - Use certificates and keys you generated in the previous point.
    - The subnet of the VPN is 10.10.0.0/24.
    - Configure the VPN to use TCP and to use port 7070.
    - In the server configuration file add the directive "`client-to-client`", which enables two clients to "see" each other on the VPN.
    - The VPN ip address of s3 should be the default one, the one of pc1 and pc2 should be respectively 10.10.0.100 and 10.10.0.200.
    - Configure the server so that it pushes the route to subnet 192.168.88.0/24 to the clients.
    - The VPN should not be run at startup.
9. +1 points: Configure r10's firewall enabling devices to connect to the VPN of the previous point. If done correctly, on pc1 and pc2 you should be able to connect with OpenVPN to s3 by specifying r10's public address.
10. +1 points: Generate some traffic on the VPN by opening a netcat listener on TCP:8080 on pc1 and connect to it from pc2. Capture these packets on one of the interfaces of r1 and save the capture in "`/shared/capture_1.pcap`".
11. +1 points: Generate some other traffic on the VPN by performing a ping from pc1 to pc3. Capture packets on the interface of s3 and save the capture in "`/shared/capture_2.pcap`".

12. +0.5 points: Set up a firewall on r4 blocking all forwarded traffic which is *not* SSH (TCP:22). Now the OpenVPN connection from pc1 should no longer work.
13. +2 point: On pc1, write an SSH port forwarding using the host s1 such that pc1 is able to connect to the VPN. Modify the OpenVPN client configuration of pc1 such that it leverages the port forwarding. It is not mandatory to start the port forwarding at startup. In the lab folder, create a text file "`commands.txt`" and write down the SSH port forwarding command.
14. +1 points: With the port forwarding in place, generate some traffic on the VPN following the same steps described in point **10.** and save the capture in "`/shared/capture_3.pcap`".

The capture and the netcat traffic generation for point **10, 11, 14** should not be run at startup. A tcpdump capture can be saved in a .pcap file with the `-w capture_file.pcap` option.