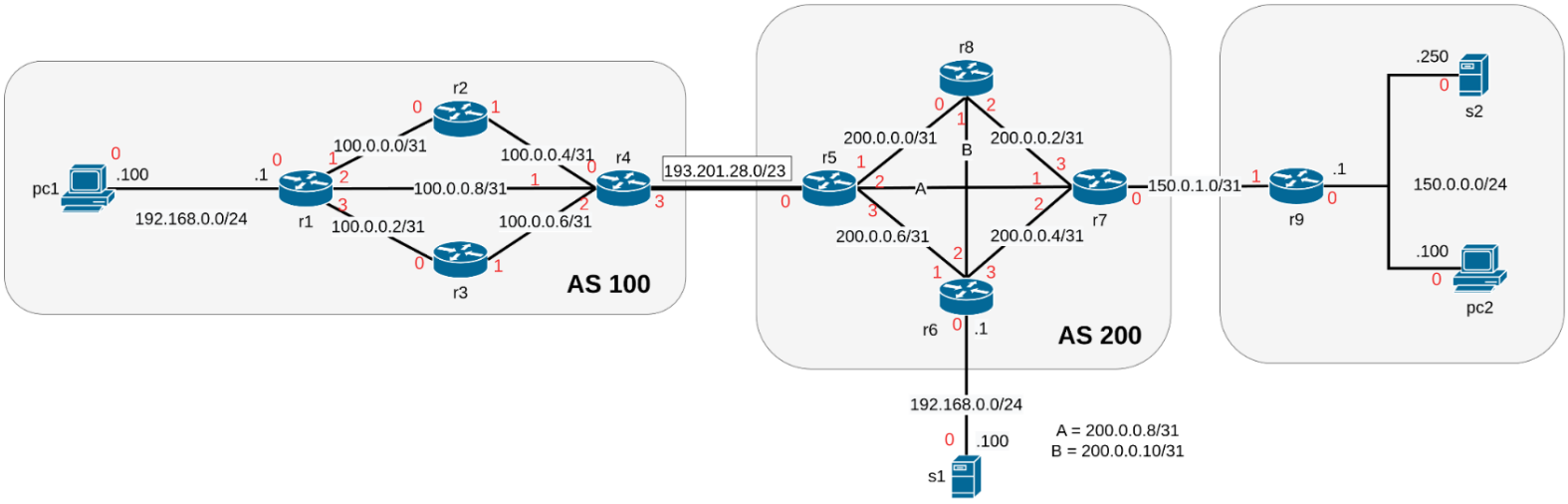
Network Infrastructures A.A. 2024 / 2025
Homework



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.
For /31 subnets, the addresses are assigned with the following rule: the lower router number takes the even address. The maximum points are **18** and are assigned as follows:

1. +0.01 points: Lab created with correct lab.conf and folders created correctly. Nodes pc2, s2 and the rightmost interface of r9 are in the same collision domain. Assign to all routers, PC and servers static IP addresses via /etc/network/interfaces.
2. +0.49 points: Configure OSPF on routers r{6,7,8,9} using the same 0.0.0.0 area. Such routers have a static route to reach 100.0.0.0/24 through r5.
3. +1 point: Configure static routing on routers r{1,2,3}, so that they can reach all interfaces of r{1,2,3,4}.
4. +3.5 points: Configure a BGP peering between r4-r5. AS100 should announce the prefix 100.0.0.0/24 while AS200 should announce 200.0.0.0/24 and 150.0.0.0/16. The peering happens on the peering LAN reported in figure, with r4 and r5 taking respectively the lowest and highest assignable address. Add a default gateway on r5 to make it reach r9, s2 and pc2.
5. +0.5 points: Set up a NAT on r1 and r6 for traffic exiting their private subnets.
6. +0.5 points: Set up a firewall on r1 and r6 blocking all traffic directed to their private subnets unless it is instantiated by the private subnets themselves.

7. +1 points: Set up a firewall on r1 blocking all traffic forwarded to/from its private subnet unless it is SSH (TCP:22) or OpenVPN on TCP (TCP:1194).
8. +2 points: Set up a SSH server on s2 with a user "myuser2", accessible via pubkey authentication from pc1. It is not mandatory to create the SSH key at startup.
9. +2 points: create a **new** CA having as CN your numeric id (a.k.a. your matricola) and generate a certificate for a server with CN "vpn_server" and for two clients with CN "client_1" and "client_2". It is not mandatory to create certificates and keys at startup.
10. +3 points: Set Up an OpenVPN server on s1 and an OpenVPN client on pc1 and pc2.
    - Use certificates and keys you generated in the previous point.
    - The subnet of the VPN is 10.0.0.0/24.
    - Configure the server to use TCP, with the configuration `proto tcp`
    - Configure the server to listen on port 7000.
    - 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 s1 should be the default one, the one of pc1 and pc2 should be respectively 10.0.0.100 and 10.0.0.200.
    - The VPN should not be run at startup.
11. +1 points: Add exceptions to r6'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 via OpenVPN to s1 by specifying r6's public address. 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 r1 router. Save the capture in "`shared/capture_1.pcap`". The capture and the netcat traffic generation should not be run at startup.
12. +0.5 points: Set up a firewall on r4 blocking all forwarded traffic which is *not* SSH. Now the OpenVPN connection from pc1 should no longer work.
13. +2 point: On pc1, write an SSH port forwarding using the host s2 such that pc1 is able to connect to the VPN. Modify the OpenVPN client configuration of pc1 such that it uses the port forwarding. It is not mandatory to start the port forwarding at startup.
14. +0.5 points: Generate some traffic on the VPN following the same steps described in point **11.** and save the capture in "`shared/capture_2.pcap`".

In the lab folder, create a text file "`commands.txt`" and write down the SSH port forwarding command of point **13.**