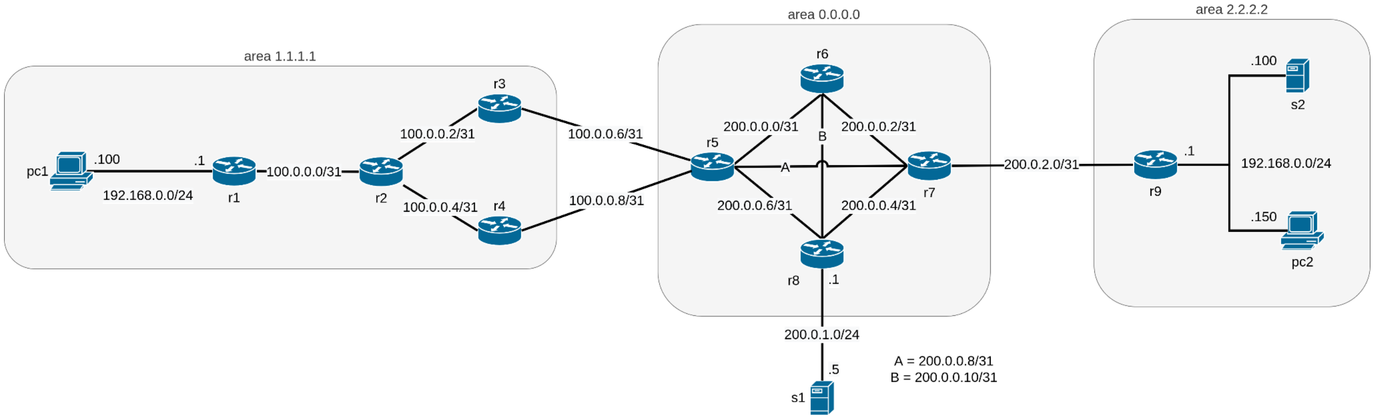
Network Infrastructures – Second Midterm



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 **10** 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. Respect areas given in figure.
3. +0.5 points: Set up a source NAT on r1 and r9 for traffic exiting the subnets 192.168.0.0/24
4. +1 points: Set up a firewall on r1 and r9 blocking all traffic unless is instantiated by the respective NATted subnets
5. +1 points: Set up a SSH server on s1 with a user "myuser1" and on s2 with a user "myuser2", both accessible via pubkey authentication from pc1. Use the same key.
6. +1.5 points: create a **new** CA and generate a certificate for a server with CN "myserver" and for two clients with CN "pc1" and "pc2"
7. +2 points: Set Up an OpenVPN server on s2 and an OpenVPN client on pc1 and pc2.
    - Use the certificates you generated in the previous point.
    - The subnet of the VPN is 10.0.0.0/24.
    - Use TCP instead of UDP.
    - Configure the server to listen on port 5000.

  - 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 s2 should be the default one, the one of pc1 and pc2 should be respectively 10.0.0.100 and 10.0.0.200.

8. +1.5 points: Add firewall rules enabling you to redirect incoming packets TCP on port 1194 directed r9 to s2 port 5000. If done correctly, on pc1 you should be able to connect via OpenVPN to s2 by specifying r9's public address. Generate some traffic on the VPN and capture it on r1 router. Save the capture in "shared/capture_1.pcap".
9. +0.5 points: Set up a firewall on r2 blocking all traffic which is *not* SSH. Now the OpenVPN connection from pc1 from the previous point should no longer work.
10. +1 point: On pc1, configure the OpenVPN client and an SSH port forwarding using the server s1 such that pc1 is able to connect to the VPN on s2.
11. +0.5 points: Now you should be able to connect from pc1 to s2 via SSH. While connected capture some SSH traffic on the interface of s2 and save it 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 10.

It is not required to perform SSH key generation, SSH port forwarding, VPN connection and CA certificate generation at startup.