Redes de Comunicações 1
Project Recurso

Professors:

Susana Sargento	susana@ua.pt
Amaro de Sousa	asou@ua.pt
António Nogueira	nogueira@ua.pt
Paulo Salvador	salvador@ua.pt
Pedro Rito	pedrorito@ua.pt

Num. Mec.: =0/1$x_1x_2x_3x_4x_5$

Description:

Consider the communication network of a very small company depicted in the following figure:

(a) it contains the IPv4 public class C addresses 193.1x_2x_1.1x_3x_4.128/25;

(b) it contains the IPv6 global address 2002:x_3x_5::/60;

(c) it internally uses the range of IPv4 private class B address 172.16.0.0/16 to build several class C networks;

(d) every local network has a private IPv4 and an IPv6 global network;

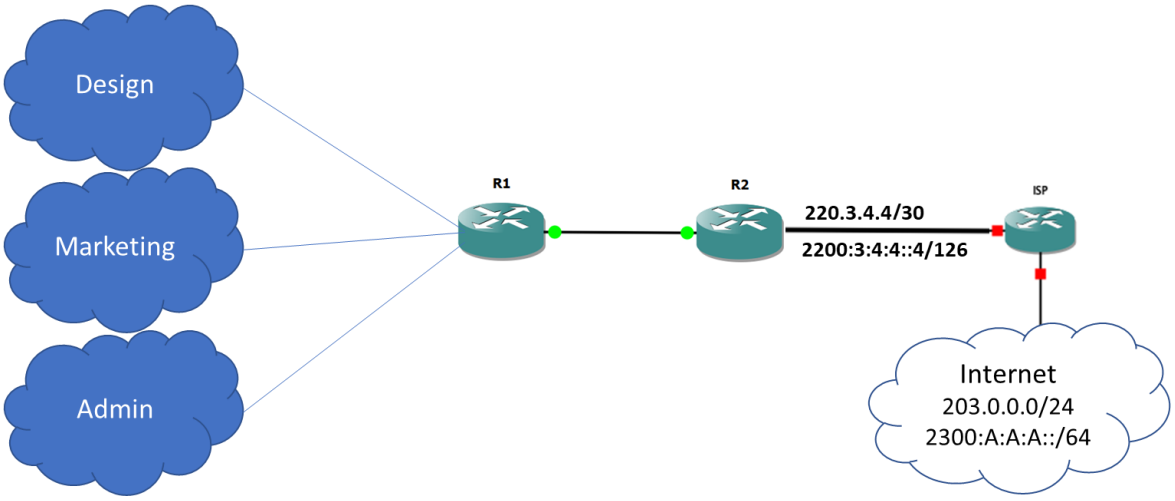
(e) the connection between R1 and R2 is a private network using the already available private addressing;

(f) considering the public IPv4 addressing, there are several equipments in the network that need public addressing: 53 servers at the Design sub-network, 25 servers at the Marketing sub-network, 10 servers in the Admin sub-network, and R2 needs 9 IPv4 public addresses to configure NAT/PAT mechanisms;

(g) the Internet is simulated with the IPv4 network 203.0.0.0/24 and the IPv6 network 2300:A:A:A::/64.

1. Configure the IPv4 and IPv6 addressing in the different equipments.
2. In R2, configure the NAT/PAT mechanisms in an appropriate way. Use the range of public IPv4 addresses to configure the translation with the private network.
3. DHCP server must be configured in R1 to assign private addresses to the network equipments.
4. Configure the IPv4 and IPv6 static/default routing.
5. Place and configure a terminal in the “Internet” to test IPv4 and IPv6 connectivity (place and configure enough terminals in each sub-network to demonstrate full IPv4 and IPv6 connectivity).
6. Configure a DNS server to enable the access through names to the previous server, using sub-domains for the different sub-networks.
7. Considering the sub-domains, configure and test a HTTP server accessed from all terminals, in IPv4 and IPv6.
8. Develop a client-server application (in python using sockets) that allows a client to contact the server to get information about the server HTTP (name, IP address) and company information, such as name and services available.

Deadline: 28/02/2022 for the final demonstration.


Design
Marketing
Admin
R1
R2
220.3.4.4/30
2200:3:4:4::4/126
ISP
Internet
203.0.0.0/24
2300:A:A:A::/64