Universidade do Minho

João Carlos Delgado Monteiro

Relatório

1.
a) Imagem
b) Imagem
c) Relativamente aos pacotes ICMP enviados pelo nodo 1, verifica-se que este enviou o primeiro pacote com o valor de TTL (Time To Live) igual a 1, o segundo com o TTL igual a 2, o terceiro com TTL igual a 3 e assim sucessivamente. Qualquer pacote que dá entrada num router vê o seu TTL reduzido em uma unidade e, caso esse valor (após o decremento) seja igual a zero, este envia um pacote ICMP do tipo TTL exceed ao emissor, cada um dos routers presentes na topologia enviou ao nodo 1 uma mensagem ICMP do tipo referido anteriormente. Os pacotes ICMP posteriormente recebidos têm já, como emissor, o nodo servidor, comunicando ao cliente que a porta escolhida para estabelecer a ligação é inalcançável.

d) TTL = 4
Tempo médio: 0,000139s

2.
e) TTL e header checksum
f) tamanho trama, a versao do ip, o protocolo, o fragment offset
g) O identification, inicialmente a 0x8743 (34627)
h) o ttl varia entre 1 e 117 e identificação 0x4a7e (19070)
i)

3.
a) A mensagem foi fragmentada o suficiente para poder ser transmitida por uma conexão com o MTU menor que o datagrama original.

b) O valor do bit More fragments do campo Flags é 1, indicando que existem mais fragmentos relativos ao datagrama original, ou seja, o datagrama foi fragmentado.
Podemos ver que é o primeiro fragmento através do Fragment offset que é zero.
O tamanho é de 1500 bytes.

c) Neste segundo fragmento, o valor do campo Fragment offset é maior do que zero, concluindo-se que não se trata do primeiro fragmento.
O bit More fragments está a 0, logo não existem mais fragmentos.

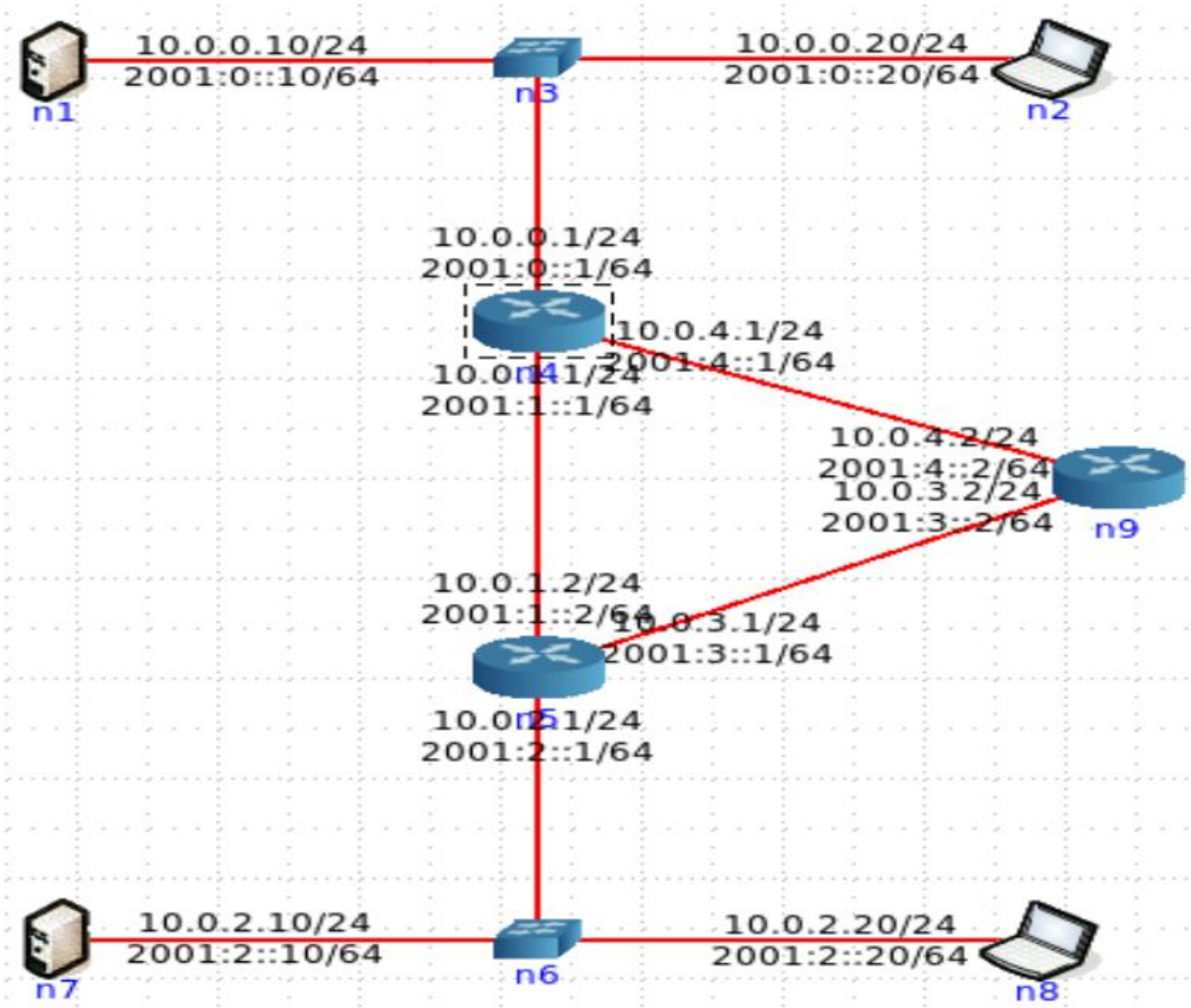
d) Foram criados 3 fragmentos.

e) Analisado o wireshark, concluiu-se que os campos que mudam são: fragment offset, length,

flag e header checksum.

parte2

1.
a



Através da imagem pode-se ver facilmente os IPs de cada equipamento. Os endereços IP são de classe A porque o valor do primeiro octecto (10) está entre 0 e 126.
Quanto às máscaras de rede criadas pelo core são 255.255.255.000, tal como mostra a imagem exemplo em baixo. [XXXX LATEX XXXX]

```
eth0      Link encap:Ethernet  HWaddr 00:00:00:aa:00:02
          inet addr:10.0.0.1  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:feaa:2/64 Scope:Link
          inet6 addr: 2001::1/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1720 (1.7 KB)  TX bytes:1220 (1.2 KB)

eth1      Link encap:Ethernet  HWaddr 00:00:00:aa:00:03
          inet addr:10.0.1.1  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:feaa:3/64 Scope:Link
          inet6 addr: 2001:1::1/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2074 (2.0 KB)  TX bytes:1228 (1.2 KB)

eth2      Link encap:Ethernet  HWaddr 00:00:00:aa:00:0a
          inet addr:10.0.4.1  Bcast:0.0.0.0  Mask:255.255.255.0
```

b

O CORE atribui automaticamente endereços privados. Estes encontram-se nas gamas de valores privadas [10.0.0.0 – 10.255.255.255] 10.0.0.0/8.

c

Não são atribuídos endereços IP aos switches, pois na camada protocolar estes pertence à 2.ª, camada de enlace. Nesta ainda não estão não há conexões entre as redes locais.

d

Fazendo a ligação entre os dois pisos é notória a ligação.

```
root@n1:/tmp/pycore.33218/n1.conf# ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_req=1 ttl=62 time=0.230 ms
64 bytes from 10.0.2.10: icmp_req=2 ttl=62 time=0.216 ms
64 bytes from 10.0.2.10: icmp_req=3 ttl=62 time=0.188 ms
64 bytes from 10.0.2.10: icmp_req=4 ttl=62 time=0.352 ms
64 bytes from 10.0.2.10: icmp_req=5 ttl=62 time=0.152 ms
64 bytes from 10.0.2.10: icmp_req=6 ttl=62 time=0.113 ms
64 bytes from 10.0.2.10: icmp_req=7 ttl=62 time=0.357 ms
```

Ligação Host-to-Host:

Ligação Laptop-to-Laptop:

3

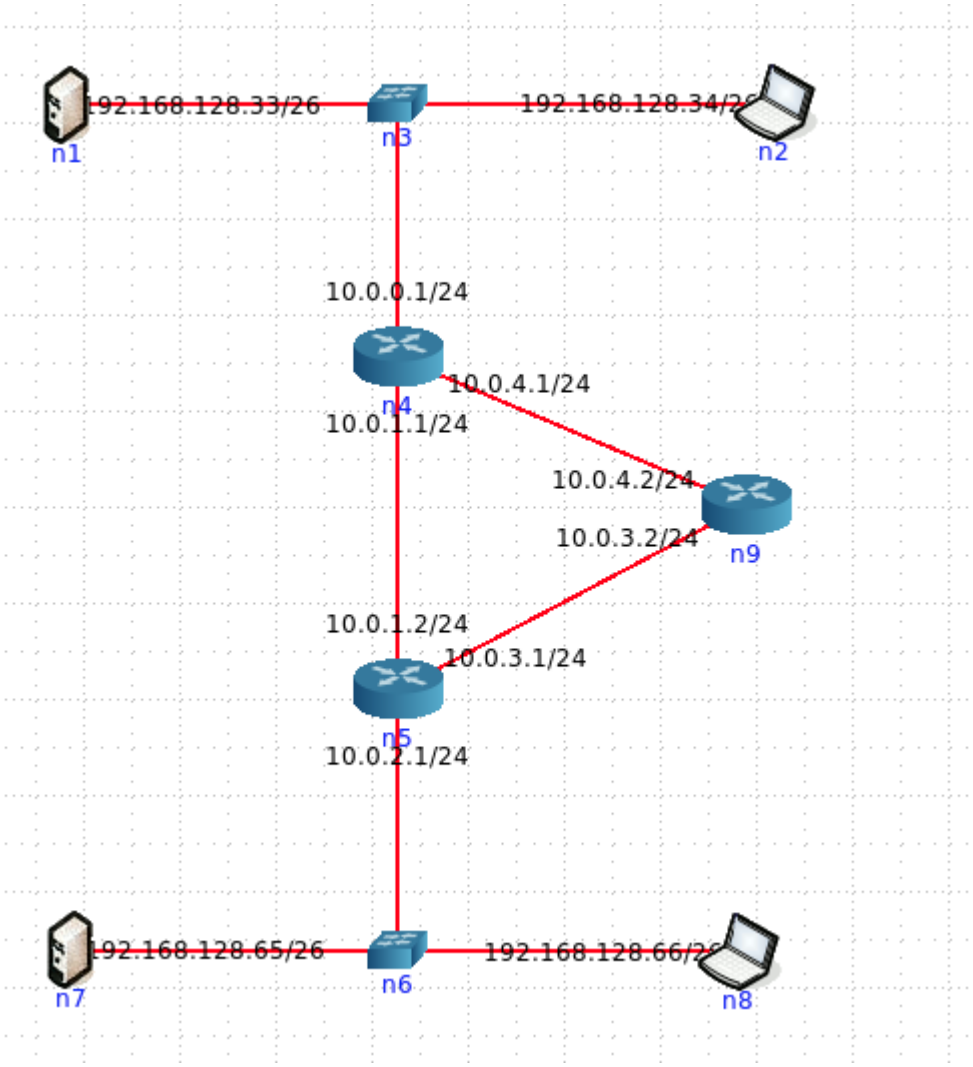
3.1) Foram criadas 2 sub-redes apartir da rede 192.168.128.0/24, respetivos aos 2

```
root@n2:/tmp/pycore.33218/n2.conf# ping 10.0.2.20
PING 10.0.2.20 (10.0.2.20) 56(84) bytes of data.
64 bytes from 10.0.2.20: icmp_req=1 ttl=62 time=0.616 ms
64 bytes from 10.0.2.20: icmp_req=2 ttl=62 time=0.135 ms
64 bytes from 10.0.2.20: icmp_req=3 ttl=62 time=0.308 ms
64 bytes from 10.0.2.20: icmp_req=4 ttl=62 time=0.135 ms
64 bytes from 10.0.2.20: icmp_req=5 ttl=62 time=0.118 ms
64 bytes from 10.0.2.20: icmp_req=6 ttl=62 time=0.331 ms
64 bytes from 10.0.2.20: icmp_req=7 ttl=62 time=0.368 ms
```

departamentos. A criação dessas 2 sub-redes corresponde acrescentar, no mínimo, mais 2 bits para a máscara de rede, que passou a ser de 26 bits. As redes criadas foram:

-> 192.168.128.32/26

-> 192.168.128.64/26



3.2) A máscara de rede a usar é o 255.255.255.192, por ser 26 bits.

3.3) Não é possível anunciar o prefixo das redes do departamento para o exterior.

3.4)Pode ser exportada para o exterior o prefixo 192.168.128.0/24, pois do exterior as redes da empresa são vistas como uma rede única.

3.5)Sendo uma máscara com 26 bits, teremos 10 bits para subnetting e 6 bits para os hosts. Com isso, o número de hosts que se pode interligar em cada departamento seria 2^6 - 2 = 62.

3.6) A conectividade foi mantida.

Conclusão

Na elaboração deste trabalho prático aprofundamos os nossos conhecimentos sobre o Internet&Protocol (IP).
Analisamos no CORE a realização da troca de mensagens entre origens e destinos através IP. Verificamos também, o funcionamento do protocolo ICMP e IPv4. Compreendemos a fragmentação de mensagens.
Já na segunda parte do trabalho prático, fizemos um caso exemplo usando uma topologia no CORE, onde foi possível verificar as “ligações entre pisos”. Desde tabelas de routing e alterações de endereços de rede.