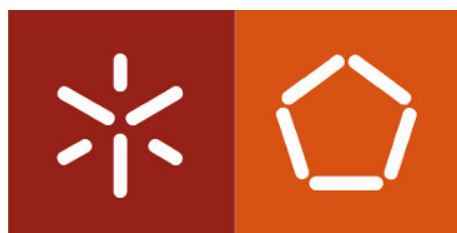


Redes de Computadores

11 de Novembro de 2019

Grupo nr. 15

a83899	André Morais
a85954	Luís Ribeiro
a84783	Pedro Rodrigues



Mestrado Integrado em Engenharia Informática
Universidade do Minho

Conteúdo

1	Introdução	2
2	Questões e Respostas	3
2.1	Captura e Análise de tramas Ethernet	3
2.2	Protocolo ARP	6
2.3	Domínios de Colisão	10

1 Introdução

2 Questões e Respostas

2.1 Captura e Análise de tramas Ethernet

Ative o Wireshark na sua máquina nativa e acesse ao URL *http://miei.di.uminho.pt*.

Obtenha o número de ordem da sequência de bytes capturada correspondente à mensagem HTTP GET enviada pelo seu computador para o servidor Web, bem como o endereço da respectiva mensagem HTTP Response proveniente do servidor.

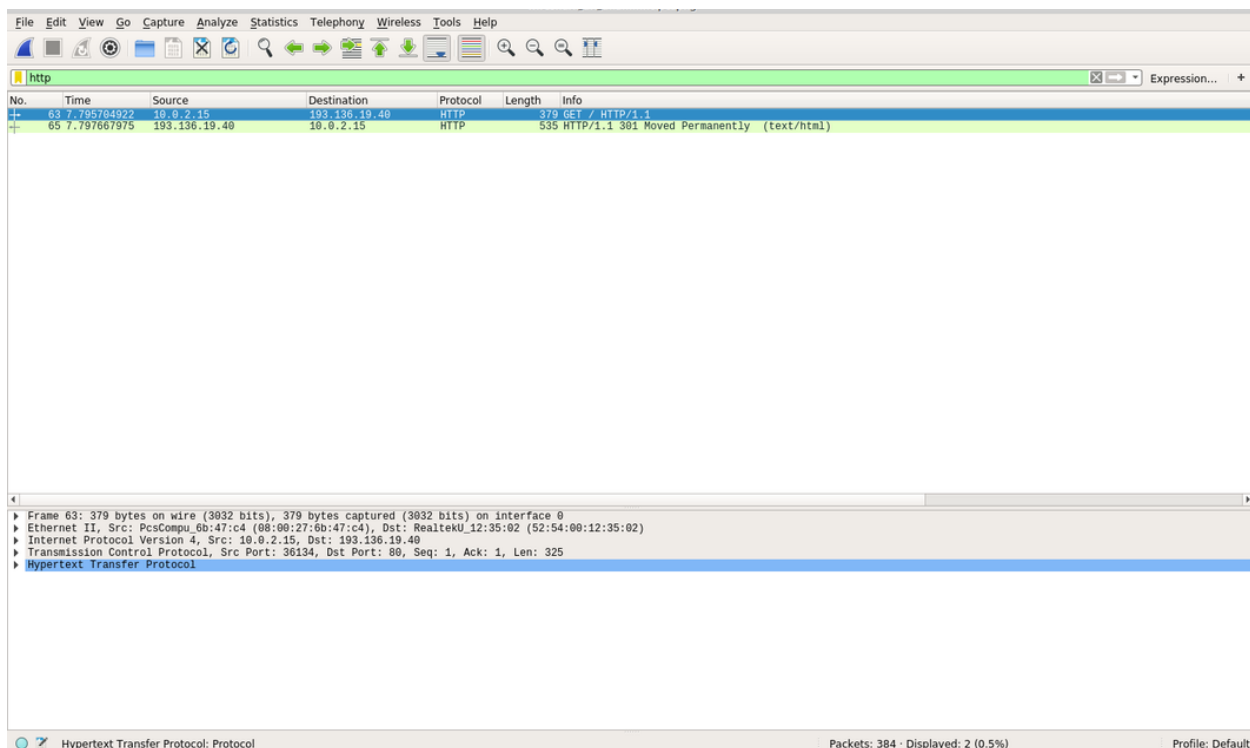


Figura 1: Captura de mensagens HTTP

Sequência de bytes *GET* é **63** e do *Response* é **65**.

1. Anote os endereços MAC de origem e de destino da trama capturada.

```
▶ Frame 63: 379 bytes on wire (3032 bits), 379 bytes captured (3032 bits) on interface 0
▼ Ethernet II, Src: PcsCompu_6b:47:c4 (08:00:27:6b:47:c4), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
  ▼ Destination: RealtekU_12:35:02 (52:54:00:12:35:02)
    Address: RealtekU_12:35:02 (52:54:00:12:35:02)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: PcsCompu_6b:47:c4 (08:00:27:6b:47:c4)
    Address: PcsCompu_6b:47:c4 (08:00:27:6b:47:c4)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.19.40
▶ Transmission Control Protocol, Src Port: 36134, Dst Port: 80, Seq: 1, Ack: 1, Len: 325
▶ Hypertext Transfer Protocol
```

Figura 2: Endereços MAC de origem e destino

Endereço MAC de origem: 08:00:27:6b:47:c4

Endereço MAC de destino: 52.54.00.12.35.02

2. Identifique a que sistemas se referem. Justifique.

R: Como selecionamos a trama da mensagem GET, o endereço de origem representa o local de onde é enviada a mensagem, ou seja, representa a **interface ethernet da nossa máquina**. O endereço de destino corresponde **à interface do router da rede**.

3. Qual o valor hexadecimal do campo *Type* da trama Ethernet? O que significa?

R: Como podemos ver na Figura 2, o valor do campo *Type* é **0x0800** e indica que encapsula um pacote IPv4.

4. Quantos bytes são usados desde o início da trama até ao caractere ASCII “G” do método HTTP GET? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar no envio do HTTP GET.

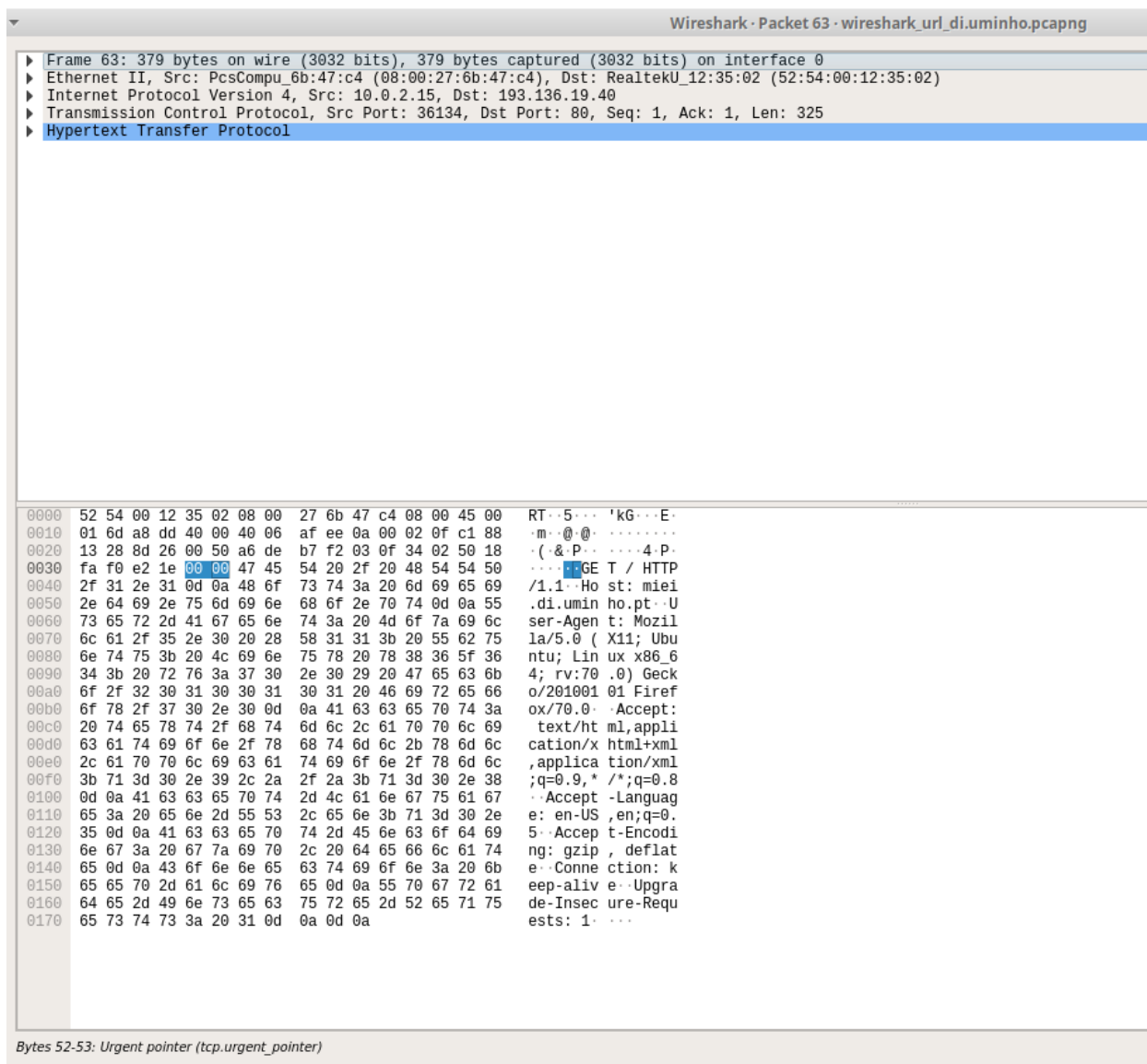


Figura 3: Trama da mensagem HTTP GET

O número de bytes até ao caractere ASCII "G" é **54**.

$$\frac{54}{379} \times 100 = 14.25\%$$

- Através de visualização direta de uma trama capturada, verifique que, possivelmente, o campo FCS (Frame Check Sequence) usado para deteção de erros não está a ser usado. Em sua opinião, porque será?

R: O campo FCS é utilizado para detecção de erros. Neste caso este campo não é utilizado porque numa ligação Ethernet a probabilidade de haver erros é muito reduzida.

6. Qual é o endereço Ethernet da fonte? A que sistema de rede corresponde? Justifique.

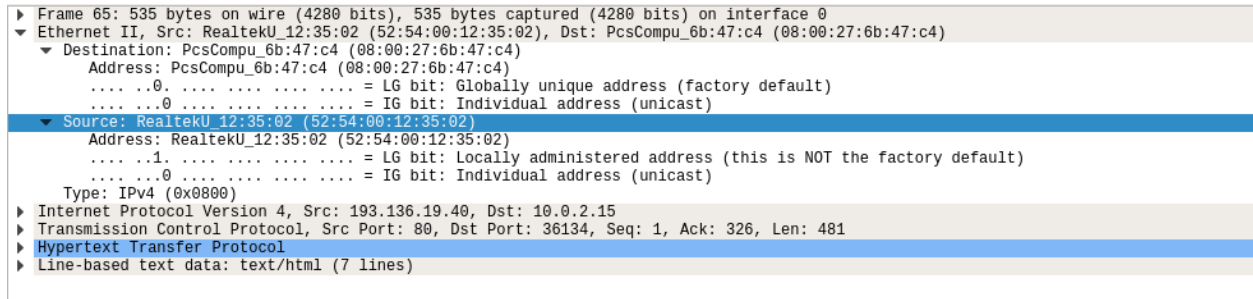


Figura 4: Endereço ethernet da fonte

Endereço da fonte: **52:54:00:12:35:02**, corresponde ao gateway da rede.

7. Qual é o endereço MAC do destino? A que sistema corresponde?

R: Endereço MAC do destino: **08:00:27:6b:47:c4**, corresponde à interface ethernet da nossa máquina.

8. Atendendo ao conceito de desencapsulamento protocolar, identifique os vários protocolos contidos na trama recebida.

R: Ethernet, IPv4, TCP e HTTP.

2.2 Protocolo ARP

9. Observe o conteúdo da tabela ARP. Explique o significado de cada uma das colunas.

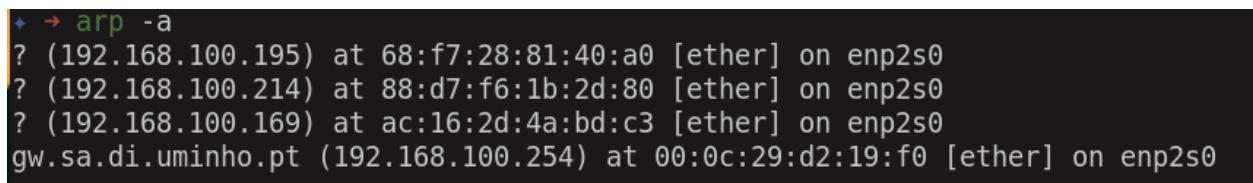


Figura 5: Resultado da execução do comando *arp -a*

A primeira coluna identifica o *host*, a segunda coluna representa o endereço ip do mesmo, a terceira o MAC address e as colunas seguintes indicam o tipo de ligação e a interface.

```
~
♦ → sudo arp -d 192.168.100.195

~
♦ → sudo arp -d 192.168.100.214

~
♦ → sudo arp -d 192.168.100.169

~
♦ → arp -a
gw.sa.di.uminho.pt (192.168.100.254) at 00:0c:29:d2:19:f0 [ether] on enp2s0
```

Figura 6: Eliminação de linhas da tabelas ARP

10. Qual é o valor hexadecimal dos endereços origem e destino na trama Ethernet que contém a mensagem com o pedido ARP (ARP Request)? Como interpreta e justifica o endereço destino usado?

```
▶ Frame 688: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: AsustekC_2c:42:59 (88:d7:f6:2c:42:59), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: AsustekC_2c:42:59 (88:d7:f6:2c:42:59)
    Type: ARP (0x0806)
▶ Address Resolution Protocol (request)
```

Figura 7: Endereços na trama Ethernet com o pedido ARP

Origem: 88:d7:f6:2c:42:59

Destino: ff:ff:ff:ff:ff:ff

O endereço de destino usado é o endereço de broadcast para ser enviado para todos os *hosts* da rede.

11. Qual o valor hexadecimal do campo tipo da trama Ethernet? O que indica?

R: Como podemos ver na Figura 7, o campo *Type* é **0x0806**, e indica que encapsula um frame ARP.

12. Qual o valor do campo ARP opcode? O que especifica?

```
▶ Frame 688: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▶ Ethernet II, Src: AsustekC_2c:42:59 (88:d7:f6:2c:42:59), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: AsustekC_2c:42:59 (88:d7:f6:2c:42:59)
  Sender IP address: 192.168.100.197
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.224
```

Figura 8: Valor do campo ARP opcode

O valor do opcode é *request* (1) e mostra que a trama selecionada representa um request ARP.

13. Identifique que tipo de endereços está contido na mensagem ARP? Que conclui?

R: Analisando a Figura 8 verificamos que os endereços contidos na mensagem ARP são **Sender MAC address**, **Sender IP address**, **Target MAC address** e **Target IP address**.

O *host* com endereço de IP 192.168.100.197 e MAC address 88:d7:f6:2c:42:59 pergunta à rede (*broadcast*) qual é o MAC address do endereço de IP 192.168.100.224.

14. Explícite que tipo de pedido ou pergunta é feito pelo host de origem?

arp					
No.	Time	Source	Destination	Protocol	Length Info
2	0.454296663	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
3	1.453789933	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
104	11.740636170	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
107	12.741585081	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
111	13.740716358	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
625	19.936559952	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
659	20.936224049	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
673	21.936725745	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
688	24.906432222	AsustekC_2c:42:59	Broadcast	ARP	42 Who has 192.168.100.224? Tell 192.168.100.197
689	24.907261750	AsustekC_37:e5:63	AsustekC_2c:42:59	ARP	60 192.168.100.224 is at 70:4d:7b:37:e5:63
712	28.501672903	AsustekC_1b:5c:4b	Broadcast	ARP	60 Who has 192.168.100.254? Tell 192.168.100.161
725	30.111731864	AsustekC_37:e5:63	AsustekC_2c:42:59	ARP	60 Who has 192.168.100.197? Tell 192.168.100.224
726	30.111762099	AsustekC_2c:42:59	AsustekC_37:e5:63	ARP	42 192.168.100.197 is at 88:d7:f6:2c:42:59
727	30.158812692	Vmware_d2:19:f0	AsustekC_2c:42:59	ARP	60 Who has 192.168.100.197? Tell 192.168.100.254
728	30.158838607	AsustekC_2c:42:59	Vmware_d2:19:f0	ARP	42 192.168.100.197 is at 88:d7:f6:2c:42:59
741	32.219432346	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
754	33.218918454	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
757	33.376814925	RealtekS_64:b8:08	Broadcast	ARP	60 Who has 192.168.100.254? Tell 192.168.100.226
758	33.395453994	RealtekS_64:b8:08	Broadcast	ARP	60 Who has 192.168.100.254? Tell 192.168.100.226
764	33.633678892	RealtekS_64:b8:08	Broadcast	ARP	60 Who has 192.168.100.226? Tell 0.0.0.0
776	34.220520605	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
801	39.694756431	AsustekC_1b:5c:4b	Broadcast	ARP	60 Who has 192.168.100.254? Tell 192.168.100.161
831	40.411357947	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.185? Tell 192.168.100.254
833	40.634573038	RealtekS_64:b8:08	Broadcast	ARP	60 Who has 192.168.100.226? Tell 0.0.0.0
840	41.133632907	RealtekS_64:b8:08	Broadcast	ARP	60 Who has 192.168.100.254? Tell 192.168.100.226

Figura 9: Exemplo das mensagens ARP

Como podemos ver pela Figura 9, a mensagem enviada é "Who has *destination*? Tell *source*". No nosso exemplo, a mensagem é "Who has 192.168.100.224? Tell 192.168.100.197"

15. Localize a mensagem ARP que é a resposta ao pedido ARP efetuado.

▶ Frame 689: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶ Ethernet II, Src: AsustekC_37:e5:63 (70:4d:7b:37:e5:63), Dst: AsustekC_2c:42:59 (88:d7:f6:2c:42:59)
▼ Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: AsustekC_37:e5:63 (70:4d:7b:37:e5:63)
Sender IP address: 192.168.100.224
Target MAC address: AsustekC_2c:42:59 (88:d7:f6:2c:42:59)
Target IP address: 192.168.100.197

Figura 10: Resposta ao pedido ARP efetuado

(a) Qual o valor do campo ARP opcode? O que especifica?

R: O valor do campo ARP opcode é *reply (2)*, mostra que a mensagem é uma resposta.

(b) Em que posição da mensagem ARP está a resposta ao pedido ARP?

R: Como vimos acima, o nosso pedido era o MAC address do IP 192.168.100.224, então, a resposta está presente no **Sender MAC address**.

16. Identifique um pacote de pedido ARP gratuito originado pelo seu sistema. Analise o conteúdo de um pedido ARP gratuito e identifique em que se distingue dos restantes pedidos ARP. Registe a trama Ethernet correspondente. Qual o resultado esperado face ao pedido ARP gratuito enviado?

```
► Frame 30: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
► Ethernet II, Src: Apple_23:a1:df (38:c9:86:23:a1:df), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▼ Address Resolution Protocol (request/gratuitous ARP)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  [Is gratuitous: True]
  Sender MAC address: Apple_23:a1:df (38:c9:86:23:a1:df)
  Sender IP address: 192.168.100.150
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.150
```

Figura 11: Pacote do pedido ARP gratuito

2.3 Domínios de Colisão

Construa uma topologia no emulador CORE com um host (n1) e dois servidores (n2, n3) interligados através de um hub.

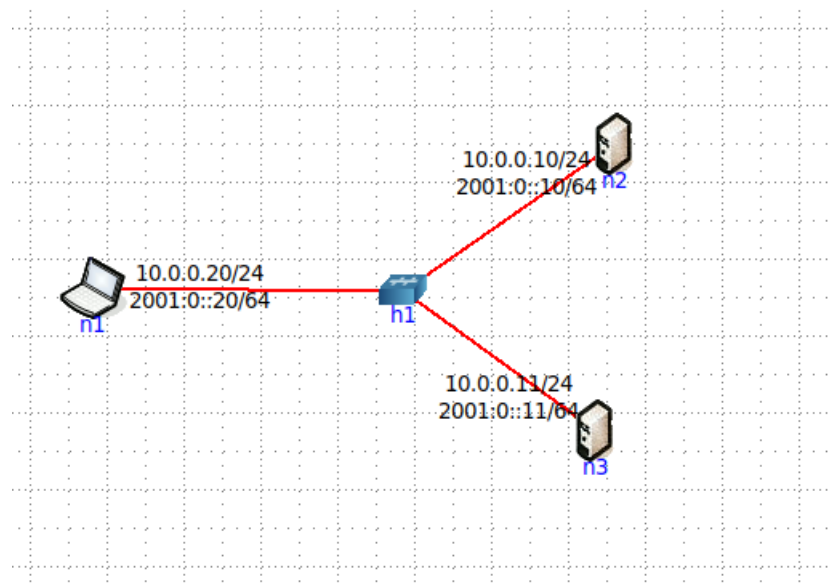
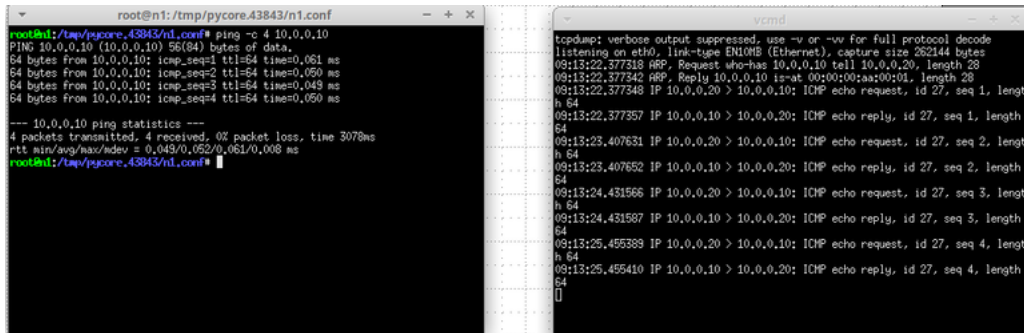


Figura 12: Topologia *CORE* com nodos interligados através de um *hub*

17. Faça ping de n1 para n2. Verifique com a opção tcpdump como flui o tráfego nas diversas interfaces dos vários dispositivos. Que conclui?



```
root@n1: /tmp/pycore.43843/n1.conf
root@n1: /tmp/pycore.43843/n1.conf# ping -c 4 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data:
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=0.061 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.050 ms

--- 10.0.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/ndev = 0.049/0.052/0.061/0.008 ms
root@n1: /tmp/pycore.43843/n1.conf#

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:13:22.377318 ARP, Request who-has 10.0.0.10 tell 10.0.0.20, length 28
09:13:22.377348 ARP, Reply 10.0.0.10 is-at 00:00:00:aa:00:01, length 28
09:13:22.377348 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 27, seq 1, length 64
09:13:22.377357 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 27, seq 1, length 64
09:13:23.407631 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 27, seq 2, length 64
09:13:23.407652 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 27, seq 2, length 64
09:13:24.431566 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 27, seq 3, length 64
09:13:24.431587 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 27, seq 3, length 64
09:13:25.455389 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 27, seq 4, length 64
09:13:25.455410 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 27, seq 4, length 64
```

Figura 13: Resultado do ping de n1 para n2 e análise de tráfego em n3

Analisando o tráfego em n3 que está fora da comunicação, reparamos que ele recebe tráfego na mesma, ou seja, quando temos uma *hub* ele envia os pacotes para todos os *hosts* presentes na rede. Isto aumenta o risco de colisão.

18. Na topologia de rede substitua o hub por um switch. Repita os procedimentos que realizou na pergunta anterior. Comente os resultados obtidos quanto à utilização de hubs e switches no contexto de controlar ou dividir domínios de colisão. Documente as suas observações e conclusões com base no tráfego observado/capturado.

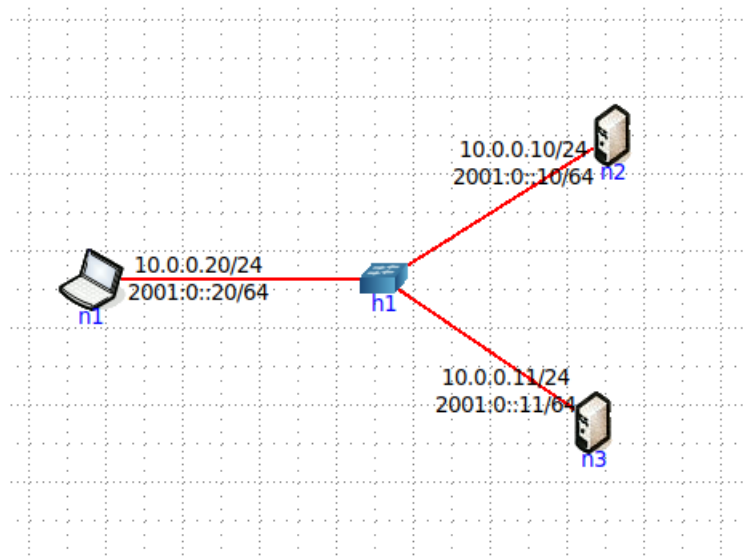
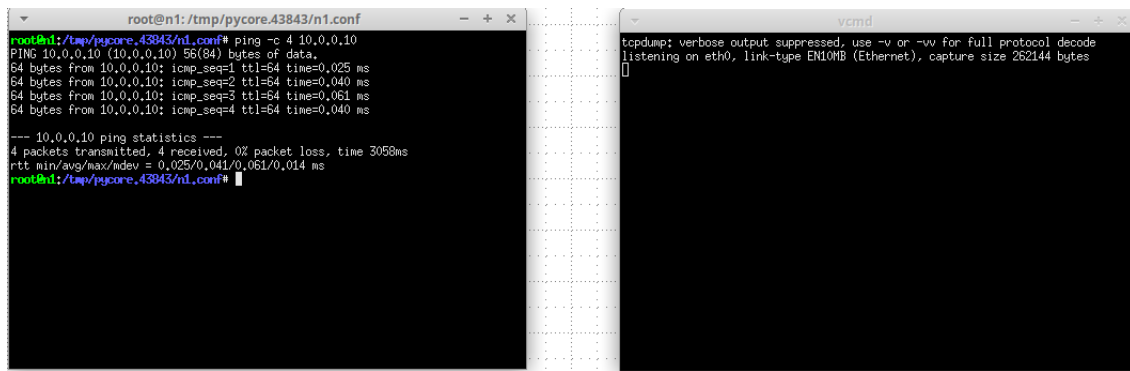


Figura 14: Topologia *CORE* com nodos interligados através de um *switch*



```
root@n1:/tmp/pycore.43843/n1.conf# ping -c 4 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data:
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.061 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.040 ms

--- 10.0.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3058ms
rtt min/avg/max/mdev = 0.025/0.041/0.061/0.014 ms
root@n1:/tmp/pycore.43843/n1.conf#
```

```
vcmd
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
[]
```

Figura 15: Resultado do ping de n1 para n2 e análise de tráfego em n3

Com o *switch*, ao analisar o tráfego em n3, reparamos que não recebe tráfego nenhum. Isto acontece porque nos *hubs* o pedido vai fluir por todas as máquinas conectadas, no caso do *switch* apenas passa pelas máquinas intervenientes na comunicação. Assim, há menos probabilidade de colisões com a utilização de *switches*