

Alunos: João F. Carvalho, Jorge C. da Cunha, Matheus J. da Cunha

Professor: Felipe Viel



### **Avaliação – Camada de Enlace**

Este relatório tem por finalidade descrever e documentar os processos efetuados para simular a camada de enlace, conforme projetos passados e responder as perguntas que estão nos roteiros.

#### **Projeto 1- Wireshark Lab: 802.11**

Este roteiro foi executado baseado no arquivo da documentação Wireshark\_802\_11.pcap que foi aberto no software wireshark e realizado as análises necessários para as questões abaixo.

1. What are the SSIDs of the two access points that are issuing most of the beacon frames in this trace?

R: Os SSID são 30 Munroe ST e Linksys\_SES\_24086

2. What are the intervals of time between the transmissions of the beacon frames the linksys\_ses\_24086 access point? From the 30 Munroe St. access point? (Hint: this interval of time is contained in the beacon frame itself).

R: O intervalo é 1024 segundos.

3. What (in hexadecimal notation) is the source MAC address on the beacon frame from 30 Munroe St? Recall from figure 7.13 in the text that the source, destination, and BSS are three addresses used in an 802.11 frame. For a detailed discussion of the 802.11 frame structure, see section 7 in the IEEE 802.11 standards document (cited above).

R: O endereço físico é o 00:16:b6:f7:1d:51, conforme Imagem 1.

```
> Frame 291: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits)
> Radiotap Header v0, Length 24
> 802.11 radio information
▼ IEEE 802.11 Beacon frame, Flags: .....C
  Type/Subtype: Beacon frame (0x0008)
  > Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    Source address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    .... .... 0000 = Fragment number: 0
    1011 1011 1110 .... = Sequence number: 3006
    Frame check sequence: 0x0a05fd72 [unverified]
    [FCS Status: Unverified]
  > IEEE 802.11 Wireless Management
```

Imagem 1 - Evidência do endereço físico.

4. What (in hexadecimal notation) is the destination MAC address on the beacon frame from 30 Munroe St??

R: O endereço físico de destino é o ff:ff:ff:ff:ff:ff, conforme Imagem 2.

```
> Frame 291: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits)
> Radiotap Header v0, Length 24
> 802.11 radio information
▼ IEEE 802.11 Beacon frame, Flags: .....C
  Type/Subtype: Beacon frame (0x0008)
  > Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    Source address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    .... .... 0000 = Fragment number: 0
    1011 1011 1110 .... = Sequence number: 3006
    Frame check sequence: 0x0a05fd72 [unverified]
    [FCS Status: Unverified]
  > IEEE 802.11 Wireless Management
```

Imagem 2 – Evidência do endereço físico de destino.

5. What (in hexadecimal notation) is the MAC BSS id on the beacon frame from 30 Munroe St?

R: O endereço físico BSS é o 00:16:b6:f7:1d:51, conforme Imagem 3.

```
> Frame 38: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits)
> Radiotap Header v0, Length 24
> 802.11 radio information
▼ IEEE 802.11 Beacon frame, Flags: .....C
    Type/Subtype: Beacon frame (0x0008)
    > Frame Control Field: 0x8000
        .000 0000 0000 0000 = Duration: 0 microseconds
        Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
        Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
        Transmitter address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
        Source address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
        BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
        .... .... 0000 = Fragment number: 0
        1011 0011 1001 .... = Sequence number: 2873
        Frame check sequence: 0x03f297db [unverified]
        [FCS Status: Unverified]
    > IEEE 802.11 Wireless Management
```

Imagem 3 – Evidência do endereço físico BSS.

6. The beacon frames from the 30 Munroe St access point advertise that the access point can support four data rates and eight additional “extended supported rates.” What are these rates?

R: As taxas de suporte são: 1.0, 2.0, 5.5, 11.0 Mbps, conforme Imagem 4. As taxas estendidas são: 6.0, 9.0, 12.0, 18.0, 24.0, 36.0, 48.0 e 54.0 Mbps, conforme Imagem 5.

```
▼ Tagged parameters (119 bytes)
    ▼ Tag: SSID parameter set: 30 Munroe St
        Tag Number: SSID parameter set (0)
        Tag length: 12
        SSID: 30 Munroe St
    ▼ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), [Mbit/sec]
        Tag Number: Supported Rates (1)
        Tag length: 4
        Supported Rates: 1(B) (0x82)
        Supported Rates: 2(B) (0x84)
        Supported Rates: 5.5(B) (0x8b)
        Supported Rates: 11(B) (0x96)
```

Imagem 4 – Evidência das taxas de suporte.

Tag: Extended Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]  
 Tag Number: Extended Supported Rates (50)  
 Tag length: 8  
 Extended Supported Rates: 6(B) (0x8c)  
 Extended Supported Rates: 9 (0x12)  
 Extended Supported Rates: 12(B) (0x98)  
 Extended Supported Rates: 18 (0x24)  
 Extended Supported Rates: 24(B) (0xb0)  
 Extended Supported Rates: 36 (0x48)  
 Extended Supported Rates: 48 (0x60)  
 Extended Supported Rates: 54 (0x6c)

Imagem 5 – Evidência das taxas estendidas.

- Find the 802.11 frame containing the SYN TCP segment for this first TCP session (that downloads alice.txt). What are three MAC address fields in the 802.11 frame? Which MAC address in this frame corresponds to the wireless host (give the hexadecimal representation of the MAC address for the host)? To the access point? To the first-hop router? What is the IP address of the wireless host sending this TCP segment? What is the destination IP address? Does this destination IP address correspond to the host, access point, first-hop router, or some other network-attached device? Explain.

R: Os endereços MACs presentes em 802.11 são: BSS ID, Source e Destination. O endereço MAC do host wireless é o 00:13:02:d1:b6:4f. O endereço físico do ponto de acesso é o 00:16:b6:f4:eb:a8. Correspondente ao host sem fio que envia este segmento TCP é 00:16:b6:f7:1d:51. O IP correspondente do host sem fio é 192.168.1.109. Do destino o IP é o 128.199.245.12 e este IP corresponde ao host, conforme Imagem 6, Imagem 7 e Imagem 8.

868	25.126724	128.119.245.12	192.168.1.109	HTTP	400 HTTP/1.1 200 OK (text/plain)
-----	-----------	----------------	---------------	------	----------------------------------

Imagem 6 – Evidência dos IPs de origem e destino.

```

> Frame 868: 400 bytes on wire (3200 bits), 400 bytes captured (3200 bits)
> Radiotap Header v0, Length 24
> 802.11 radio information
> IEEE 802.11 QoS Data, Flags: .....F.C
> Logical-Link Control
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.109
> Transmission Control Protocol, Src Port: 80, Dst Port: 2538, Seq: 152154, Ack: 436, Len:
> [106 Reassembled TCP Segments (152451 bytes): #486(313), #488(1460), #501(1460), #504(14
  ✓ Hypertext Transfer Protocol
    > HTTP/1.1 200 OK\r\n
      Date: Fri, 29 Jun 2007 02:05:39 GMT\r
    > y\003\032kr: Apache/2.0.52 (CentOS)\r\n
      Last-Modified: Sat, 21 Aug 2004 14:21:11 GMT\r\n
      ETag: "8734f-2524a-ba3a03c0"\r\n
    > Accept-Rang@s: 6{tes\r\n
    > Content-Length: 152138\r\n
      Keep-Alive: timeout=10, max=100\r\n
      Connection: Keep-Alive\r\n
      Content-Type: text/plain; charset=ISO-8859-1\r\n
      \r\n
      [HTTP response 1/2]
      [Time since request: 0.298471000 seconds]
      [Request in frame: 480]
      [Next request in frame: 873]
      [Next response in frame: 875]
      [Request URI: http://gaia.cs.umass.edu/wireshark-labs/alice.txt]
      File Data: 152138 bytes

```

Imagem 7 – Datagrama expandido do HTTP.

```

> Frame 868: 400 bytes on wire (3200 bits), 400 bytes captured (3200 bits)
> Radiotap Header v0, Length 24
> 802.11 radio information
  ✓ IEEE 802.11 QoS Data, Flags: .....F.C
    Type/Subtype: QoS Data (0x0028)
    > Frame Control Field: 0x8802
      .000 0000 0010 1000 = Duration: 40 microseconds
      Receiver address: IntelCor_d1:b6:4f (00:13:02:d1:b6:4f)
      Transmitter address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
      Destination address: IntelCor_d1:b6:4f (00:13:02:d1:b6:4f)
      Source address: Cisco-Li_f4:eb:a8 (00:16:b6:f4:eb:a8)
      BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
      STA address: IntelCor_d1:b6:4f (00:13:02:d1:b6:4f)
      .... 0000 = Fragment number: 0
      1100 1010 0010 .... = Sequence number: 3234
      Frame check sequence: 0xde06ad2 [unverified]
      [FCS Status: Unverified]
    > Qos Control: 0x0300

```

Imagem 8 – Evidência dos endereços de origem e destino.

8. Find the 802.11 frame containing the SYNACK segment for this TCP session. What are three MAC address fields in the 802.11 frame? Which MAC address in this frame corresponds to the host? To the access point? To the first-hop router? Does the sender MAC address in the frame correspond to the IP address of the device that sent the TCP segment encapsulated within this datagram? (Hint: review figure 6.19 in the text if you are unsure of how to answer this question, or the

corresponding part of the previous question. It's particularly important that you understand this).

R: BSS id: 00:16:b6:f7:1d:51, Destination: 00:13:02:d1:b6:4f e endereço de origem: 00:16:b6:f4:eb:a8. O MAC corresponde ao host é 00:13:02:d1:b6:4f (destino). Source: 00:16:b6:f4:eb:a8. O endereço MAC do remetente no quadro não corresponde ao endereço IP do dispositivo que enviou o segmento TCP encapsulado dentro deste datagrama, porque o TCP SYNACK O endereço IP é 128.199.245.12, mas o endereço IP de destino é 192.168.1.109. Vide Imagem 9 e Imagem 10.

476	24.827751	128.119.245.12	192.168.1.109	TCP	110	80 → 2538	[SYN, ACK]	Seq=0	Ack=1	Win=5840	Len=0	SACK_PERM=1
-----	-----------	----------------	---------------	-----	-----	-----------	------------	-------	-------	----------	-------	-------------

Imagem 9 – Evidência do endereço de destino.

```
> Frame 476: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
> Radiotap Header v0, Length 24
> 802.11 radio information
▼ IEEE 802.11 QoS Data, Flags: ..mP..F.C
  Type/Subtype: QoS Data (0x0028)
  > Frame Control Field: 0x8832
  Duration/ID: 11560 (reserved)
  Receiver address: 91:2a:b0:49:b6:4f (91:2a:b0:49:b6:4f)
  Transmitter address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
  Destination address: 91:2a:b0:49:b6:4f (91:2a:b0:49:b6:4f)
  Source address: Cisco-Li_f4:eb:a8 (00:16:b6:f4:eb:a8)
  BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
  STA address: 91:2a:b0:49:b6:4f (91:2a:b0:49:b6:4f)
  .... 0000 = Fragment number: 0
  1100 0011 0100 .... = Sequence number: 3124
  Frame check sequence: 0xecdc407d [unverified]
  [FCS Status: Unverified]
  > Qos Control: 0x0100
> Logical-Link Control
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.109
> Transmission Control Protocol, Src Port: 80, Dst Port: 2538, Seq: 0, Ack: 1, Len: 0
```

Imagem 10 – Evidência do datagrama que mostra os endereços de destino e origem.

9. What two actions are taken (i.e., frames are sent) by the host in the trace just after t=49, to end the association with the 30 Munroe St AP that was initially in place when trace collection began? (Hint: one is an IP-layer action, and one is an 802.11-layer action). Looking at the 802.11 specification, is there another frame that you might have expected to see, but don't see here?

R: Um DHCP é enviado para 192.168.1.1. O host envia um quadro DEAUTHENTICATION após 0,02s, conforme Imagem 11.

1732 49.542481	Cisco-Li_f7:1d:51	Broadcast	802.11	183 Beacon frame, SN=3588, FN=0, Flags=.....C, BI=100, SSID=30 Munroe St
1733 49.583615	192.168.1.109	DHCP	390	DHCP Release - Transaction ID 0xea5a526
1734 49.583771		IntelCor_d1:b6:4f (-	802.11	38 Acknowledgement, Flags=.....C
1735 49.609617	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	54 Deauthentication, SN=1605, FN=0, Flags=.....C
1736 49.609770		IntelCor_d1:b6:4f (-	802.11	38 Acknowledgement, Flags=.....C
1737 49.614478	IntelCor_d1:b6:4f	Broadcast	802.11	99 Probe Request, SN=1606, FN=0, Flags=.....C, SSID=linksys_SES_24086
1738 49.615869		Cisco-Li_f5:ba:bb (-	802.11	38 Acknowledgement, Flags=.....C
1739 49.617713		Cisco-Li_f5:ba:bb (-	802.11	38 Acknowledgement, Flags=.....C
1740 49.638857	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=.....C
1741 49.639700	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=.....R...C

Imagem 11 – Evidência das duas ações identificadas na simulação.

10. Examine the trace file and look for AUTHENTICATION frames sent from the host to an AP and vice versa. How many AUTHENTICATION messages are sent from the wireless host to the linksys\_ses\_24086 AP (which has a MAC address of Cisco\_Li\_f5:ba:bb) starting at around t=49? .

R: São enviadas no total 17 mensagens, conforme imagem 12.

2160 63.169707	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	58 Authentication, SN=1647, FN=0, Flags=....R...C
2156 63.168087	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	58 Authentication, SN=1647, FN=0, Flags=.....C
2124 62.174070	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1644, FN=0, Flags=....R...C
2123 62.172946	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1644, FN=0, Flags=....R...C
2122 62.171951	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1644, FN=0, Flags=.....C
1924 57.896970	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1619, FN=0, Flags=....R...C
1923 57.891321	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1619, FN=0, Flags=....R...C
1922 57.890325	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1619, FN=0, Flags=....R...C
1921 57.889232	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1619, FN=0, Flags=.....C
1822 53.787070	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1612, FN=0, Flags=....R...C
1821 53.785833	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1612, FN=0, Flags=.....C
1749 49.649705	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=....R...C
1746 49.645319	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=....R...C
1744 49.642315	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=....R...C
1742 49.640702	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=....R...C
1741 49.639700	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=....R...C
1740 49.638857	IntelCor_d1:b6:4f	Cisco-Li_f5:ba:bb	802.11	58 Authentication, SN=1606, FN=0, Flags=.....C

Imagem 12 – Evidência do total de mensagens enviadas na simulação.

11. Does the host want the authentication to require a key or be open?

R: Exige uma autenticação



12. Do you see a reply AUTHENTICATION from the linksys\_ses\_24086 AP in the trace?

R: Não possui retorno.

13. Now let's consider what happens as the host gives up trying to associate with the linksys\_ses\_24086 AP and now tries to associate with the 30 Munroe St AP. Look for AUTHENTICATION frames sent from the host to and AP and vice versa. At what times are there an AUTHENTICATION frame from the host to the 30 Munroe St. AP, and when is there a reply AUTHENTICATION sent from that AP to the host in reply? (Note that you can use the filter expression "wlan.fc.subtype== 11and wlan.fc.type == 0 and wlan.addr == IntelCor\_d1:b6:4f" to display only the AUTHENTICATION frames in this trace for this wireless host.)

R: Há um quadro de AUTENTICAÇÃO de 00:13:02:d1:b6:4f a 00:16:b7:f7:1d:51 quando t = 63.168087. A AUTENTICAÇÃO enviada de volta em t = 63,169071, conforme Imagem 13.

2154 63.142860		Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51) (RA)	802.11	38 Acknowledgement, Flags=.....C
2155 63.161272	Cisco-Li_f7:1d:51	Broadcast	802.11	183 Beacon frame, SN=3725, FN=0, Flags=.....C, BI=100, SSID=30 Munroe St
2156 63.168087	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	58 Authentication, SN=1647, FN=0, Flags=.....C
2157 63.168222		IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C
2158 63.169071	Cisco-Li_f7:1d:51	IntelCor_d1:b6:4f	802.11	58 Authentication, SN=3726, FN=0, Flags=.....C
2159 63.169592		Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51) (RA)	802.11	38 Acknowledgement, Flags=.....C
2160 63.169797	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	58 Authentication, SN=1647, FN=0, Flags=....R...C
2161 63.169814		IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C
2162 63.169910	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	89 Association Request, SN=1648, FN=0, Flags=.....C, SSID=30 Munroe St
2163 63.170008		IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C
2164 63.170692	Cisco-Li_f7:1d:51	IntelCor_d1:b6:4f	802.11	58 Authentication, SN=3727, FN=0, Flags=.....C
2165 63.171000		Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51) (RA)	802.11	38 Acknowledgement, Flags=.....C
2166 63.192101	Cisco-Li_f7:1d:51	IntelCor_d1:b6:4f	802.11	94 Association Response, SN=3728, FN=0, Flags=.....C
2167 63.192956		Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51) (RA)	802.11	38 Acknowledgement, Flags=.....C

Imagem 13 – Evidência do quadro de autenticação.

14. An ASSOCIATE REQUEST from host to AP, and a corresponding ASSOCIATE RESPONSE frame from AP to host are used for the host to associated with an AP. At what time is there an ASSOCIATE REQUEST from host to the 30 Munroe St AP? When is the corresponding ASSOCIATE REPLY sent? (Note that you can use the filter expression "wlan.fc.subtype < 2 and wlan.fc.type == 0 and wlan.addr == IntelCor\_d1:b6:4f" to display only the ASSOCIATE REQUEST and ASSOCIATE RESPONSE frames for this trace.)

R: Associate request do host para o SSID 30 Munroe St em t = 63.169910 e respondeu em t 63.192101, conforme Imagem 14.



2155	63.161272	Cisco-Li_f7:1d:51	Broadcast	802.11	183 Beacon frame, SN=3725, FN=0, Flags=.....C, BI=100, SSID=30 Munroe St
2156	63.168087	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	58 Authentication, SN=1647, FN=0, Flags=.....C
2157	63.168222	IntelCor_d1:b6:4f	IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C
2158	63.169071	Cisco-Li_f7:1d:51	IntelCor_d1:b6:4f	802.11	58 Authentication, SN=3726, FN=0, Flags=.....C
2159	63.169592	Cisco-Li_f7:1d:51	Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51) (RA)	802.11	38 Acknowledgement, Flags=.....C
2160	63.169707	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	58 Authentication, SN=1647, FN=0, Flags=.....R...C
2161	63.169814	IntelCor_d1:b6:4f	IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C
2162	63.169910	IntelCor_d1:b6:4f	Cisco-Li_f7:1d:51	802.11	89 Association Request, SN=1648, FN=0, Flags=.....C, SSID=30 Munroe St
2163	63.170086	IntelCor_d1:b6:4f	IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C
2164	63.170692	Cisco-Li_f7:1d:51	IntelCor_d1:b6:4f	802.11	58 Authentication, SN=3727, FN=0, Flags=.....C
2165	63.171000	Cisco-Li_f7:1d:51	Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51) (RA)	802.11	38 Acknowledgement, Flags=.....C
2166	63.192101	IntelCor_d1:b6:4f	IntelCor_d1:b6:4f	802.11	94 Association Response, SN=3728, FN=0, Flags=.....C
2167	63.192956	Cisco-Li_f7:1d:51	Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51) (RA)	802.11	38 Acknowledgement, Flags=.....C
2168	63.194842	0.0.0.0	255.255.255.255	DHCP	390 DHCP Discover - Transaction ID 0x101b218a
2169	63.194871	IntelCor_d1:b6:4f	IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C
2170	63.201481	0.0.0.0	255.255.255.255	DHCP	390 DHCP Discover - Transaction ID 0x2733a47c
2171	63.201639	0.0.0.0	255.255.255.255	DHCP	390 DHCP Discover - Transaction ID 0x2733a47c
2172	63.201736	IntelCor_d1:b6:4f	IntelCor_d1:b6:4f (00:13:02:d1:b6:4f) (RA)	802.11	38 Acknowledgement, Flags=.....C

Imagem 14 – Evidência das associações e respostas na simulação.

15. What transmission rates is the host willing to use? The AP? To answer this question, you will need to look into the parameters fields of the 802.11 wireless LAN management frame.

R: As taxas possíveis são 1, 2, 5,5, 11, 6, 9, 12, 18, 24, 32, 48, 54 Mbps. Basta analisar as taxas "supported" e "exnteded" dos AP em broadcast já respondidas na questão 6.

16. What are the sender, receiver and BSS ID MAC addresses in these frames? What is the purpose of these two types of frames? (To answer this last question, you'll need to dig into the online references cited earlier in this lab).

R: Probe request: Source: 00:12:f0:1f:57:13, destination: ff:ff:ff:ff:ff:ff, BSSID: ff:ff:ff:ff:ff:ff  
 Probe response: Source: 00:16:b6:f7:1d:51, destination: 00:16:b6:f7:1d:51, BSSID: 00:16:b6:f7:1d:51  
 É uma transmissão para procurar um ponto de acesso do host.

## Projeto 1 – Ethernet and ARP

Este roteiro foi executado em um seguinte cenário: Notebook SAMSUNG utilizando uma rede wireless, onde se conectava ao roteador da TP-LINK que era conectado no modem da NET. Utilizado a URL <http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html> para capturar as requisições no wireshark e realizar as seguintes análises para os questionários abaixo.

1. What is the 48-bit Ethernet address of your computer?

R: O endereço é o 98:83:89:e4:40:94 (SamsungE\_e4:40:94) conforme Imagem 15.

29	0.274427	192.168.1.1	192.168.1.204
783	10.136737	192.168.1.204	128.119.245.12
1139	14.205658	192.168.1.204	184.28.51.192
1009	13.218317	192.168.1.204	184.28.51.192
841	11.138655	192.168.1.204	184.28.51.192

```

Frame 783: 684 bytes on wire (5472 bits), 684 bytes captured (5472 bits) on interface \Device\NPF_{82534E97-41D0-48F6-A8DC-D78DDAFF1087}
Ethernet II, Src: SamsungE_e4:40:94 (98:83:89:e4:40:94), Dst: Tp-LinkT_be:6d:21 (90:9a:4a:be:6d:21)
  > Destination: Tp-LinkT_be:6d:21 (90:9a:4a:be:6d:21)
  > Source: SamsungE_e4:40:94 (98:83:89:e4:40:94)
    Address: SamsungE_e4:40:94 (98:83:89:e4:40:94)
      ....0. .... = LG bit: Globally unique address (factory default)
      ....0. .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
  
```

Imagem 15 – Evidência do endereço na simulação.

2. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu? (Hint: the answer is no). What device has this as its Ethernet address? [Note: this is an important question, and one that students sometimes get wrong. Re-read pages 468-469 in the text and make sure you understand the answer here. ]

R: O endereço é o 90:9a:4a:be:6d:21 (Tp-LinkT\_be:6d:21) conforme Imagem 16.

29	0.274427	192.168.1.1	192.168.1.204	HTTP/X..	559	HTTP/1.1 200 OK
783	10.136737	192.168.1.204	128.119.245.12	HTTP	684	GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1
1139	14.205658	192.168.1.204	184.28.51.192	HTTP	481	GET /filestreamingservice/files/2f079537-2880-4827-ae39-c6
1009	13.218317	192.168.1.204	184.28.51.192	HTTP	480	GET /filestreamingservice/files/2f079537-2880-4827-ae39-c6
841	11.138655	192.168.1.204	184.28.51.192	HTTP	480	GET /filestreamingservice/files/2f079537-2880-4827-ae39-c6

```

Frame 783: 684 bytes on wire (5472 bits), 684 bytes captured (5472 bits) on interface \Device\NPF_{82534E97-41D0-48F6-A8DC-D78DDAFF1087}, id 0
Ethernet II, Src: SamsungE_e4:40:94 (98:83:89:e4:40:94), Dst: Tp-LinkT_be:6d:21 (90:9a:4a:be:6d:21)
  > Destination: Tp-LinkT_be:6d:21 (90:9a:4a:be:6d:21)
  > Source: SamsungE_e4:40:94 (98:83:89:e4:40:94)
    Address: SamsungE_e4:40:94 (98:83:89:e4:40:94)
      ....0. .... = LG bit: Globally unique address (factory default)
      ....0. .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
  
```

Imagem 16 – Evidência do endereço na simulação.

3. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

R: Ipv4 (0x0800) conforme Imagem 17.

29	0.274427	192.168.1.1	192.168.1.204
783	10.136737	192.168.1.204	128.119.245.12
1139	14.205658	192.168.1.204	184.28.51.192
1009	13.218317	192.168.1.204	184.28.51.192
841	11.138655	192.168.1.204	184.28.51.192

```

Frame 783: 684 bytes on wire (5472 bits), 684 bytes captured (5472 bits) on interface \Device\NPF_{
Ethernet II, Src: SamsungE_e4:40:94 (98:83:89:e4:40:94), Dst: Tp-LinkT_be:6d:21 (90:9a:4a:be:6d:21)
  Destination: Tp-LinkT_be:6d:21 (90:9a:4a:be:6d:21)
    Source: SamsungE_e4:40:94 (98:83:89:e4:40:94)
      Address: SamsungE_e4:40:94 (98:83:89:e4:40:94)
        ....0. .... = LG bit: Globally unique address (factory default)
        ....0. .... = IG bit: Individual address (unicast)
      Type: IPv4 (0x0800)
    Internet Protocol Version 4, Src: 192.168.1.204, Dst: 128.119.245.12
    Transmission Control Protocol, Src Port: 49923, Dst Port: 80, Seq: 1, Ack: 1, Len: 630
    Hypertext Transfer Protocol

```

Imagem 17 – Evidência do tipo do protocolo na simulação.

4. How many bytes from the very start of the Ethernet frame does the ASCII “G” in “GET” appear in the Ethernet frame?

R: Sua aparição é em 55 bytes conforme Imagem 18.

```

Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1\r\n
  Host: gaia.cs.umass.edu\r\n
  Connection: keep-alive\r\n
  Cache-Control: max-age=0\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36 Edg/99.0.1150.55\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: pt-BR,pt;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
  If-None-Match: "1194-5db693dfb41bc"\r\n
  If-Modified-Since: Wed, 30 Mar 2022 05:59:01 GMT\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html]
  [HTTP request 1/1]
  [Response in frame: 790]

```

0000	90 9a 4a be 6d 21 98 83 89 e4 40 94 08 00 45 00	..J.m!..@...E
0010	02 9e cb 8b 40 00 00 06 f4 d5 c0 a8 01 cc 80 77	...@...w
0020	f5 0c c3 03 00 50 23 ee c4 c7 0a 81 c9 03 50 18	...P#...P
0030	02 01 18 41 00 00 47 45 54 20 2f 77 69 72 65 73	...A..GE T /wires
0040	68 61 72 6b 2d 6c 61 62 73 2f 48 54 54 50 2d 65	hark-lab s/HTTP-e
0050	74 68 65 72 65 61 6c 2d 6c 61 62 2d 66 69 6c 65	thereal- lab-file
0060	33 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d	3.html H TTP/1.1

Imagem 18 – Evidência da aparição do quadro na simulação.

5. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu (Hint: the answer is no). What device has this as its Ethernet address?

R: É o de meu roteador Tp-LinkT\_be:6d:21 (90:9a:4a:be:6d:21) conforme Imagem 19.

781	10.136430	128.119.245.12	192.168.1.204	TCP	66 80 → 49923 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
782	10.136479	192.168.1.204	128.119.245.12	TCP	54 49923 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
783	10.136737	192.168.1.204	128.119.245.12	HTTP	684 GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1
784	10.137344	128.119.245.12	192.168.1.204	TCP	66 80 → 49924 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
785	10.137392	192.168.1.204	128.119.245.12	TCP	54 49924 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
786	10.145615	128.119.245.12	192.168.1.204	TCP	54 80 → 49526 [ACK] Seq=1 Ack=2 Win=229 Len=0
787	10.182566	20.189.173.6	192.168.1.204	TCP	54 443 → 49517 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
788	10.282593	192.168.1.204	184.28.51.225	TCP	54 [TCP Retransmission] 49580 → 443 [FIN, ACK] Seq=1 Ack=1 Win=512 Len=0
789	10.315362	128.119.245.12	192.168.1.204	TCP	60 80 → 49923 [ACK] Seq=1 Ack=631 Win=30464 Len=0
790	10.315914	128.119.245.12	192.168.1.204	HTTP	295 HTTP/1.1 304 Not Modified

> Frame 781: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF\_{82534E97-41D0-48F6-ABDC-D78DDAFF1087}, id 0

▼ Ethernet II, Src: Tp-LinkT\_be:6d:21 (90:9a:4a:be:6d:21), Dst: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

▼ Destination: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

Address: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

.... 00 .... = LG bit: Globally unique address (factory default)

.... 00 .... = IG bit: Individual address (unicast)

▼ Source: Tp-LinkT\_be:6d:21 (90:9a:4a:be:6d:21)

Address: Tp-LinkT\_be:6d:21 (90:9a:4a:be:6d:21)

.... 00 .... = LG bit: Globally unique address (factory default)

.... 00 .... = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.204

Imagem 19 – Evidência do endereço de origem na simulação.

6. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

R: Pertence ao meu computador SamsungE\_e4:40:94 (98:83:89:e4:40:94) conforme Imagem 20.

781	10.136430	128.119.245.12	192.168.1.204	TCP	66 80 → 49923 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
782	10.136479	192.168.1.204	128.119.245.12	TCP	54 49923 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
783	10.136737	192.168.1.204	128.119.245.12	HTTP	684 GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1
784	10.137344	128.119.245.12	192.168.1.204	TCP	66 80 → 49924 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
785	10.137392	192.168.1.204	128.119.245.12	TCP	54 49924 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
786	10.145615	128.119.245.12	192.168.1.204	TCP	54 80 → 49526 [ACK] Seq=1 Ack=2 Win=229 Len=0
787	10.182566	20.189.173.6	192.168.1.204	TCP	54 443 → 49517 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
788	10.282593	192.168.1.204	184.28.51.225	TCP	54 [TCP Retransmission] 49580 → 443 [FIN, ACK] Seq=1 Ack=1 Win=512 Len=0
789	10.315362	128.119.245.12	192.168.1.204	TCP	60 80 → 49923 [ACK] Seq=1 Ack=631 Win=30464 Len=0
790	10.315914	128.119.245.12	192.168.1.204	HTTP	295 HTTP/1.1 304 Not Modified

> Frame 781: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF\_{82534E97-41D0-48F6-ABDC-D78DDAFF1087}, id 0

▼ Ethernet II, Src: Tp-LinkT\_be:6d:21 (90:9a:4a:be:6d:21), Dst: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

▼ Destination: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

Address: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

.... 00 .... = LG bit: Globally unique address (factory default)

.... 00 .... = IG bit: Individual address (unicast)

▼ Source: Tp-LinkT\_be:6d:21 (90:9a:4a:be:6d:21)

Address: Tp-LinkT\_be:6d:21 (90:9a:4a:be:6d:21)

.... 00 .... = LG bit: Globally unique address (factory default)

.... 00 .... = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.204

Imagem 20 – Evidência do endereço de destino na simulação.

7. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

R: 0x0800 (IPv4)

8. How many bytes from the very start of the Ethernet frame does the ASCII “O” in “OK” (i.e., the HTTP response code) appear in the Ethernet frame?

R: Sua aparição é em 14 bytes conforme Imagem 21.

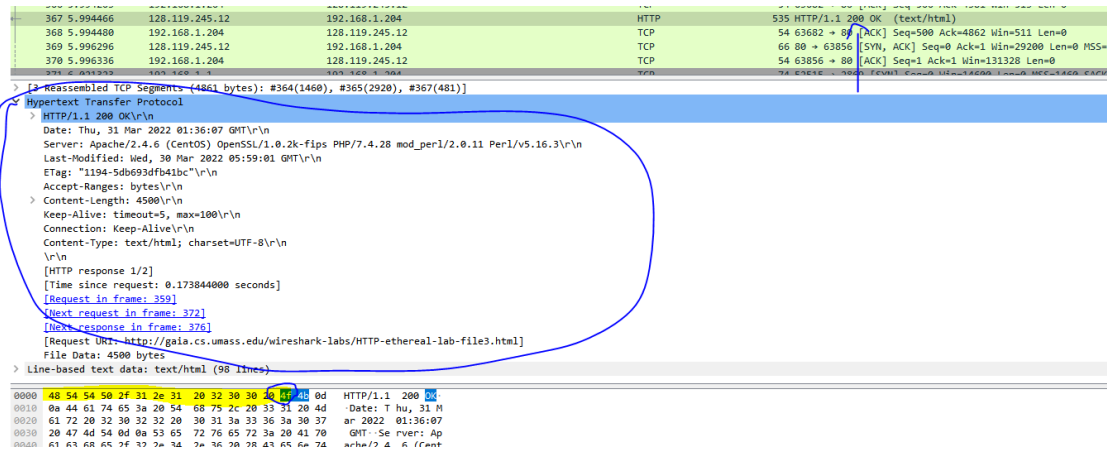


Imagem 21 – Evidência de quantos bytes o quadro inicia na simulação.

9. Write down the contents of your computer's ARP cache. What is the meaning of each column value?

R: Representam o endereço IP na camada de rede, o endereço MAC para se comunicar fisicamente com o hardware que está localizado nesse endereço IP e se ele está mudando ou não (dinâmico) ou estático conforme Imagem 22.

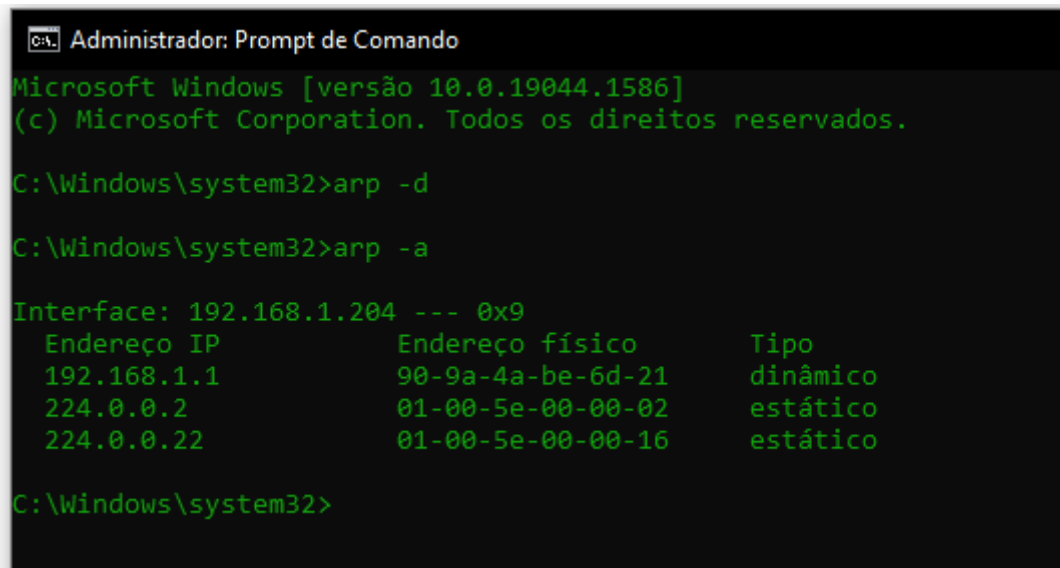


Imagem 22 – Evidência da consulta de cache ARP na simulação.

10. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?

R: O endereço origem é o 98:83:89:e4:40:94 e o destino é o ff:ff:ff:ff:ff:ff conforme Imagem 23.

Qualcomm Atheros QCA9300 Wireless Network Adapter: Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
210	3.076603	SamsungE_e4:40:94	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.204
211	3.078378	Tp-LinkT_be:6d:21	SamsungE_e4:40:94	ARP	42	192.168.1.1 is at 90:9a:4a:be:6d:21
4	0.002731	192.168.1.204	192.168.1.1	HTTP	254	GET /qyyjn/WANCfg.xml HTTP/1.1
16	0.011667	192.168.1.204	192.168.1.1	HTTP	329	SUBSCRIBE /qyyjn/evt/CmnIfCfg HTTP/1.1
18	0.013908	192.168.1.1	192.168.1.204	HTTP	276	HTTP/1.1 200 OK
95	0.056769	192.168.1.204	34.104.35.123	HTTP	425	GET /edgedl/release2/chrome_component/ad3na
98	0.104336	34.104.35.123	192.168.1.204	HTTP	348	HTTP/1.1 206 Partial Content

> Frame 210: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF\_{82534E97-41D0-48F6-ABDC-D78DDAFF1087}, id 0

▼ Ethernet II, Src: SamsungE\_e4:40:94 (98:83:89:e4:40:94), Dst: Broadcast (ff:ff:ff:ff:ff:ff), id 0

▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)

Address: Broadcast (ff:ff:ff:ff:ff:ff)

.... 1 ..... = LG bit: Locally administered address (this is NOT the factory default)

.... 1 ..... = IG bit: Group address (multicast/broadcast)

▼ Source: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

Address: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

.... 0 ..... = LG bit: Globally unique address (factory default)

.... 0 ..... = IG bit: Individual address (unicast)

Type: ARP (0x0806)

> Address Resolution Protocol (request)

Imagem 23 – Evidência do endereço de origem e destino na simulação.

11. Give the hexadecimal value for the two-byte Ethernet Frame type field.  
What upper layer protocol does this correspond to?

R: ARP (0x0806) conforme Imagem 24.

Qualcomm Atheros QCA9300 Wireless Network Adapter: Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
210	3.076603	SamsungE_e4:40:94	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.204
211	3.078378	Tp-LinkT_be:6d:21	SamsungE_e4:40:94	ARP	42	192.168.1.1 is at 90:9a:4a:be:6d:21
4	0.002731	192.168.1.204	192.168.1.1	HTTP	254	GET /qyyjn/WANCfg.xml HTTP/1.1
16	0.011667	192.168.1.204	192.168.1.1	HTTP	329	SUBSCRIBE /qyyjn/evt/CmnIfCfg HTTP/1.1
18	0.013908	192.168.1.1	192.168.1.204	HTTP	276	HTTP/1.1 200 OK
95	0.056769	192.168.1.204	34.104.35.123	HTTP	425	GET /edgedl/release2/chrome_component/ad3na
98	0.104336	34.104.35.123	192.168.1.204	HTTP	348	HTTP/1.1 206 Partial Content

> Frame 210: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF\_{82534E97-41D0-48F6-ABDC-D78DDAFF1087}, id 0

▼ Ethernet II, Src: SamsungE\_e4:40:94 (98:83:89:e4:40:94), Dst: Broadcast (ff:ff:ff:ff:ff:ff), id 0

▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)

Address: Broadcast (ff:ff:ff:ff:ff:ff)

.... 1 ..... = LG bit: Locally administered address (this is NOT the factory default)

.... 1 ..... = IG bit: Group address (multicast/broadcast)

▼ Source: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

Address: SamsungE\_e4:40:94 (98:83:89:e4:40:94)

.... 0 ..... = LG bit: Globally unique address (factory default)

.... 0 ..... = IG bit: Individual address (unicast)

Type: ARP (0x0806)

> Address Resolution Protocol (request)

Imagem 24 – Evidência do campo tipo do quadro de Ethernet na simulação.

12. Download the ARP specification from <ftp://ftp.rfc-editor.org/in-notes/std/std37.txt>. A readable, detailed discussion of ARP is also at <http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html>.

- a) How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

R: 21 bytes

- b) What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP request is made?

R: 001

- c) Does the ARP message contain the IP address of the sender?

R: Sim

13. Now find the ARP reply that was sent in response to the ARP request.

- a) How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

R: 21 bytes

- b) What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP response is made?

R: 002

- c) Where in the ARP message does the “answer” to the earlier ARP request appear – the IP address of the machine having the ethernet address whose corresponding IP address is being queried?

R: No endereço MAC do remetente

14. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?

R: Origem: 98:83:89:e4:40:94. Destino: ff:ff:ff:ff:ff:ff

## **Projeto 2 - Simulação Packet Tracer**



## Primeira topologia – Com fio.

Foi feita uma topologia utilizando um roteador 2811, um switch 2960, um laptop, quatro PCs e um server, conforme Imagem 25

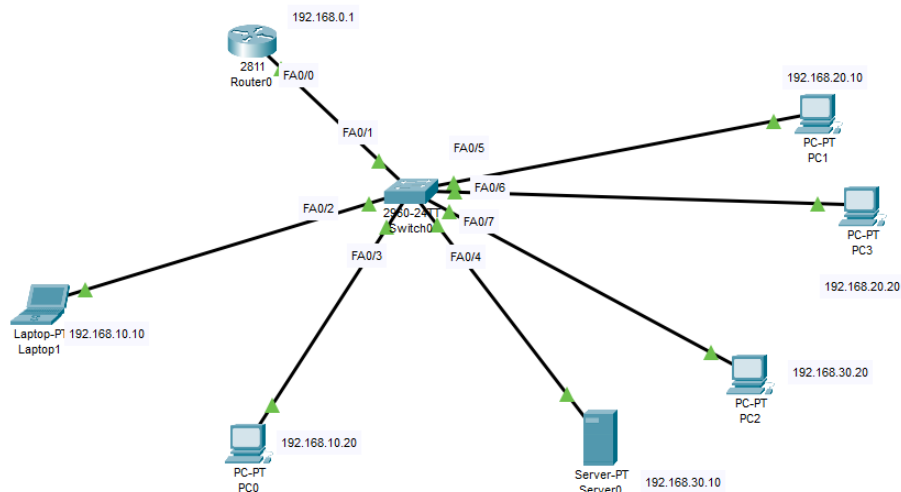


Imagem 25 – Topologia com fio.

Dentro da topologia foi configurado 3 VLANs, chamadas de Lab1, Lab2 e Lab3. No Lab1 foram mapeadas as portas FA0/2 e FA0/3, no LAB2 foram mapeadas as portas FA0/4 e FA0/7 e por fim no LAB3 foram mapeadas as portas FA0/5 e FA0/6, conforme Imagem 26. Desta forma os computadores de cada VLAN não podem acessar os computadores de outra VLAN, conforme Imagem 27.

```
LABS#show vlan
```

VLAN	Name	Status	Ports
1	default	active	Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23 Fa0/24, Gig0/1, Gig0/2
10	LAB1	active	Fa0/2, Fa0/3
20	LAB2	active	Fa0/4, Fa0/7
30	LAB3	active	Fa0/5, Fa0/6
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Imagem 26 – Evidência de configuração VLAN.

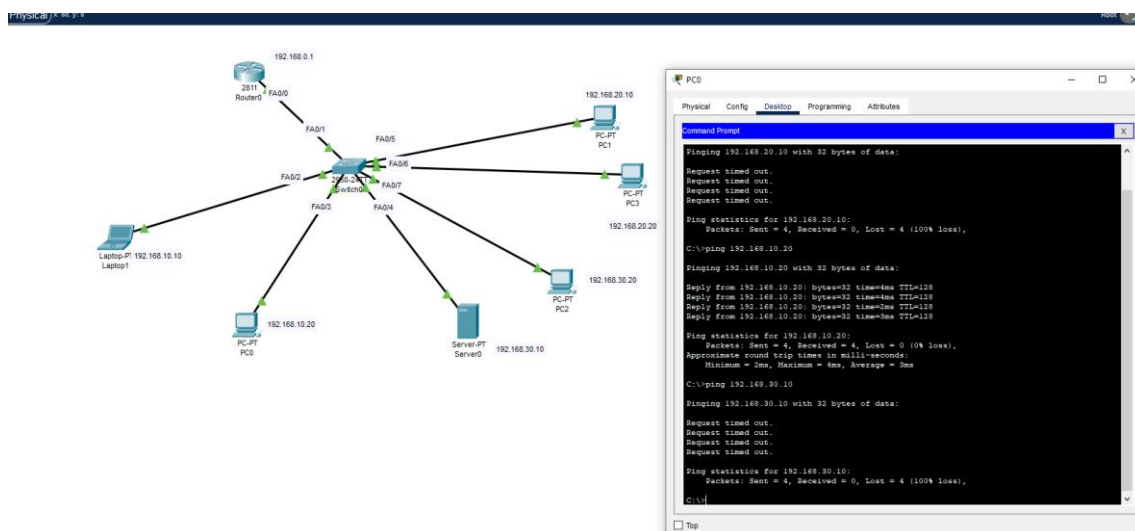


Imagem 27 – Simulação de ping entre VLANs.

Como pode ser visto na Imagem 27, ao tentar fazer o ping para um computador de mesma VLAN é possível comunicar com o host, já fazendo o ping para um ip de outra VLAN não é fechada a comunicação.

No modo simulação é possível ver que o switch ao receber a solicitação de ping, envia para os hosts dividido por VLANs, ou seja, o broadcast de envio não é feito para todos os hosts conectados, mas para todos os hosts de cada VLAN. A imagem 28 mostra um pouco disso.

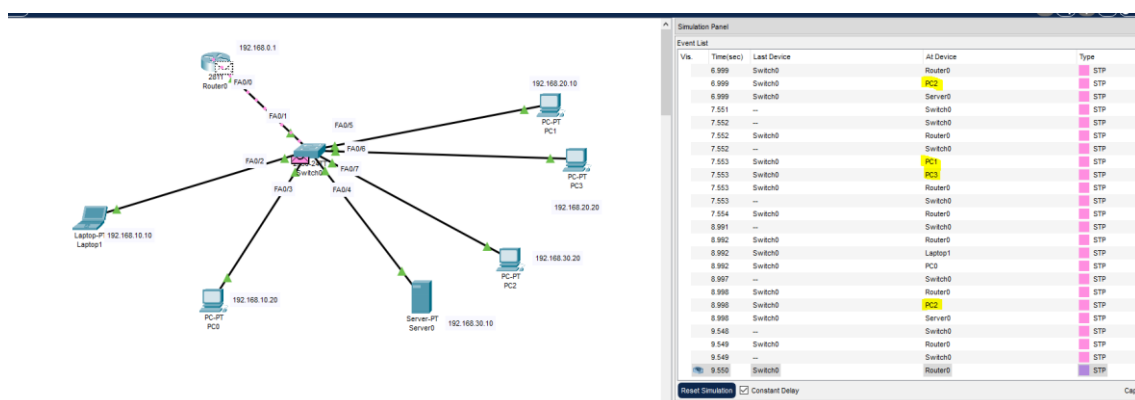


Imagem 28 – Simulação onde é enviado apenas para os hosts da VLAN.

No que se refere a tabela ARP, ao consultar a tabela de endereços MAC, só foi possível encontrar o MAC relacionado ao roteador, visto que não foi mapeado os MACs no modo estático, mesmo seguindo à risca a configuração da documentação. A Imagem 29 e Imagem 30 exemplificam isso.

```

LABS#show mac-address-table
      Mac Address Table
-----
Vlan    Mac Address      Type    Ports
-----
1       0001.96b9.6501   DYNAMIC Fa0/1
LABS#
  
```

Imagem 29 – Tabela de endereços MAC dinâmicos.

```
LABS#show mac-address-table static
      Mac Address Table
-----
Vlan    Mac Address      Type      Ports
----    -

```

Imagem 30 – Tabela de endereços MAC estáticos.

No que se refere aos protocolos usados, a Imagem 31 mostra os protocolos usados ao fazer um ping entre hosts de mesma VLAN, onde primeiro é executado o protocolo ICMP para mandar o ping de dados, após é executado o protocolo ARP para consultar o MAC de destino e assim ter o endereço de destino. Após a obtenção do endereço MAC de destino é enviado o pacote ICMP que é retornado com sucesso.

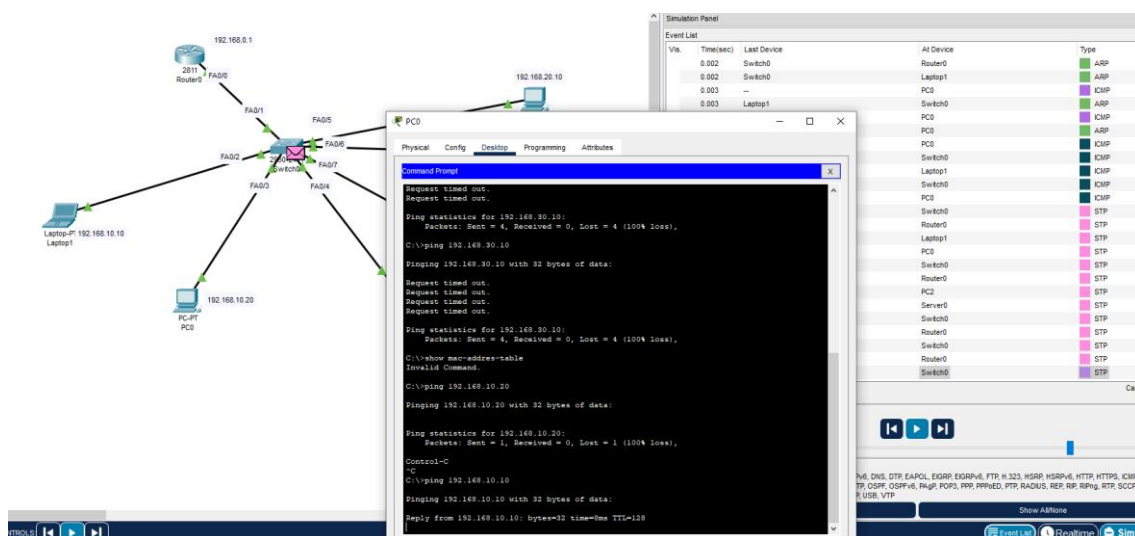


Imagem 31 – Simulação de ping mostrando protocolos usados.

## Primeira topologia – Sem fio.

Nesta topologia foi simulado uma rede sem fio utilizando um roteador ICR4331, um switch 2960, um access point, um home router, um wireless end device, dois smartphones e um tablet, conforme Imagem 32.

Foi feita a configuração das VLANs semelhante a topologia anterior, porém nesta foi nomeado apenas duas VLANs, uma chamada LAB1 e outra chamada LAB2, conectas nas portas FA0/1 e FA0/3, conforme mostra a Imagem 33.

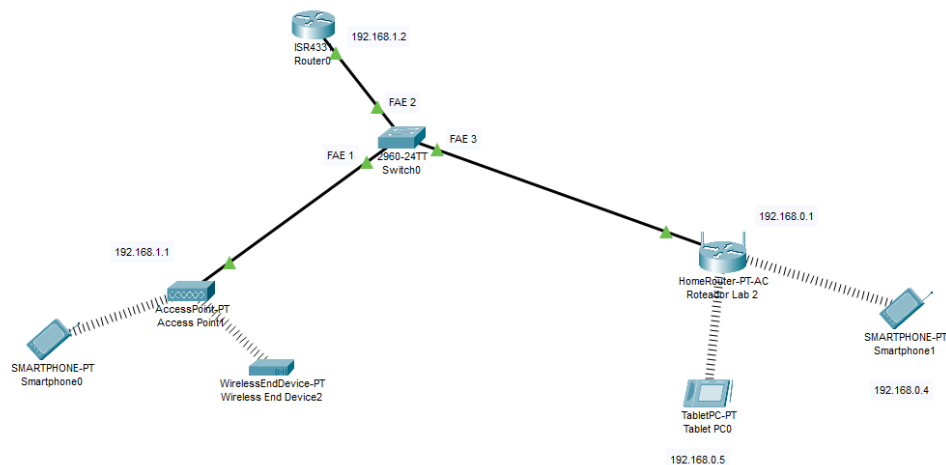


Imagem 32 – topologia sem fio.

```

H01S/enable
H01S#show vlan

```

VLAN	Name	Status	Ports
1	default	active	Fa0/2, Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22 Fa0/23, Fa0/24, Gig0/1, Gig0/2
10	LAB1	active	Fa0/1
20	LAB2	active	Fa0/3
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Imagem 33 – Configuração VLAN.

Assim como na primeira topologia, ao tentar executar o ping para um host de outra VLAN não é possível, entretanto ao executar o ping para a mesma VLAN a consulta é satisfeita, conforme Imagem 34 que mostra um teste para um host da VLAN do home router e outra para um host da VLAN do access point.

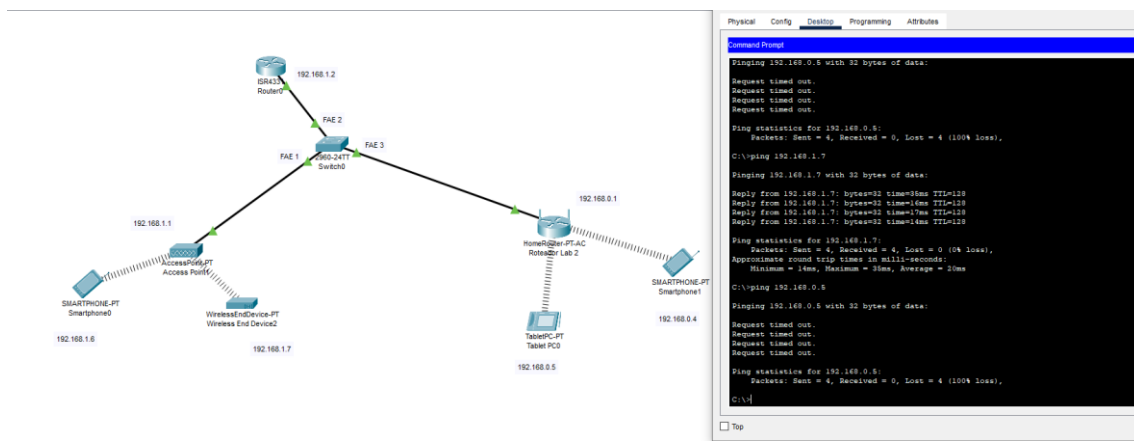


Imagem 34 – simulação ping entre VLANs e na mesma VLAN.

Em relação a tabela de MACs, nesta topologia foi possível verificar que foi preenchida a tabela com os MACs relacionados as duas VLANs diferentemente da topologia com fio, conforme mostra a Imagem 35.

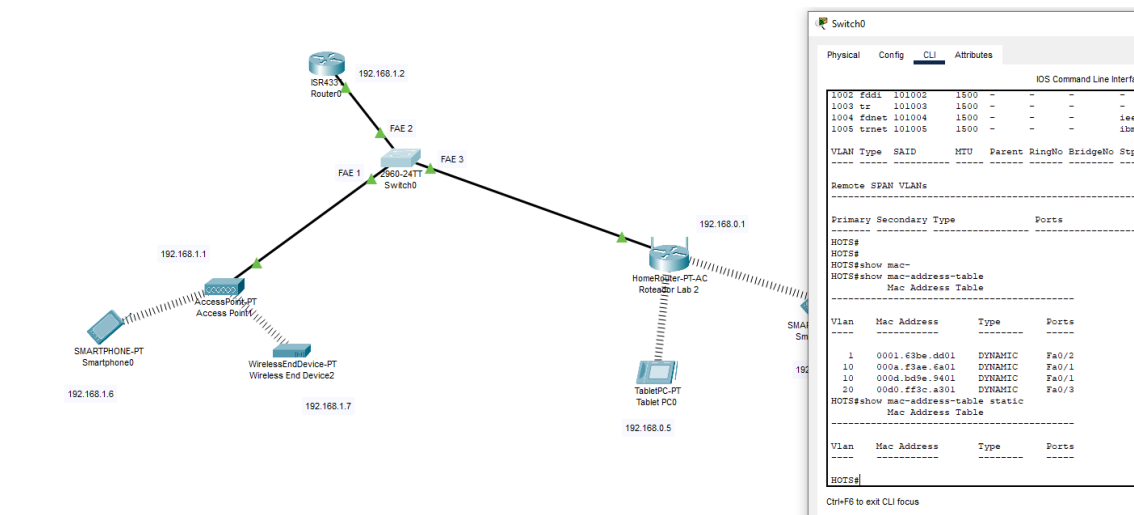


Imagem 35 – Tabela de MACs.

Em relação aos protocolos, como a tabela de ARP já estava preenchida, nesta simulação de ping, não aparece nenhum protocolo ARP, apenas a comunicação via ICMP, conforme Imagem 36.

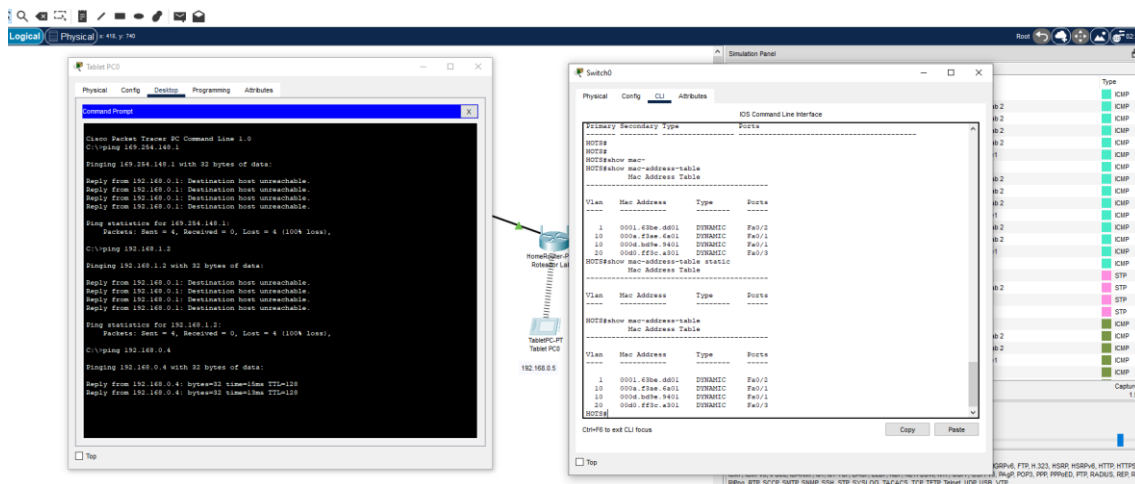


Imagem 36 – Simulação de ping topologia sem fio – protocolos.

### Projeto 3 - Desenvolvimento de EDC

O checksum é uma técnica baseada em soma binária para detecção de erros de transmissão de pacotes. Seu funcionamento basicamente é somar todas as mensagens do pacote e após a soma inverter os bits para gerar o checksum do pacote. Feito isso o cliente ao receber o pacote irá realizar a soma das mensagens e invés de inverter os bits da soma, irá efetuar uma nova soma com o checksum recebido junto com o pacote, o resultado tem que resultar tudo 1, ou seja, em um caso de exemplo de uma mensagem de 255 bits, o checksum tem que ser igual a 255.

O desenvolvimento deste EDC foi feito em Python, separando o processo em três funções: `binarySum()`, `calculate_checkSum()` e `corrupt()`. A primeira função `binarySum(a,b)` é responsável pela soma binária, utilizando operadores de bits para realizar isso. A segunda função `calculate_checkSum(cls,data)` é responsável por calcular o checksum da mensagem, chamando a função de soma sempre que necessário. A última função, `corrupt(msg, computed_checksum_S)`, é responsável por verificar se houve algum problema no recebimento do pacote, esta função recebe a mensagem e o checksum associado a ela, faz o cálculo do checksum da mensagem recebida e depois compara ela com o checksum computado, caso o valor seja diferente de 255 ela retorna que o pacote estava corrompido e caso contrário, retorna que o pacote chegou em perfeito estado. Foi feita também uma função que simula o recebimento do pacote, apenas para ilustrar que são processos diferentes. A imagem 25 mostra as funções explanadas.

```

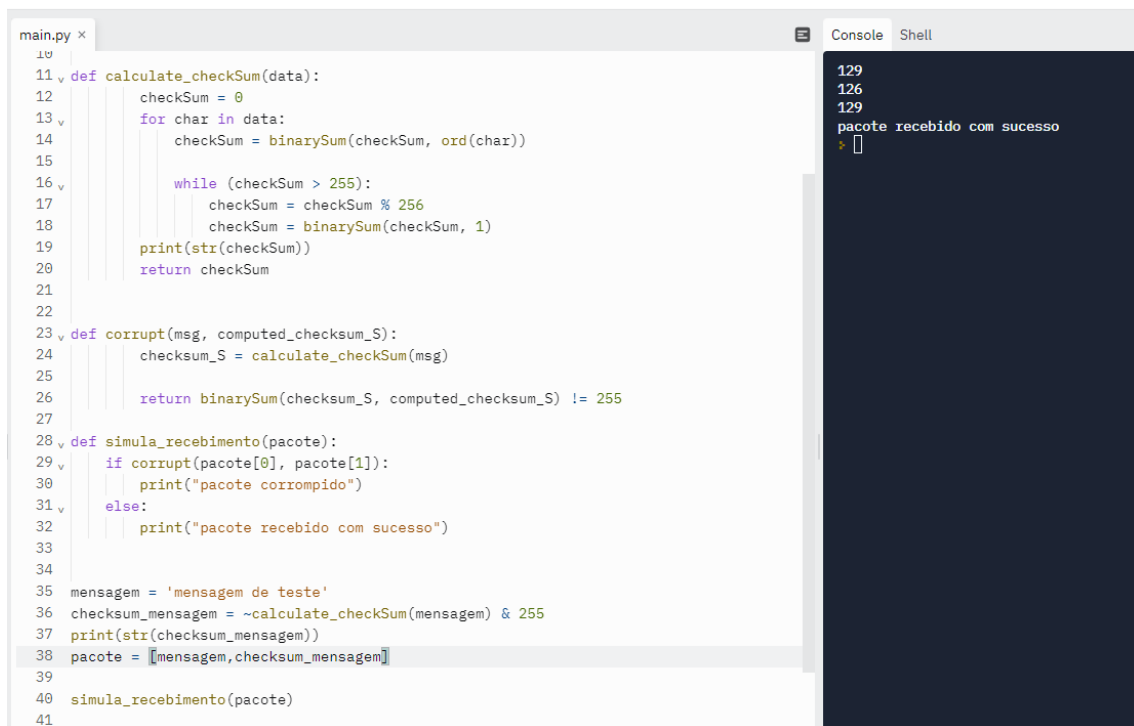
1 def binarySum(a, b):
2     while (b != 0):
3         carry = a & b
4         a = a ^ b
5         b = carry << 1
6     return a
7
8
9 def calculate_checkSum(data):
10    checksum = 0
11    for char in data:
12        checksum = binarySum(checksum, ord(char))
13
14    while (checksum > 255):
15        checksum = checksum % 256
16        checksum = binarySum(checksum, 1)
17    print(str(checksum))
18    return checksum
19
20
21 def corrupt(msg, computed_checksum_S):
22    checksum_S = calculate_checkSum(msg)
23
24    return binarySum(checksum_S, computed_checksum_S) != 255
25
26
27 def simula_recebimento(pacote):
28    if corrupt(pacote[0], pacote[1]):
29        print("pacote corrompido")
30    else:
31        print("pacote recebido com sucesso")
32
33
34

```

Imagem 25 – Código desenvolvido para cálculo do checksum.

Foram feitas duas simulações com o código, uma com a mensagem correta onde o checksum seria validado com sucesso e outra em que a mensagem foi alterada e assim o checksum recebido pelo destinatário seria diferente do checksum do pacote. A imagem 26 evidencia o cenário do pacote recebido sem erros. Na linha 35 da Imagem 26 é possível ver a imagem utilizada, o processo de cálculo do checksum está na linha 36 e na linha 38 ele é colocado junto a mensagem e enviado no pacote. Na função de simula recebimento, o algoritmo envia a mensagem e o checksum recebido para validar se o pacote foi recebido com sucesso.





The image shows a code editor window with a file named 'main.py'. The code defines several functions: `calculate_checkSum`, `corrupt`, and `simula_recebimento`. It then creates a message, calculates its checksum, and simulates its reception. The console output on the right shows the execution results.

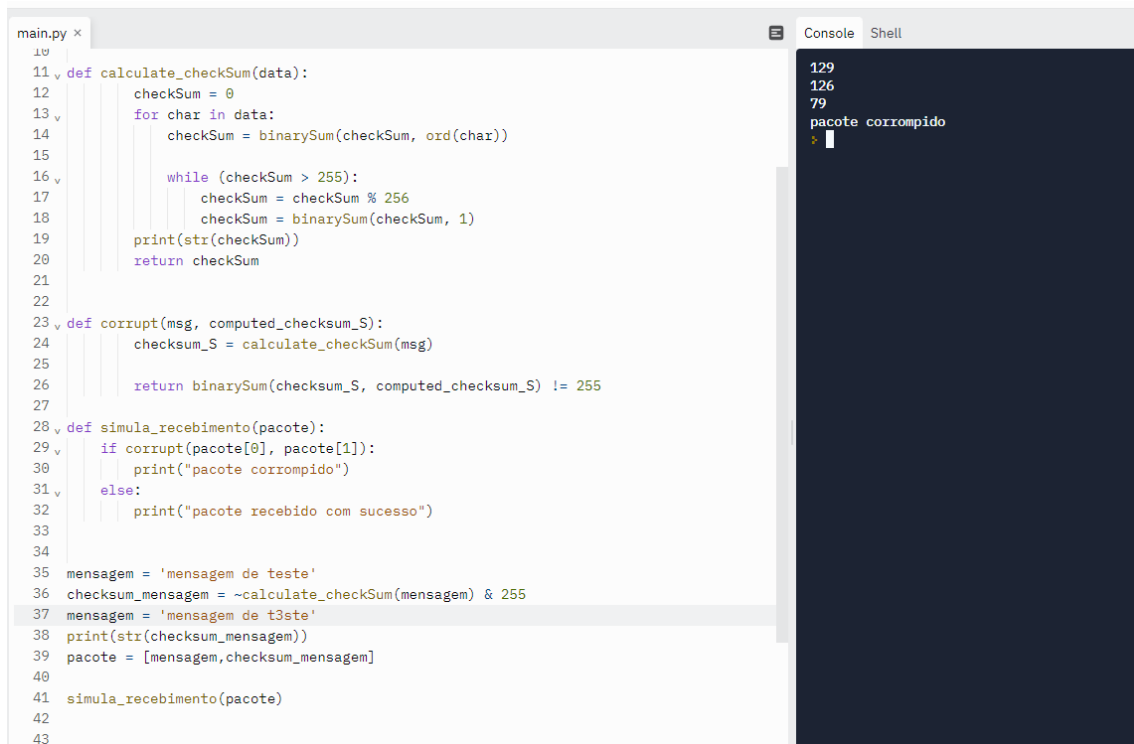
```
main.py x
10
11 def calculate_checkSum(data):
12     checksum = 0
13     for char in data:
14         checksum = binarySum(checksum, ord(char))
15
16     while (checksum > 255):
17         checksum = checksum % 256
18         checksum = binarySum(checksum, 1)
19     print(str(checksum))
20     return checksum
21
22
23 def corrupt(msg, computed_checksum_S):
24     checksum_S = calculate_checkSum(msg)
25
26     return binarySum(checksum_S, computed_checksum_S) != 255
27
28 def simula_recebimento(pacote):
29     if corrupt(pacote[0], pacote[1]):
30         print("pacote corrompido")
31     else:
32         print("pacote recebido com sucesso")
33
34
35 mensagem = 'mensagem de teste'
36 checksum_mensagem = ~calculate_checkSum(mensagem) & 255
37 print(str(checksum_mensagem))
38 pacote = [mensagem, checksum_mensagem]
39
40 simula_recebimento(pacote)
41
```

Console Shell

```
129
126
129
pacote recebido com sucesso
> []
```

Imagem 26 – simulação em que a mensagem chega corretamente.

Na segunda simulação, foi alterada a mensagem logo após o cálculo do checksum de envio, simulando algum corrompimento. Após isso é feito todo processo explicado anteriormente e é retornado que o pacote foi corrompido. A imagem 27 evidencia isso, destacando na linha 37 que é feita a mudança na mensagem para simular o corrompimento.



```
main.py x
10
11 def calculate_checksum(data):
12     checksum = 0
13     for char in data:
14         checksum = binarySum(checksum, ord(char))
15
16     while (checksum > 255):
17         checksum = checksum % 256
18         checksum = binarySum(checksum, 1)
19     print(str(checksum))
20     return checksum
21
22
23 def corrupt(msg, computed_checksum_S):
24     checksum_S = calculate_checksum(msg)
25
26     return binarySum(checksum_S, computed_checksum_S) != 255
27
28 def simula_recebimento(pacote):
29     if corrupt(pacote[0], pacote[1]):
30         print("pacote corrompido")
31     else:
32         print("pacote recebido com sucesso")
33
34
35 mensagem = 'mensagem de teste'
36 checksum_mensagem = ~calculate_checksum(mensagem) & 255
37 mensagem = 'mensagem de t3ste'
38 print(str(checksum_mensagem))
39 pacote = [mensagem, checksum_mensagem]
40
41 simula_recebimento(pacote)
42
43
```

Console

```
129
126
79
pacote corrompido
>
```

Imagem 27 – Simulação em que a mensagem foi corrompida.