Redes de Computadores

Wireshark

Prof. Luís Eduardo Tenório Silva
luis.silva@garanhuns.ifpe.edu.br

INSTITUTO FEDERAL
Pernambuco
Campus Garanhuns

Análise de tráfego

- *Trobleshooting* de rede;
- Investigação de incidentes de segurança;
- Estudo dos protocolos de rede;
- Análise de desempenho da rede.

Análise de tráfego

- Ferramentas de análise de rede
  - tcpdump/wireshark
  - traceroute/tracert
  - ping
  - nmap
  - nc
  - iptables
  - nperf
  - ...

1. Wireshark

- Ferramenta de **análise de pacotes** (packet sniffer)
  - » Realiza passivamente uma cópia dos pacotes que trafegam em uma ou mais interfaces de rede;
  - » Identifica todos os cabeçalhos (*headers*) e cargas (*payload*) dos protocolos utilizados.

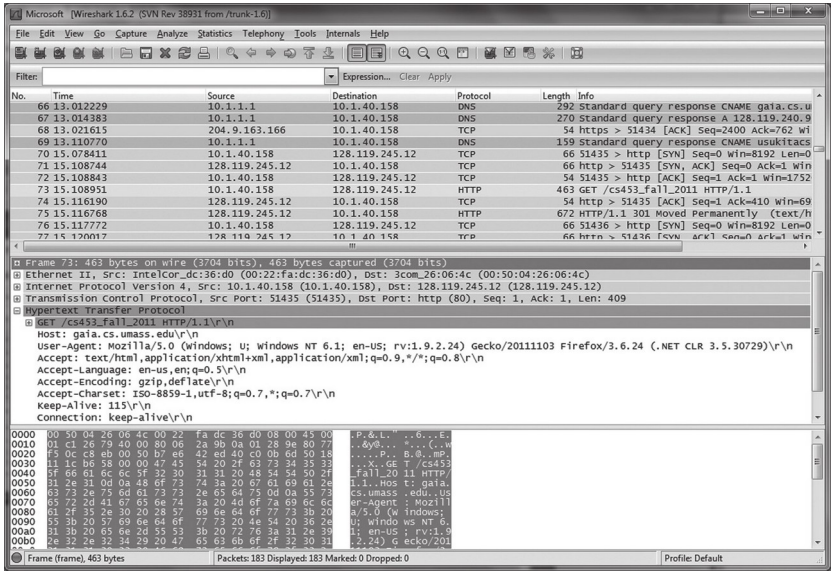


Figura: Captura de pacotes usando o Wireshark

1. Wireshark

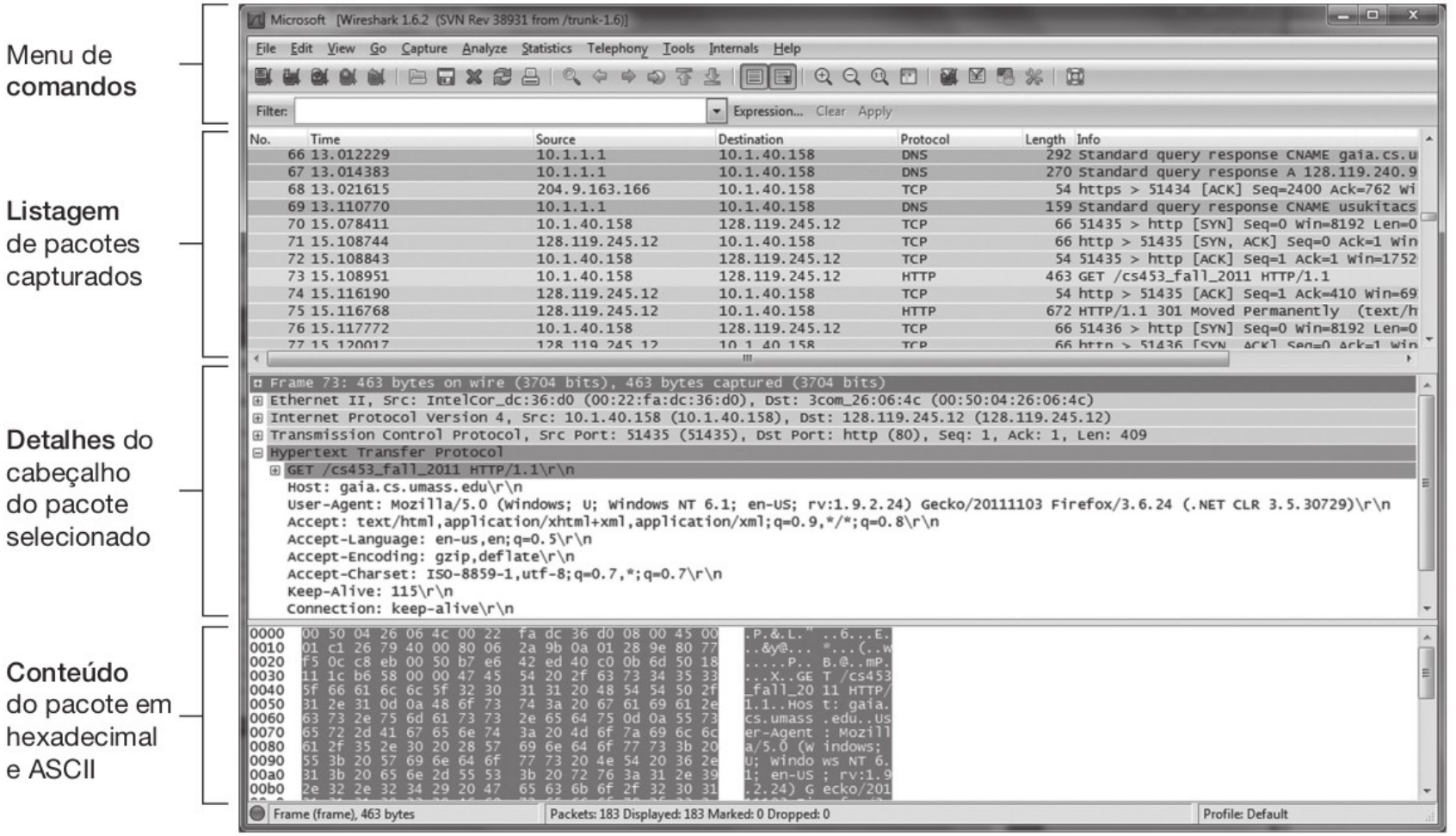
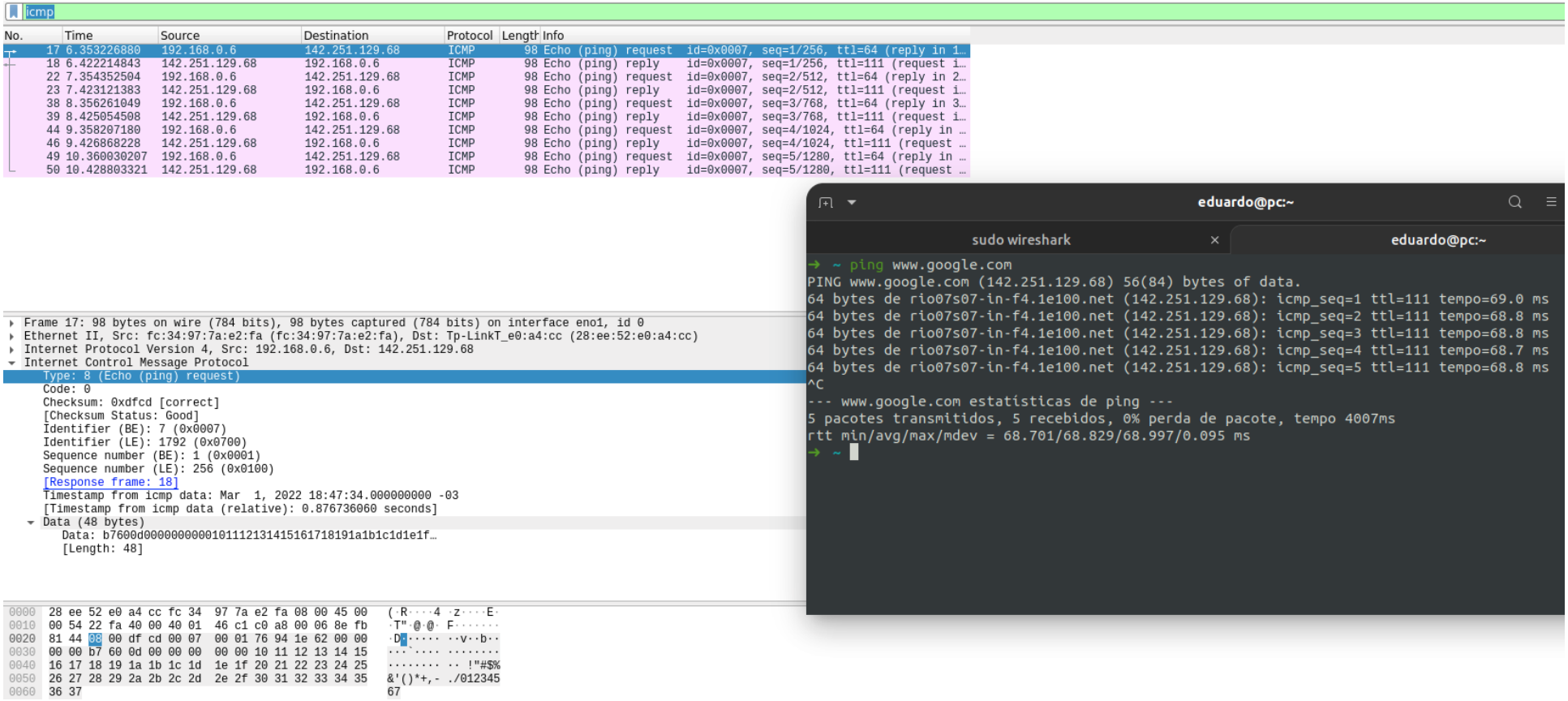


Figura: Estrutura do Wireshark

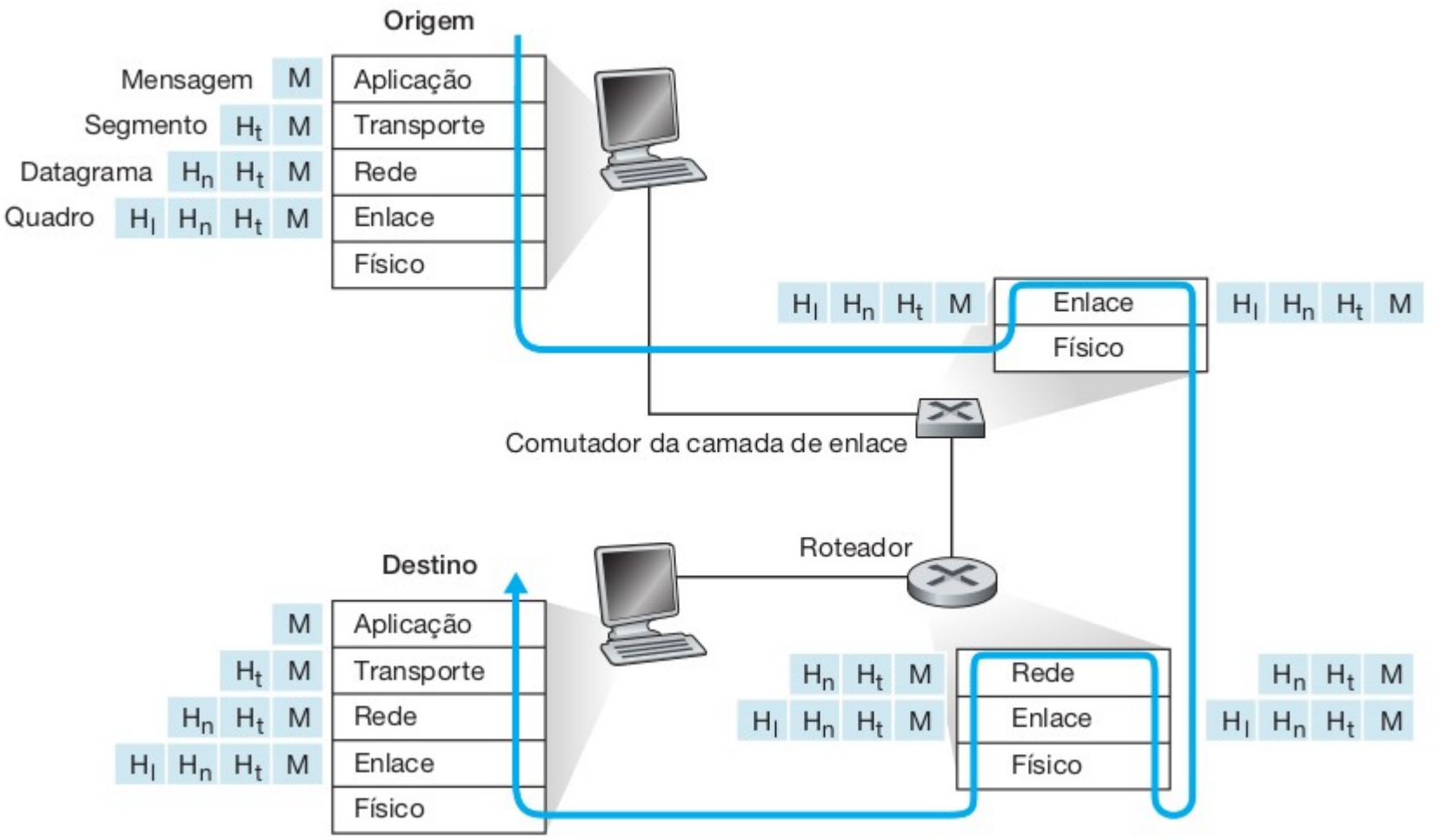
2. Análise

- Testar conectividade entre duas máquinas
  - Comando ping
    - Envia uma mensagem (**eco request**) para um dispositivo e recebe um resposta (**eco response**) caso o dispositivo receba a mensagem;
    - Utiliza o protocolo **ICMP** (Internet Control Message Protocol);
    - Trabalha na camada de rede (3) do modelo híbrido.
- Ex:

```
➜  ~ ping www.google.com
PING www.google.com (142.250.78.228) 56(84) bytes of data.
64 bytes de rio07s02-in-f4.1e100.net (142.250.78.228): icmp_seq=1 ttl=111 tempo=68.6 ms
64 bytes de rio07s02-in-f4.1e100.net (142.250.78.228): icmp_seq=2 ttl=111 tempo=68.8 ms
64 bytes de rio07s02-in-f4.1e100.net (142.250.78.228): icmp_seq=3 ttl=111 tempo=68.7 ms
64 bytes de rio07s02-in-f4.1e100.net (142.250.78.228): icmp_seq=4 ttl=111 tempo=68.7 ms
^C
--- www.google.com estatísticas de ping ---
4 pacotes transmitidos, 4 recebidos, 0% perda de pacote, tempo 3002ms
rtt min/avg/max/mdev = 68.577/68.697/68.784/0.076 ms
```


icmp
No. Time Source Destination Protocol Length Info
17 6.353226880 192.168.0.6 142.251.129.68 ICMP 98 Echo (ping) request id=0x0007, seq=1/256, ttl=64 (reply in 1…
18 6.422214843 142.251.129.68 192.168.0.6 ICMP 98 Echo (ping) reply id=0x0007, seq=1/256, ttl=111 (request i…
22 7.354352504 192.168.0.6 142.251.129.68 ICMP 98 Echo (ping) request id=0x0007, seq=2/512, ttl=64 (reply in 2…
23 7.423121383 142.251.129.68 192.168.0.6 ICMP 98 Echo (ping) reply id=0x0007, seq=2/512, ttl=111 (request i…
38 8.356261049 192.168.0.6 142.251.129.68 ICMP 98 Echo (ping) request id=0x0007, seq=3/768, ttl=64 (reply in 3…
39 8.425054508 142.251.129.68 192.168.0.6 ICMP 98 Echo (ping) reply id=0x0007, seq=3/768, ttl=111 (request i…
44 9.358207180 192.168.0.6 142.251.129.68 ICMP 98 Echo (ping) request id=0x0007, seq=4/1024, ttl=64 (reply in …
46 9.426868228 142.251.129.68 192.168.0.6 ICMP 98 Echo (ping) reply id=0x0007, seq=4/1024, ttl=111 (request …
49 10.360030207 192.168.0.6 142.251.129.68 ICMP 98 Echo (ping) request id=0x0007, seq=5/1280, ttl=64 (reply in …
50 10.428803321 142.251.129.68 192.168.0.6 ICMP 98 Echo (ping) reply id=0x0007, seq=5/1280, ttl=111 (request …
Frame 17: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eno1, id 0
Ethernet II, Src: fc:34:97:7a:e2:fa (fc:34:97:7a:e2:fa), Dst: Tp-LinkT_e0:a4:cc (28:ee:52:e0:a4:cc)
Internet Protocol Version 4, Src: 192.168.0.6, Dst: 142.251.129.68
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xdfcd [correct]
[Checksum Status: Good]
Identifier (BE): 7 (0x0007)
Identifier (LE): 1792 (0x0700)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Response frame: 18]
Timestamp from icmp data: Mar 1, 2022 18:47:34.000000000 -03
[Timestamp from icmp data (relative): 0.876736060 seconds]
Data (48 bytes)
Data: b7600d000000000101112131415161718191a1b1c1d1e1f…
[Length: 48]
0000 28 ee 52 e0 a4 cc fc 34 97 7a e2 fa 08 00 45 00 (·R····4 ·z····E·
0010 00 54 22 fa 40 00 40 01 46 c1 c0 a8 00 06 8e fb ·T"·@·@· F·····
0020 81 44 08 00 df cd 00 07 00 01 76 94 1e 62 00 00 ·D····· ··v··b··
0030 00 00 b7 60 0d 00 00 00 00 00 10 11 12 13 14 15 ···`···· ····
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ····· ·· !"#$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,- ./012345
0060 36 37 67
eduardo@pc:~
sudo wireshark
eduardo@pc:~
➜ ~ ping www.google.com
PING www.google.com (142.251.129.68) 56(84) bytes of data.
64 bytes de rio07s07-in-f4.1e100.net (142.251.129.68): icmp_seq=1 ttl=111 tempo=69.0 ms
64 bytes de rio07s07-in-f4.1e100.net (142.251.129.68): icmp_seq=2 ttl=111 tempo=68.8 ms
64 bytes de rio07s07-in-f4.1e100.net (142.251.129.68): icmp_seq=3 ttl=111 tempo=68.8 ms
64 bytes de rio07s07-in-f4.1e100.net (142.251.129.68): icmp_seq=4 ttl=111 tempo=68.7 ms
64 bytes de rio07s07-in-f4.1e100.net (142.251.129.68): icmp_seq=5 ttl=111 tempo=68.8 ms
^C
--- www.google.com estatísticas de ping ---
5 pacotes transmitidos, 5 recebidos, 0% perda de pacote, tempo 4007ms
rtt min/avg/max/mdev = 68.701/68.829/68.997/0.095 ms
➜ ~

2. Análise


Origem
Mensagem M
Segmento Ht M
Datagrama Hn Ht M
Quadro Hl Hn Ht M
Aplicação
Transporte
Rede
Enlace
Físico
Hl Hn Ht M
Enlace
Físico
Hl Hn Ht M
Comutador da camada de enlace
Roteador
Destino
M
Ht M
Hn Ht M
Hl Hn Ht M
Aplicação
Transporte
Rede
Enlace
Físico
Hn Ht M
Hl Hn Ht M
Rede
Enlace
Físico
Hn Ht M
Hl Hn Ht M

2. Análise

- **Camada física**: sinal eletromagnético (interface eno1)
- **Camada de enlace**: Protocolo ethernet
  - Endereço físico de origem (MAC): fc:34:97:7a:e2:fa
  - Endereço físico de destino (MAC): 28:ee:52:e0:a4:cc
- **Camada de rede**: Protocolo ICMP
  - Endereço lógico de origem (IP): 192.168.0.6
  - Endereço lógico de destino (IP): 142.251.129.68

2. Análise

- **Camada física**: sinal eletromagnético (interface eno1)
- **Camada de enlace**: Protocolo ethernet
  - Endereço físico de origem (MAC): fc:34:97:7a:e2:fa
  - Endereço físico de destino (MAC): 28:ee:52:e0:a4:cc
- **Camada de rede**: Protocolo ICMP
  - Endereço lógico de origem (IP): 192.168.0.6
  - Endereço lógico de destino (IP): 142.251.129.68

Como obter o endereço físico de destino?

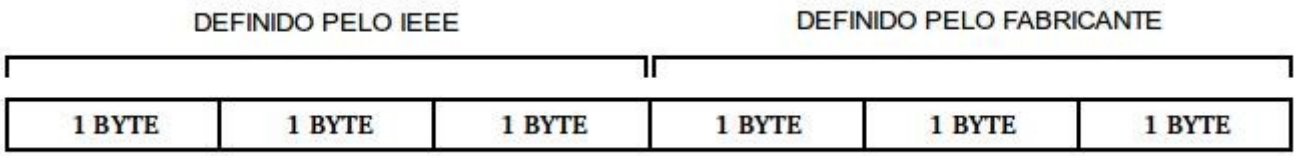
- Protocolo **ARP**

Como obter o endereço lógico de destino usando o nome?

- Protocolo **DNS**

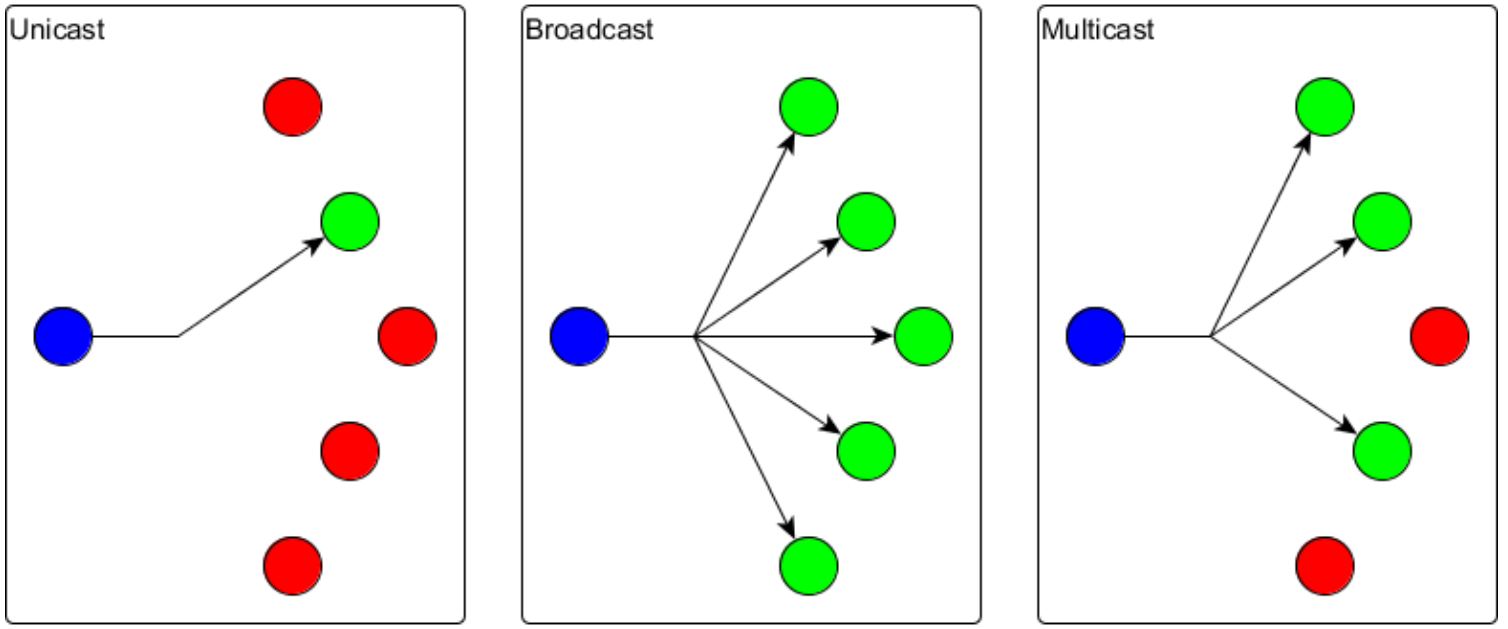
3. Endereço MAC

- Identificador único de uma interface de rede (NIC);
- Usado em muitos protocolos de enlace (Ethernet, WiFi, Bluetooth);
- Endereço físico;
- Identificado por 6 grupos (bytes) de dois números hexadecimais (0-F):
  - **fc:34:97:7a:e2:fa**



Tipos de Endereço

- **Unicast**: Endereçamento realizado a um único destino;
- **Multicast**: Endereçamento realizado a um grupo;
- **Broadcast**: Endereçamento realizado a todos.



3. Endereço MAC

- Como **verificar** o endereço físico da minha placa de rede?
  - » Windows: ipconfig /all
  - » Linux: ip link list (ou: ip l l)

```
→ ~ ip l l
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether fc:34:97:7a:e2:fa brd ff:ff:ff:ff:ff:ff
    altname enp0s31f6
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default qlen 1000
    link/ether 52:54:00:12:4a:81 brd ff:ff:ff:ff:ff:ff
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN mode DEFAULT group default qlen 1000
    link/ether 52:54:00:12:4a:81 brd ff:ff:ff:ff:ff:ff
5: lxcbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default qlen 1000
    link/ether 00:16:3e:00:00:00 brd ff:ff:ff:ff:ff:ff
```

2. Análise

- O que significa encaminhar **echo request** pelo nome e não receber **echo reply**?
- O que significa encaminhar **echo request** pelo endereço lógico e não receber **echo reply**?
- Como verificar o possível ponto de falha de um **echo request** usando o comando ping?
- O que é o protocolo ICMP?
- O que é o protocolo ARP?

Dúvidas?