



**Universidade do Minho**  
Escola de Engenharia

# Comunicações e Redes

**TP4**

## **Análise de Encapsulamento**



João Miguel da Silva Alves (83624)

Paulo Jorge Alves (84480)

**MESTRADO INTEGRADO EM ENGENHARIA BIOMÉDICA**  
**INFORMÁTICA MÉDICA 2020/2021**

O Modelo OSI é um modelo de referência da ISO que tem como principal objetivo ser um modelo standard, para protocolos de comunicação entre os mais diversos sistemas, e assim garantir a comunicação *end-to-end*.

O modelo é composto por 7 camadas (Aplicação; Apresentação; Sessão; Transporte; Rede; Dados; Física), em que cada camada realiza funções específicas.

O encapsulamento empacota as informações de protocolo necessárias antes de passar pela rede. Assim, à medida que o pacote de dados desce ou sobe pelas camadas do modelo OSI, ele recebe cabeçalhos e outras informações.

Uma vez que os dados são enviados da origem, eles viajam através da camada de aplicação para baixo através das outras camadas. O empacotamento e o fluxo dos dados que são trocados passam por alterações à medida que as redes executam seus serviços. [1]

A figura 1 exemplifica técnicas de encapsulamento, tomando como base o envio/recepção de uma Mensagem Aplicacional (HTTP, exemplo da figura).

Cada camada introduz o seu próprio cabeçalho, o que implica a introdução de um *overhead* adicional aos dados disponibilizados na camada acima.

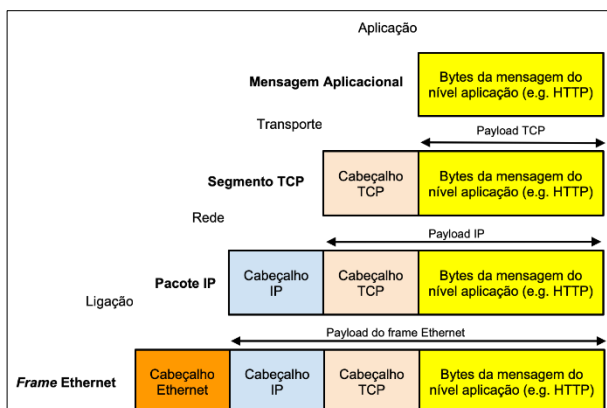


Figura 1 – Técnicas de encapsulamento

O *Wireshark* é um programa que monitoriza os pacotes que atravessam uma interface de rede de um computador, ou seja, analisa o tráfego de rede, e organiza-o por protocolos.

Utilizando o software *Wireshark* e a captura de tráfego originada no ficheiro fornecido pelo professor, analisar-se-ão os frames 15, 22 e 88.

Para cada frame, responder-se-á, sequencialmente, às seguintes questões.

- Quais os tamanhos dos dados transportados;
- Quais os valores de sobrecarga introduzidos pelas camadas de: Transporte e Rede;
- Quais os endereços IP utilizados como Origem (origem) e Destino (destino);
- Quais os ajustes de cada uma das transmissões, calculados como:  
$$\frac{\text{<nº bytes úteis de dados>}}{(\text{<nº bytes úteis de dados>} + \text{<nº total de bytes de sobrecarga>})};$$

## Análise do Frame 15

Pela figura 2, podemos verificar que o tamanho dos dados transportados corresponde a 426 bytes ou 3408 bits.

| Time         | Source          | Destination     | Protocol | Length | Info  |
|--------------|-----------------|-----------------|----------|--------|---|
| 10 47.835086 | 193.137.89.181  | 172.217.168.170 | HTTP     | 494    | GET /css?family=Merriweather:400,300,300italic,700,400italic,700... |
| 11 47.842273 | 193.137.89.181  | 151.101.133.46  | HTTP     | 446    | GET /css/social-icons.css?buildTime=1536346644 HTTP/1.1             |
| 12 47.842338 | 193.137.89.181  | 151.101.133.46  | HTTP     | 436    | GET /css/old/fancybox.css?1536346644 HTTP/1.1                       |
| 13 47.842405 | 193.137.89.181  | 151.101.133.46  | HTTP     | 429    | GET /js/lang/pt/stl.js?buildTime=1536346644 HTTP/1.1                |
| 14 47.844416 | 193.137.89.181  | 151.101.133.46  | HTTP     | 439    | GET /css/sites.css?buildTime=1536346644 HTTP/1.1                    |
| 15 47.844617 | 193.137.89.181  | 151.101.133.46  | HTTP     | 426    | GET /js/site/main.js?buildTime=1536346644 HTTP/1.1                  |
| 16 47.844626 | 193.137.89.181  | 151.101.133.46  | HTTP     | 449    | GET /js/site/main-customer-accounts-site.js?buildTime=1536346644... |
| 17 47.864387 | 151.101.133.46  | 193.137.89.181  | HTTP     | 1027   | Continuation  |
| 18 47.864389 | 151.101.133.46  | 193.137.89.181  | TCP      | 605    | 80 → 64315 [PSH, ACK] Seq=1239 Ack=371 Win=59 Len=539 TSval=1624... |
| 19 47.871812 | 172.217.168.170 | 193.137.89.181  | HTTP     | 86     | Continuation  |

Frame 15: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface en0, id 0  
Internet II, Src: Apple\_4b:3d:19 (8c:85:90:4b:3d:19), Dst: ComdaEnt\_ff:94:00 (00:d0:03:ff:94:00)  
Internet Protocol Version 4, Src: 193.137.89.181, Dst: 151.101.133.46  
Transmission Control Protocol, Src Port: 64317, Dst Port: 80, Seq: 1, Ack: 1, Len: 360  
Hypertext Transfer Protocol

|   |                    |
|---|--------------------|
| 00 d0 03 ff 94 00 8c 85 90 4b 3d 19 08 00 45 00 | .....K....E..      |
| 01 9c 00 00 40 00 40 06 01 8a c1 89 59 b5 97 65 | ....@..Y...e       |
| 05 2e fb 3d 00 50 ce db 70 b6 8d d0 e1 10 80 18 | ...=P...p.....     |
| 08 02 60 9a 00 00 01 01 08 0a 0b 4b 3b 91 60 d4 | ..Kj...            |
| cc 34 47 45 54 20 2f 6a 73 2f 73 69 74 65 2f 6d | -GET /j s/site/m   |
| 61 69 6e 2e 6a 73 3f 62 75 69 6c 64 54 69 6d 65 | ain.js?b uildTime  |
| 3d 31 35 33 36 33 34 36 36 34 34 20 48 54 54 50 | =1536346 644 HTTP  |
| 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 63 64 6e 32 | /1.1-Host: cdn2    |
| 2e 65 64 69 74 6d 79 73 69 74 65 2e 63 6f 6d 0d | .editmys ite.com   |
| 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 | -Accept- Encoding  |
| 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d | : gzip, deflate    |
| 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 | : Connect ion: kee |

Frame (frame), 426 byte(s) | Packets: 107 · Displayed: 107 (100.0%) | Profile:

Figura 2 – Frame 15 em detalhe

Pela figura 3, conseguimos retirar o valor de *overhead* ou *header* introduzido pela camada de transporte (TCP), que é 32 bytes ou 256 bits.

| No. | Time      | Source          | Destination     | Protocol | Length | Info  |
|-----|-----------|-----------------|-----------------|----------|--------|---|
| 10  | 47.835086 | 193.137.89.181  | 172.217.168.170 | HTTP     | 494    | GET /css?family=Merriweather:400,300,300italic,700,400italic,700... |
| 11  | 47.842273 | 193.137.89.181  | 151.101.133.46  | HTTP     | 446    | GET /css/social-icons.css?buildTime=1536346644 HTTP/1.1             |
| 12  | 47.842338 | 193.137.89.181  | 151.101.133.46  | HTTP     | 436    | GET /css/old/fancybox.css?1536346644 HTTP/1.1                       |
| 13  | 47.842405 | 193.137.89.181  | 151.101.133.46  | HTTP     | 429    | GET /js/lang/pt/stl.js?buildTime=1536346644 HTTP/1.1                |
| 14  | 47.844416 | 193.137.89.181  | 151.101.133.46  | HTTP     | 439    | GET /css/sites.css?buildTime=1536346644 HTTP/1.1                    |
| 15  | 47.844617 | 193.137.89.181  | 151.101.133.46  | HTTP     | 426    | GET /js/site/main.js?buildTime=1536346644 HTTP/1.1                  |
| 16  | 47.844626 | 193.137.89.181  | 151.101.133.46  | HTTP     | 449    | GET /js/site/main-customer-accounts-site.js?buildTime=1536346644... |
| 17  | 47.864387 | 151.101.133.46  | 193.137.89.181  | HTTP     | 1027   | Continuation  |
| 18  | 47.864389 | 151.101.133.46  | 193.137.89.181  | TCP      | 605    | 80 → 64315 [PSH, ACK] Seq=1239 Ack=371 Win=59 Len=539 TSval=1624... |
| 19  | 47.871812 | 172.217.168.170 | 193.137.89.181  | HTTP     | 86     | Continuation  |

Sequence Number: 1 (relative sequence number)  
Sequence Number (raw): 3470487734  
[Next Sequence Number: 361 (relative sequence number)]  
Acknowledgment Number: 1 (relative ack number)  
Acknowledgment number (raw): 2379276560  
1000 .... = Header Length: 32 bytes (8)  
> Flags: 0x018 (PSH, ACK)  
Window: 2050  
[Calculated window size: 2050]

|      |   |                    |
|------|---|--------------------|
| 0020 | 85 2e fb 3d 00 50 ce db 70 b6 8d d0 e1 10 80 18 | ...=P...p.....     |
| 0030 | 08 02 60 9a 00 00 01 01 08 0a 0b 4b 3b 91 60 d4 | ..Kj...            |
| 0040 | cc 34 47 45 54 20 2f 6a 73 2f 73 69 74 65 2f 6d | -GET /j s/site/m   |
| 0050 | 61 69 6e 2e 6a 73 3f 62 75 69 6c 64 54 69 6d 65 | ain.js?b uildTime  |
| 0060 | 3d 31 35 33 36 33 34 36 36 34 34 20 48 54 54 50 | =1536346 644 HTTP  |
| 0070 | 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 63 64 6e 32 | /1.1-Host: cdn2    |
| 0080 | 2e 65 64 69 74 6d 79 73 69 74 65 2e 63 6f 6d 0d | .editmys ite.com   |
| 0090 | 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 | -Accept- Encoding  |
| 00a0 | 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d | : gzip, deflate    |
| 00b0 | 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 | : Connect ion: kee |
| 00c0 | 70 2d 61 6c 69 76 65 0d 0a 41 63 63 65 70 74 3a | p-alive- Accept-   |
| 00d0 | 20 2a 2f 2a 0d 0a 55 73 65 72 2d 41 67 65 6e 74 | */ * -Us er-Agent  |

Transmission Control Protocol (tcp), 32 byte(s) | Packets: 107 · Displayed: 107 (100.0%) | Profile:

Figura 3 – Frame 15: TCP

Na figura 4, o valor de *overhead* ou *header* introduzido pela camada de rede é 20 bytes ou 160 bits.

| No. | Time      | Source          | Destination     | Protocol | Length | Info  |
|-----|-----------|-----------------|-----------------|----------|--------|---|
| 10  | 47.835086 | 193.137.89.181  | 172.217.168.170 | HTTP     | 494    | GET /css?family=Merriweather:400,300,300italic,700,400italic,700... |
| 11  | 47.842273 | 193.137.89.181  | 151.101.133.46  | HTTP     | 446    | GET /css/social-icons.css?buildTime=1536346644 HTTP/1.1             |
| 12  | 47.842338 | 193.137.89.181  | 151.101.133.46  | HTTP     | 436    | GET /css/old/fancybox.css?1536346644 HTTP/1.1                       |
| 13  | 47.842405 | 193.137.89.181  | 151.101.133.46  | HTTP     | 429    | GET /js/lang/pt/stl.js?buildTime=1536346644 HTTP/1.1                |
| 14  | 47.844416 | 193.137.89.181  | 151.101.133.46  | HTTP     | 439    | GET /css/sites.css?buildTime=1536346644 HTTP/1.1                    |
| 15  | 47.844617 | 193.137.89.181  | 151.101.133.46  | HTTP     | 426    | GET /js/site/main.js?buildTime=1536346644 HTTP/1.1                  |
| 16  | 47.844626 | 193.137.89.181  | 151.101.133.46  | HTTP     | 449    | GET /js/site/main-customer-accounts-site.js?buildTime=1536346644... |
| 17  | 47.864387 | 151.101.133.46  | 193.137.89.181  | HTTP     | 1027   | Continuation  |
| 18  | 47.864389 | 151.101.133.46  | 193.137.89.181  | TCP      | 605    | 80 → 64315 [PSH, ACK] Seq=1239 Ack=371 Win=59 Len=539 TSval=1624... |
| 19  | 47.871812 | 172.217.168.170 | 193.137.89.181  | HTTP     | 86     | Continuation  |

|   |  |
|---|--|
| > | Frame 15: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface en0, id 0   |
| > | Ethernet II, Src: Apple_4b:3d:19 (8c:85:90:4b:3d:19), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00) |
| > | Internet Protocol Version 4, Src: 193.137.89.181, Dst: 151.101.133.46                            |
| > | 0100 .... = Version: 4   |
| > | .... 0101 = Header Length: 20 bytes (5)  |
| > | Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)                                    |
| > | Total Length: 412  |
| > | Identification: 0x0000 (0)   |
| > | Flags: 0x40, Don't fragment  |

|      |   |                    |
|------|---|--------------------|
| 0000 | 00 d0 03 ff 94 00 8c 85 90 4b 3d 19 08 00 45 00 | .....K...E         |
| 0010 | 01 9c 00 00 40 00 00 00 01 8a c1 89 59 b5 97 65 | ...68...Y...t      |
| 0020 | 85 2c fb 3d 00 50 ce db 70 b6 8d d0 e1 10 80 18 | ...=P...p.....     |
| 0030 | 08 02 60 9a 00 00 01 01 08 0a 0b 4b 3b 91 60 d4 | ...Kj...           |
| 0040 | cc 34 47 45 54 20 2f 6a 73 2f 73 69 74 65 2f 6d | ..4GET /j s/site/m |
| 0050 | 61 69 6e 2e 6a 73 3f 62 75 69 6c 64 54 69 6d 65 | ain.js?b uildTime  |
| 0060 | 3d 31 35 33 36 33 34 36 36 34 34 20 48 54 54 50 | =1536346 644 HTTP  |
| 0070 | 2f 31 2e 31 00 0a 48 6f 73 74 3a 20 63 64 6e 32 | /1.1..Ho st: cdn2  |
| 0080 | 2e 65 64 69 74 6d 79 73 69 74 65 2e 63 6f 6d 0d | ..editmys ite.com  |
| 0090 | 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 | ..Accept- Encoding |
| 00a0 | 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d | : gzip, deflate    |
| 00b0 | 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 | ..Connect ion: kee |

Figura 4 – Frame 15: IP

Visualizando a figura anterior, podemos também observar o IP *source*: 193.137.89.181 e o IP *destination*: 151.101.133.46.

O número de bytes úteis de dados corresponde ao número de bytes transportados pela aplicação, portanto pelo HTTP, o qual corresponde a 360 bytes.

Para saber o número de bytes total, precisamos de somar o número de bytes úteis com o número de bytes, que, por sua vez, correspondem à soma dos valores de overhead introduzidos pela camada de transporte, pela camada de rede e pelo modelo físico.

$$\text{Rendimento} = \frac{\text{número de bytes úteis de dados}}{\text{número de bytes úteis de dados} + \text{número total de bytes de overhead}} =$$

$$\frac{360 \text{ bytes}}{360 + 32 + 20 + 14} * 100 = 84.507\%$$

## Análise do Frame 22

Pela figura 5, podemos verificar que o tamanho dos dados transportados corresponde a 1001 bytes ou 8008 bits.

| No. | Time      | Source          | Destination    | Protocol | Length | Info  |
|-----|-----------|-----------------|----------------|----------|--------|---|
| 19  | 47.871812 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 20  | 47.871814 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 21  | 47.877167 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 22  | 47.885676 | 151.101.133.46  | 193.137.89.181 | HTTP     | 1001   | Continuation  |
| 23  | 47.894642 | 151.101.133.46  | 193.137.89.181 | HTTP     | 286    | Continuation  |
| 24  | 47.943916 | 151.101.133.46  | 193.137.89.181 | HTTP     | 399    | Continuation  |
| 25  | 47.966137 | 151.101.133.46  | 193.137.89.181 | HTTP     | 194    | Continuation  |
| 26  | 47.970875 | 199.34.229.100  | 193.137.89.181 | HTTP     | 71     | [TCP Previous segment not captured] Continuation                    |
| 27  | 47.980683 | 193.137.89.181  | 199.34.229.100 | HTTP     | 449    | [TCP ACKed unseen segment] GET /files/theme/plugins.js?153558918... |
| 28  | 47.985209 | 193.137.89.181  | 199.34.229.100 | HTTP     | 486    | GET /gdpr/gdprscript.js?buildTime=1536346644&hasRemindMe=true&st... |

> Frame 22: 1001 bytes on wire (8008 bits), 1001 bytes captured (8008 bits) on interface en0, id 0  
> Ethernet II, Src: ComdaEnt\_ff:94:00 (00:00:03:ff:94:00), Dst: Apple\_4b:3d:19 (8c:85:90:4b:3d:19)  
> Internet Protocol Version 4, Src: 151.101.133.46, Dst: 193.137.89.181  
> Transmission Control Protocol, Src Port: 80, Dst Port: 64318, Seq: 23523, Ack: 364, Len: 935  
> Hypertext Transfer Protocol

0000 8c 85 90 4b 3d 19 00 d0 03 ff 94 00 08 00 45 00 ...K=... ..E  
0010 03 db 6e a8 40 00 38 06 98 a2 97 65 85 2e c1 89 ...n-g-B-...e...  
0020 59 b5 00 50 f0 3e 6d 2e 8b d1 3d cc 87 15 80 18 Y...P>m... ..  
0030 00 3b 66 b9 00 00 01 01 08 0a 1a 9a 41 81 0b 4b ...f... ..A-K  
0040 3b a3 56 c5 c4 14 d1 8e 57 f4 d2 d3 49 c7 4f b3 ;V... W...I-O  
0050 33 5b ad ec d0 f3 42 00 74 61 cd c7 ad 3d 04 41 3[...B: ta...=A  
0060 9c 9a 6e 26 66 51 5f 4a fd 49 3c 81 48 1d 36 54 ...n&FQ\_] -Ic-H-6T  
0070 8c 1a 35 83 1a 23 c5 43 8f f9 58 2e cd b0 96 6a ...5...#-C-X...j  
0080 32 6a 0a c8 34 7b 31 a2 32 27 04 14 f7 50 84 71 2j...4{1- 2'...P-q  
0090 cb 55 84 ab fe ac 86 ac a3 a4 12 dc 81 00 b0 ba -U... ..  
00a0 d6 3c 45 6e aa 07 fa 35 b7 c9 af 32 b7 cc 93 d5 ...En...5 ...2...  
00b0 7a 27 a6 18 19 d7 ea f7 79 c8 98 86 32 8d 8d 23 z'... ..y...2...#

Frame (frame), 1001 byte(s) Packets: 107 · Displayed: 107 (100.0%) Profile:

Figura 5 – Frame 22 em detalhe

Pela figura 6, conseguimos retirar o valor de *overhead* ou *header* introduzido pela camada de transporte (TCP), que é 32 bytes ou 256 bits.

Wireshark - exemplo wireshark.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter: <Ctrl-/>

| No. | Time      | Source          | Destination    | Protocol | Length | Info  |
|-----|-----------|-----------------|----------------|----------|--------|---|
| 19  | 47.871812 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 20  | 47.871814 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 21  | 47.877167 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 22  | 47.885676 | 151.101.133.46  | 193.137.89.181 | HTTP     | 1001   | Continuation  |
| 23  | 47.894642 | 151.101.133.46  | 193.137.89.181 | HTTP     | 286    | Continuation  |
| 24  | 47.943916 | 151.101.133.46  | 193.137.89.181 | HTTP     | 399    | Continuation  |
| 25  | 47.966137 | 151.101.133.46  | 193.137.89.181 | HTTP     | 194    | Continuation  |
| 26  | 47.970875 | 199.34.229.100  | 193.137.89.181 | HTTP     | 71     | [TCP Previous segment not captured] Continuation                    |
| 27  | 47.980683 | 193.137.89.181  | 199.34.229.100 | HTTP     | 449    | [TCP ACKed unseen segment] GET /files/theme/plugins.js?153558918... |
| 28  | 47.985209 | 193.137.89.181  | 199.34.229.100 | HTTP     | 486    | GET /gdpr/gdprscript.js?buildTime=1536346644&hasRemindMe=true&st... |

Sequence Number: 23523 (relative sequence number)  
Sequence Number (raw): 1831766993  
[Next Sequence Number: 24458 (relative sequence number)]  
Acknowledgment Number: 364 (relative ack number)  
Acknowledgment number (raw): 1036814101  
1000 .... = Header Length: 32 bytes (8)  
> Flags: 0x018 (PSH, ACK)  
Window: 59  
[Calculated window size: 59]

0020 59 b5 00 50 f0 3e 6d 2e 8b d1 3d cc 87 15 80 18 Y...P>m... ..  
0030 00 3b 66 b9 00 00 01 01 08 0a 1a 9a 41 81 0b 4b ...f... ..A-K  
0040 3b a3 56 c5 c4 14 d1 8e 57 f4 d2 d3 49 c7 4f b3 ;V... W...I-O  
0050 33 5b ad ec d0 f3 42 00 74 61 cd c7 ad 3d 04 41 3[...B: ta...=A  
0060 9c 9a 6e 26 66 51 5f 4a fd 49 3c 81 48 1d 36 54 ...n&FQ\_] -Ic-H-6T  
0070 8c 1a 35 83 1a 23 c5 43 8f f9 58 2e cd b0 96 6a ...5...#-C-X...j  
0080 32 6a 0a c8 34 7b 31 a2 32 27 04 14 f7 50 84 71 2j...4{1- 2'...P-q  
0090 cb 55 84 ab fe ac 86 ac a3 a4 12 dc 81 00 b0 ba -U... ..  
00a0 d6 3c 45 6e aa 07 fa 35 b7 c9 af 32 b7 cc 93 d5 ...En...5 ...2...  
00b0 7a 27 a6 18 19 d7 ea f7 79 c8 98 86 32 8d 8d 23 z'... ..y...2...#  
00c0 a5 50 aa 51 85 fc b6 22 43 9d 06 5a 1b 5f 2d 04 -P-Q... " C-Z...  
00d0 fc b7 64 af d3 9d a2 f7 dc 7c d2 54 98 09 55 50 -d... ..|...T...UP

Transmission Control Protocol (tcp), 32 byte(s) Packets: 107 · Displayed: 107 (100.0%) Profile: Default

Escreva aqui para procurar

18:23 03/11/2020

Figura 6 – Frame 22: TCP

Na figura 7, o valor de overhead ou header introduzido pela camada de rede é 20 bytes ou 160 bits.

| No. | Time      | Source          | Destination    | Protocol | Length | Info  |
|-----|-----------|-----------------|----------------|----------|--------|---|
| 19  | 47.871812 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 20  | 47.871814 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 21  | 47.877167 | 172.217.168.170 | 193.137.89.181 | HTTP     | 86     | Continuation  |
| 22  | 47.885676 | 151.101.133.46  | 193.137.89.181 | HTTP     | 1001   | Continuation  |
| 23  | 47.894642 | 151.101.133.46  | 193.137.89.181 | HTTP     | 286    | Continuation  |
| 24  | 47.943916 | 151.101.133.46  | 193.137.89.181 | HTTP     | 399    | Continuation  |
| 25  | 47.966137 | 151.101.133.46  | 193.137.89.181 | HTTP     | 194    | Continuation  |
| 26  | 47.979875 | 199.34.229.100  | 193.137.89.181 | HTTP     | 71     | [TCP Previous segment not captured] Continuation                    |
| 27  | 47.980683 | 193.137.89.181  | 199.34.229.100 | HTTP     | 449    | [TCP ACKed unseen segment] GET /files/theme/plugins.js?153558918... |
| 28  | 47.985209 | 193.137.89.181  | 199.34.229.100 | HTTP     | 486    | GET /gdpr/gdprscript.js?buildTime=1536346644&hasRemindMe=true&st... |

```

> Frame 22: 1001 bytes on wire (8008 bits), 1001 bytes captured (8008 bits) on interface en0, id 0
> Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00), Dst: Apple_4b:3d:19 (8c:85:90:4b:3d:19)
> Internet Protocol Version 4, Src: 151.101.133.46, Dst: 193.137.89.181
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 987
  Identification: 0x6ea8 (28328)
  > Flags: 0x40, Don't fragment
  > .....
```

```

0000  8c 85 90 4b 3d 19 00 d0 03 ff 94 00 08 00 45 00  ...K.....E
0010  03 db 6e a8 40 00 38 06 98 a2 97 65 85 2e c1 89  ...n-g-8-...e...
0020  59 b5 00 50 f0 3e 6d 2e 8b d1 3d cc 87 15 80 18  ...Y.P>m,....
0030  00 3b 66 b9 00 00 01 01 08 0a 1a 9a 41 81 0b 4b  ...;f.....A..K
0040  3b a3 56 c5 c4 14 d1 8e 57 f4 d2 d3 49 c7 4f b3  ...;V.....W...I.O
0050  33 5b ad ec d0 f3 42 d0 74 61 cd c7 ad 3d d4 41  3[....B.ta...A
0060  9c 9a 6e 26 66 51 5f 4a fd 49 3c 81 48 1d 36 54  ...n&fQ_)I..H.6T
0070  8c 1a 35 83 1a 23 c5 43 8f f9 58 2e cd b0 96 6a  ...5-#C-X...j
0080  32 6a 0a c8 34 7b 31 a2 32 27 04 14 f7 50 84 71  2j...4{1.2'...P.q
0090  cb 55 84 ab fe ac 86 ac a3 a4 12 dc 81 00 b0 ba  ...U.....
00a0  d6 3c 45 6e aa 07 fa 35 b7 c9 af 32 b7 cc 93 d5  <En...5...2...
00b0  7a 27 a6 18 19 d7 ea f7 79 c8 98 86 32 8d 8d 23  z'.....y...2-#
```

Internet Protocol Version 4 (IP), 20 byte(s)      Packets: 107 · Displayed: 107 (100.0%)      Profile:

Figura 7 – Frame 22: IP

Visualizando a figura anterior, podemos também observar o IP *source*: 151.101.133.46 e o IP *destination*: 193.137.89.181.

O número de bytes úteis de dados corresponde ao número de bytes transportados pela aplicação, portanto pelo HTTP, o qual corresponde a 935 bytes.

Para saber o número de bytes total, necessitamos de somar o número de bytes úteis com o número de bytes, que, por sua vez, correspondem à soma dos valores de overhead introduzidos pela camada de transporte, pela camada de rede e pelo modelo físico.

$$\text{Rendimento} = \frac{\text{número de bytes úteis de dados}}{\text{número de bytes úteis de dados} + \text{número total de bytes de overhead}} =$$

$$\frac{935 \text{ bytes}}{935 + 32 + 20 + 14} * 100 = 93.41\%$$

## Análise do Frame 88

Pela figura 8, podemos verificar que o tamanho dos dados transportados corresponde a 103 bytes ou 824 bits.

| No. | Time      | Source         | Destination    | Protocol  | Length | Info  |
|-----|-----------|----------------|----------------|-----------|--------|---|
| 82  | 48.854814 | 193.137.89.181 | 199.34.229.100 | HTTP      | 535    | [TCP ACKed unseen segment] GET /uploads/1/8/8/0/18802844/publish... |
| 83  | 48.879532 | 199.34.229.100 | 193.137.89.181 | HTTP      | 564    | [TCP Previous segment not captured] Continuation                    |
| 84  | 48.880514 | 193.137.89.181 | 199.34.229.100 | HTTP      | 535    | [TCP ACKed unseen segment] GET /uploads/1/8/8/0/18802844/publish... |
| 85  | 49.044239 | 199.34.229.100 | 193.137.89.181 | HTTP      | 1098   | [TCP Previous segment not captured] Continuation                    |
| 86  | 49.045638 | 193.137.89.181 | 199.34.229.100 | HTTP      | 535    | [TCP ACKed unseen segment] GET /uploads/1/8/8/0/18802844/publish... |
| 87  | 49.278539 | 199.34.229.100 | 193.137.89.181 | HTTP      | 813    | [TCP Previous segment not captured] Continuation                    |
| 88  | 49.278545 | 199.34.229.100 | 193.137.89.181 | TCP       | 103    | [TCP Previous segment not captured] 80 → 64305 [PSH, ACK] Seq=21... |
| 89  | 49.278545 | 199.34.229.100 | 193.137.89.181 | HTTP/J... | 785    | [TCP ACKed unseen segment] HTTP/1.1 200 OK , JavaScript Object N... |
| 90  | 49.278564 | 199.34.229.100 | 193.137.89.181 | HTTP      | 791    | [TCP Previous segment not captured] Continuation                    |
| 91  | 49.278575 | 199.34.229.100 | 193.137.89.181 | HTTP      | 1148   | [TCP Previous segment not captured] Continuation                    |

> Frame 88: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface en0, id 0

> Ethernet II, Src: ComdaEnt\_ff:94:00 (00:d0:03:ff:94:00), Dst: Apple\_4b:3d:19 (8c:85:9b:4b:3d:19)

> Internet Protocol Version 4, Src: 199.34.229.100, Dst: 193.137.89.181

> Transmission Control Protocol, Src Port: 80, Dst Port: 64305, Seq: 21768, Ack: 2210, Len: 37

0000 8c 85 90 4b 3d 19 00 00 03 ff 94 00 08 00 45 00 ...K=... ..E..

0010 00 59 e8 38 00 00 ee 06 1c a0 c7 22 e5 64 c1 89 ...Y.8... ..d..

0020 59 b5 00 50 fb 31 a7 f6 15 79 16 07 06 f3 80 18 ...Y..P.1.. ..y....

0030 30 5c 62 8b 00 00 01 01 08 0a 12 8b 03 16 06 4b ...0\b.....K...

0040 3f 72 2e b1 d4 5f 5e 5f 24 19 47 47 c6 ce 4e 68 ...p... ^ \$GG..Nh

0050 27 c4 94 19 10 38 9b 0b ec 6c 10 6d 20 d8 20 bc ...'...8...l.m...

0060 80 80 80 80 83 ff d9 .....

Frame (frame), 103 byte(s) Packets: 107 · Displayed: 107 (100.0%) Profile

Figura 8 – Frame 88 em detalhe

Pela figura 9, conseguimos retirar o valor de overhead ou header introduzido pela camada de transporte (TCP), que é 32 bytes ou 256 bits.

| No. | Time      | Source         | Destination    | Protocol  | Length | Info  |
|-----|-----------|----------------|----------------|-----------|--------|---|
| 82  | 48.854814 | 193.137.89.181 | 199.34.229.100 | HTTP      | 535    | [TCP ACKed unseen segment] GET /uploads/1/8/8/0/18802844/publish... |
| 83  | 48.879532 | 199.34.229.100 | 193.137.89.181 | HTTP      | 564    | [TCP Previous segment not captured] Continuation                    |
| 84  | 48.880514 | 193.137.89.181 | 199.34.229.100 | HTTP      | 535    | [TCP ACKed unseen segment] GET /uploads/1/8/8/0/18802844/publish... |
| 85  | 49.044239 | 199.34.229.100 | 193.137.89.181 | HTTP      | 1098   | [TCP Previous segment not captured] Continuation                    |
| 86  | 49.045638 | 193.137.89.181 | 199.34.229.100 | HTTP      | 535    | [TCP ACKed unseen segment] GET /uploads/1/8/8/0/18802844/publish... |
| 87  | 49.278539 | 199.34.229.100 | 193.137.89.181 | HTTP      | 813    | [TCP Previous segment not captured] Continuation                    |
| 88  | 49.278545 | 199.34.229.100 | 193.137.89.181 | TCP       | 103    | [TCP Previous segment not captured] 80 → 64305 [PSH, ACK] Seq=21... |
| 89  | 49.278545 | 199.34.229.100 | 193.137.89.181 | HTTP/J... | 785    | [TCP ACKed unseen segment] HTTP/1.1 200 OK , JavaScript Object N... |
| 90  | 49.278564 | 199.34.229.100 | 193.137.89.181 | HTTP      | 791    | [TCP Previous segment not captured] Continuation                    |
| 91  | 49.278575 | 199.34.229.100 | 193.137.89.181 | HTTP      | 1148   | [TCP Previous segment not captured] Continuation                    |

[Stream Index: 12]

[TCP Segment Len: 37]

Sequence Number: 21768 (relative sequence number)

Sequence Number (raw): 2817922425

[Next Sequence Number: 21805 (relative sequence number)]

Acknowledgment Number: 2210 (relative ack number)

Acknowledgment number (raw): 369559283

1000 .... = Header Length: 32 bytes (8)

> Flags: 0x018 (PSH, ACK)

0000 8c 85 90 4b 3d 19 00 00 03 ff 94 00 08 00 45 00 ...K=... ..E..

0010 00 59 e8 38 00 00 ee 06 1c a0 c7 22 e5 64 c1 89 ...Y.8... ..d..

0020 59 b5 00 50 fb 31 a7 f6 15 79 16 07 06 f3 80 18 ...Y..P.1.. ..y....

0030 30 5c 62 8b 00 00 01 01 08 0a 12 8b 03 16 06 4b ...0\b.....K...

0040 3f 72 2e b1 d4 5f 5e 5f 24 19 47 47 c6 ce 4e 68 ...p... ^ \$GG..Nh

0050 27 c4 94 19 10 38 9b 0b ec 6c 10 6d 20 d8 20 bc ...'...8...l.m...

0060 80 80 80 80 83 ff d9 .....

Transmission Control Protocol (tcp), 32 byte(s) Packets: 107 · Displayed: 107 (100.0%) Profile

Figura 9 – Frame 88: TCP



Na figura 10, o valor de overhead ou header introduzido pela camada de rede é 20 bytes ou 160 bits.

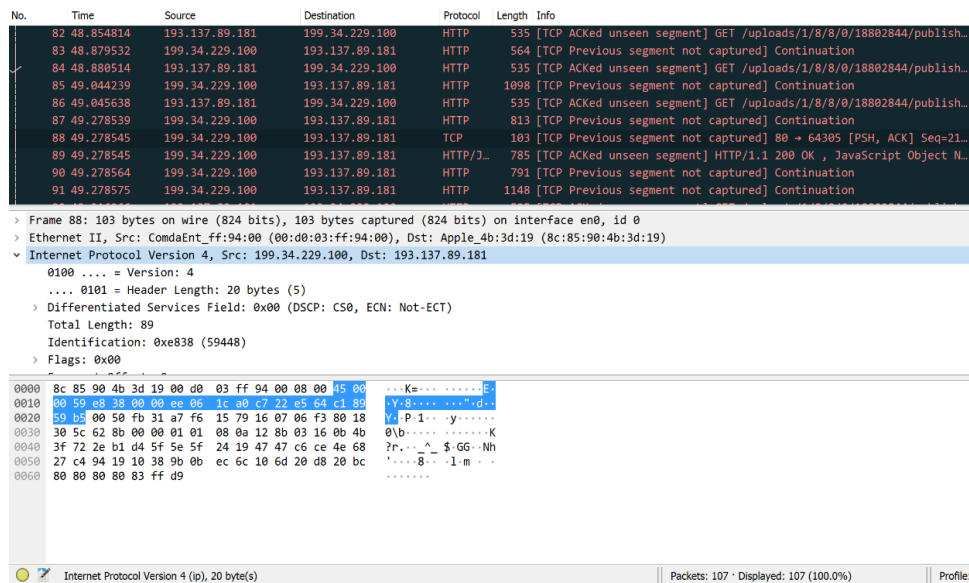


Figura 10 – Frame 88: IP

Visualizando a figura anterior, podemos também observar o IP *source*: 199.34.229.100 e o IP *destination*: 193.137.89.181.

O número de bytes úteis de dados corresponde ao número de bytes transportados pela camada de transporte (TCP), o qual corresponde a 37 bytes.

Para saber o número de bytes total, necessitamos de somar o número de bytes úteis com o número de bytes, que, por sua vez, correspondem à soma dos valores de overhead introduzidos pela camada de transporte, pela camada de rede e pelo modelo físico.

$$\text{Rendimento} = \frac{\text{número de bytes úteis de dados}}{\text{número de bytes úteis de dados} + \text{número total de bytes de overhead}} =$$

$$\frac{37 \text{ bytes}}{37 + 32 + 20 + 14} * 100 = 35.922\%$$



## **Análise dos resultados obtidos:**

A análise dos resultados obtidos anteriormente, permite compreender que quando a mensagem ou a aplicação que pretendemos transmitir se encontra na camada da aplicação, por exemplo, HTTP, o rendimento calculado é significativamente superior ao caso quando a mensagem a transmitir se encontra na camada do TCP/IP. Tal acontece, uma vez que o número útil de dados na primeira situação é superior ao da segunda situação.

Nas várias frames analisadas, a ideia principal é a existência de encapsulamento dos dados. O encapsulamento possui vantagens, mas também desvantagens relativamente à forma como os dados podem ser transmitidos de uma fonte para um destino.

Por um lado, as vantagens do encapsulamento são: promove segurança e confiabilidade na transmissão dos dados. Permite manutenção regular e precisa dos dados a transmitir, o código elaborado para este fim pode ser alterado independentemente e, além disso, aumenta a usabilidade. Submeter os dados e colocá-los em partes mais pequenas permite identificar as camadas onde ocorre um determinado conjunto de problemas.

Por outro lado, é fácil compreender que o uso desta técnica faz com que seja necessário colocar mais dados e informação aos dados relativos à nossa aplicação para a poder transmitir de um local para outro. Além disso, incrementar dados e servidores torna-se muito caro manter ao longo do tempo. É também necessário, caso a network se torne cada vez maior, uma melhor administração e para tal, é preciso um maior treino e preparação para o departamento de informática das empresas que realizam este encapsulamento. Por fim, existe sempre o risco de aparecimento de vírus ou que alguém possa hackear a transmissão. Portanto, para prevenir que isso aconteça, é necessário implementar firewalls, aumentando, assim, o custo total requerido para a transmissão. <sup>[2]</sup> <sup>[3]</sup>

## **Bibliografia:**

- [1] <https://pplware.sapo.pt/tutoriais/networking/redes-sabe-o-que-e-o-modelo-osi/>
- [2] <https://www.quora.com/What-is-the-main-benefit-of-encapsulation-in-networking>
- [3] <https://www.ukessays.com/essays/computer-science/networking-standards-osi-layers-advantages-and-disadvantages.php>