

ENCAPSULATION ANALYSIS

Comunicação e Redes, 17 de novembro de 2020

Beatriz Ceba 83813
Beatriz Soares 85815
Grupo 5
4º ano Engenharia Biomédica,
Informática Médica

O *Wireshark* é um programa capaz de monitorizar os pacotes que atravessam uma interface de rede de um computador (*host*). Esta ferramenta captura o tráfego da rede local e armazena esses dados para análise *offline*.

A figura 1 exemplifica técnicas de encapsulamento, tomando como base o envio/receção de uma Mensagem Aplicacional HTTP. A mensagem é enviada de cima para baixo, ou seja, desde o HTTP atravessa as restantes camadas: de transporte (segmento TCP), de rede (pacote IP) e, finalmente, a camada física, Ethernet. Cada camada introduz o seu próprio Cabeçalho (*Header*), o que implica a introdução de um *overhead* adicional aos dados (*Payload*) disponibilizados na camada acima. Esses cabeçalhos contêm informações que são úteis aos vários protocolos, por exemplo, na camada de rede, é introduzido o cabeçalho IP que tem informações como o endereço IP, o TTL do pacote e o número de saltos. A mensagem, depois de recebida, faz o percurso inverso. Assim, o encapsulamento consiste na passagem da mensagem pelas várias camadas, sendo-lhe adicionados os vários cabeçalhos à medida que atravessa cada uma delas.

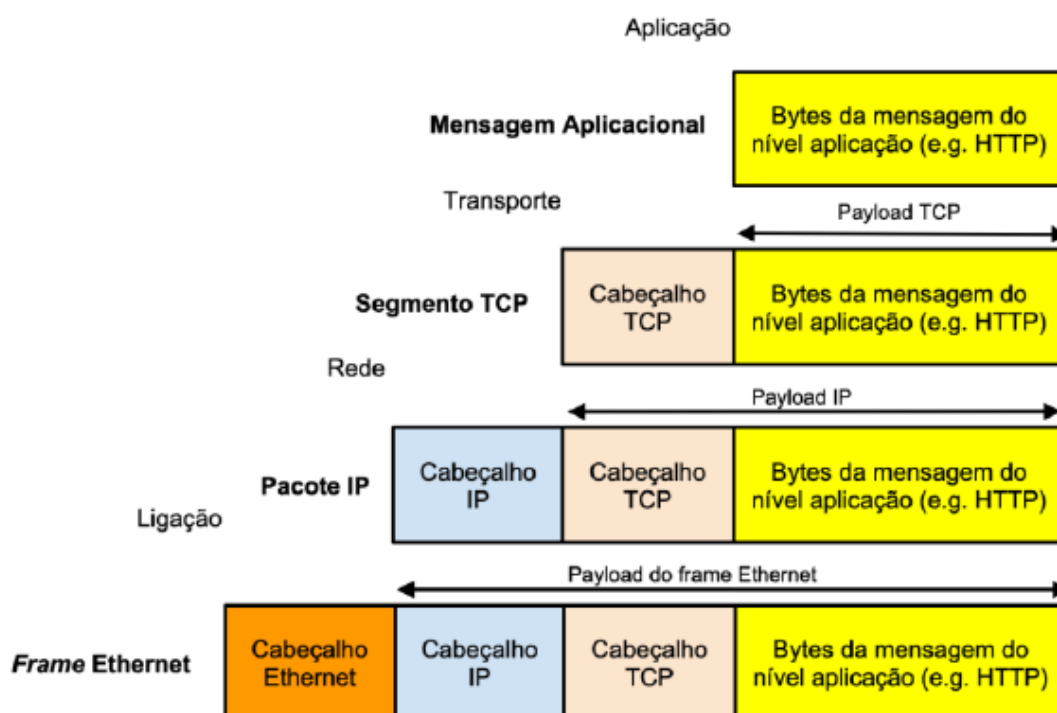


Figura 1 Encapsulamento da mensagem aplicacional

1) a. b. e c.

Frame	Tamanho dos dados	Overhead			IP	
		Transporte	Rede	Ethernet	Origem	Destino
43	69 bytes (552 bits)	32 bytes	20 bytes	14 bytes	199.34.229.100	193.137.89.181
56	1280 bytes (10240 bits)	32 bytes	20 bytes	14 bytes	172.217.168.163	193.137.89.181
104	191 bytes (1528 bits)	32 bytes	20 bytes	14 bytes	199.34.229.100	193.137.89.181

d.

O número de *bytes* úteis é o número de bytes da mensagem do nível de aplicação (HTTP, no exemplo). Podemos também pensar que o número de *bytes* úteis é o tamanho da *frame* Ethernet, ao qual é subtraído o *overhead* introduzido pelas camadas de transporte, rede e Ethernet.

Frame 43:

Nº bytes úteis = 3 (= 69-32-20-14)

Nº total de bytes = 69

$$rendimento = \frac{3}{69} = 0,043$$

Frame 56:

Nº bytes úteis = 1214 (= 1280-32-20-14)

Nº total de bytes = 1280

$$rendimento = \frac{1214}{1280} = 0,95$$

Frame 104:

Nº bytes úteis = 125 (= 191-32-20-14)

Nº total de bytes = 191

$$rendimento = \frac{125}{191} = 0,65$$

Assim, no que diz respeito à articulação entre camadas, tem-se que [1]:

- Cada camada superior faz uso dos serviços da camada inferior e presta serviços à de cima;
- Quando uma camada recebe dados da camada acima, a existência de um protocolo obriga à adição de informação de controlo com um determinado tamanho, que implica a introdução de um *overhead* adicional aos dados. Como podemos observar nos resultados, a camada de transporte introduz um *overhead* de 32 bytes aos dados, enquanto que as camadas de rede e *Ethernet* introduzem um *overhead* de 20 e 14 bytes, respetivamente;
- O resultado obtido é enviado para a camada abaixo;
- Esse processo de adicionar informação ao passar pelas diversas camadas chama-se encapsulamento;
- Do lado do recetor, este processo é inverso e chama-se desencapsulamento.

Esta disposição em camadas permite que um nível possa controlar também a informação tratada num nível mais baixo, o que implica maior qualidade nos serviços oferecidos pela rede. Esta é uma das vantagens das técnicas de encapsulamento. Existem outras como [1]:

- A decomposição das comunicações de redes em partes menores, facilitando a sua aprendizagem e compreensão;
- Facilita a programação modular (independente), evitando que modificações numa camada afetem outras;
- Permite a comunicação entre diferentes sistemas;
- Adoção de normas (*standards* de protocolos).

Pode considerar-se as normas ou protocolos um mal necessário pois, se é bom que elas garantam que máquinas de diferentes fabricantes possam comunicar entre si, normalmente, a definição de normas é um processo lento. Porém, elas são importantes, pois na comunicação são necessárias regras que definam o que fazer em determinada situação e procedimentos que determinem o caminho a dar a determinada informação.

Porém, a introdução dos cabeçalhos que contêm informações úteis aos vários protocolos, IP, TCP, etc., tem um custo, o tal *overhead*. Ora, a introdução de cabeçalhos

demasiado grandes para o tamanho da mensagem faz com que o rendimento da transmissão seja baixo, isto é, o tamanho dos dados efetivamente úteis (os *bytes* da mensagem aplicacional) é pequeno quando comparado com o tamanho de *overhead* que lhe foi adicionado (*Payload*). Isto é o que acontece no caso da *frame* 43, cujo rendimento de transmissão é de apenas 4 %, aproximadamente. Ou seja, se o tamanho da informação necessária para enviar uma mensagem é maior do que o tamanho da própria mensagem, a transmissão não é rentável. Esta é uma das desvantagens e quiçá a mais importante, associada às técnicas de encapsulamento. Por outro lado, para mensagens grandes, isto é, cujo tamanho seja superior à carga de transmissão de dados, o rendimento de transmissão é alto. Isto é o que acontece no caso da *frame* 56, cujo rendimento de transmissão é de 95 %.

Bibliografia

- [1] L. Oliveira, "Vantagens do modelo OSI e articulação entre camadas", *Redes de Comunicação*, 2020. [Online]. Available: <https://tozehgamer.wordpress.com/vantagens-do-modelo-osi-e-articulacao-entre-camadas/>. [Accessed Nov 06, 2020]