

REDES DE COMPUTADORES

2021/2022

aula 0010 - Redes de Redes

17/02/2022

Pedro Patinho <pp@di.uevora.pt>

Universidade de Évora - Departamento de Informática



SUMÁRIO

1. Redes como Sistemas Complexos

2. Camadas e Encapsulamento

3. Encapsulamento

4. A Internet como Rede de Redes

5. Transmissão digital de dados

6. Transmissão de dados

7. Erros de transmissão

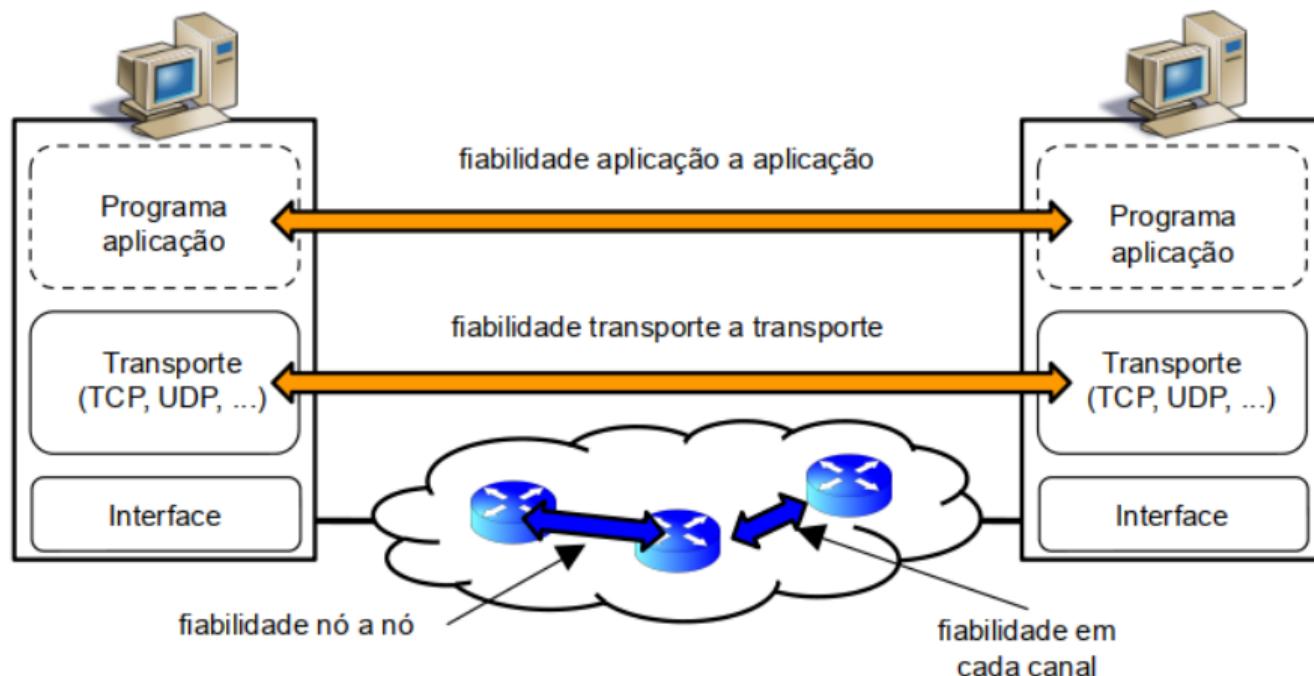
REDES COMO SISTEMAS COMPLEXOS

PRINCÍPIOS E DIRECTRIZES

- *End-to-End*
- Abstracção - isolamento entre camadas/níveis
- Encapsulamento
- Afectação de recursos adequados a cada camada
- Recursividade: redes de redes de redes...

- Funcionalidades necessárias às aplicações
 - Compressão
 - Autenticação
 - Segurança
 - Fiabilidade
 - Portabilidade (quando o host se “move”)
- Seria tudo tratado pela rede (???)

END-TO-END



VANTAGENS

- Só se “paga” o que se usa
- O sistema é mais flexível e mais extensível
- Novas necessidades podem dar origem a novas “peças do puzzle”
- Protocolos podem evoluir de forma independente (e.g., IPV6 vs IPV4)

ESTRUTURA DA REDE

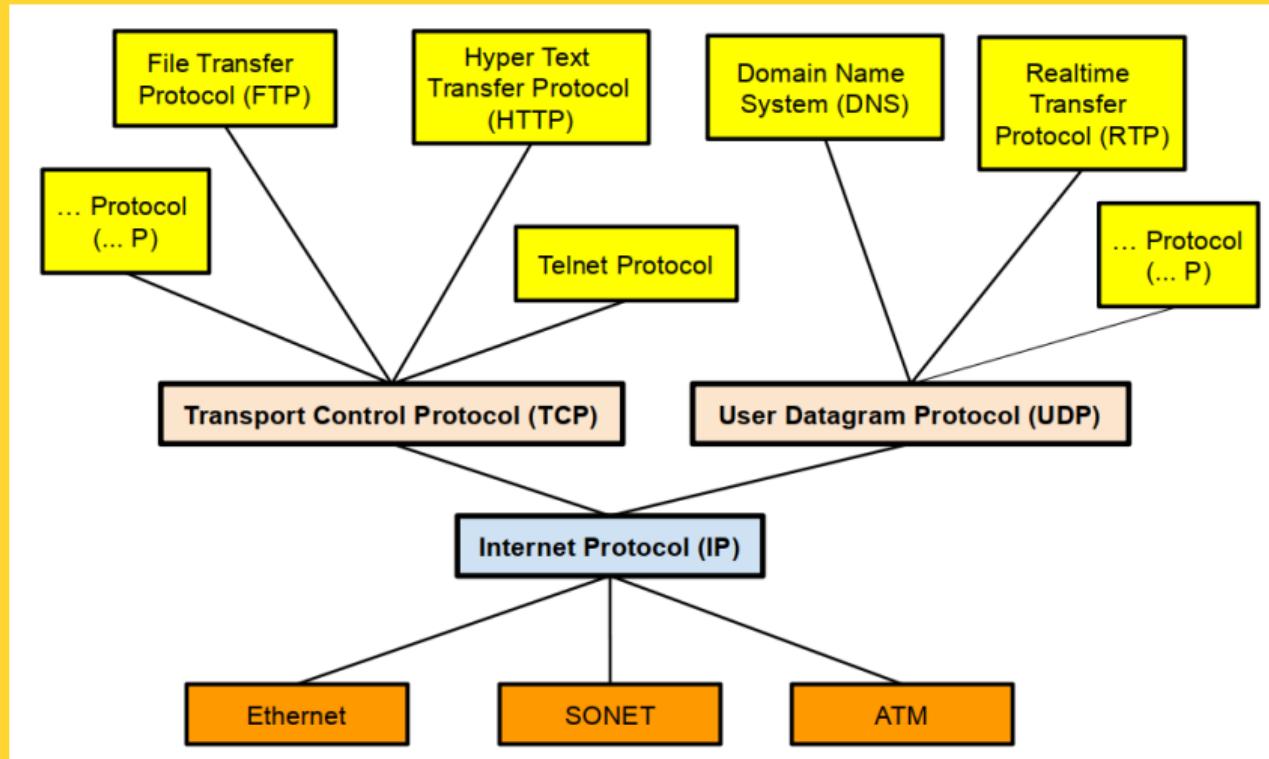
- Diversas componentes: hosts, nós, canais, protocolos, aplicações
- Configuração física da rede em forma de sub-redes
- Visão da rede como uma abstracção em termos de camadas
- Cada camada representa um conjunto de funcionalidades que apresentam um conjunto de serviços bem definidos à camada superior
- Cada camada usa os serviços da camada inferior

EXEMPLOS

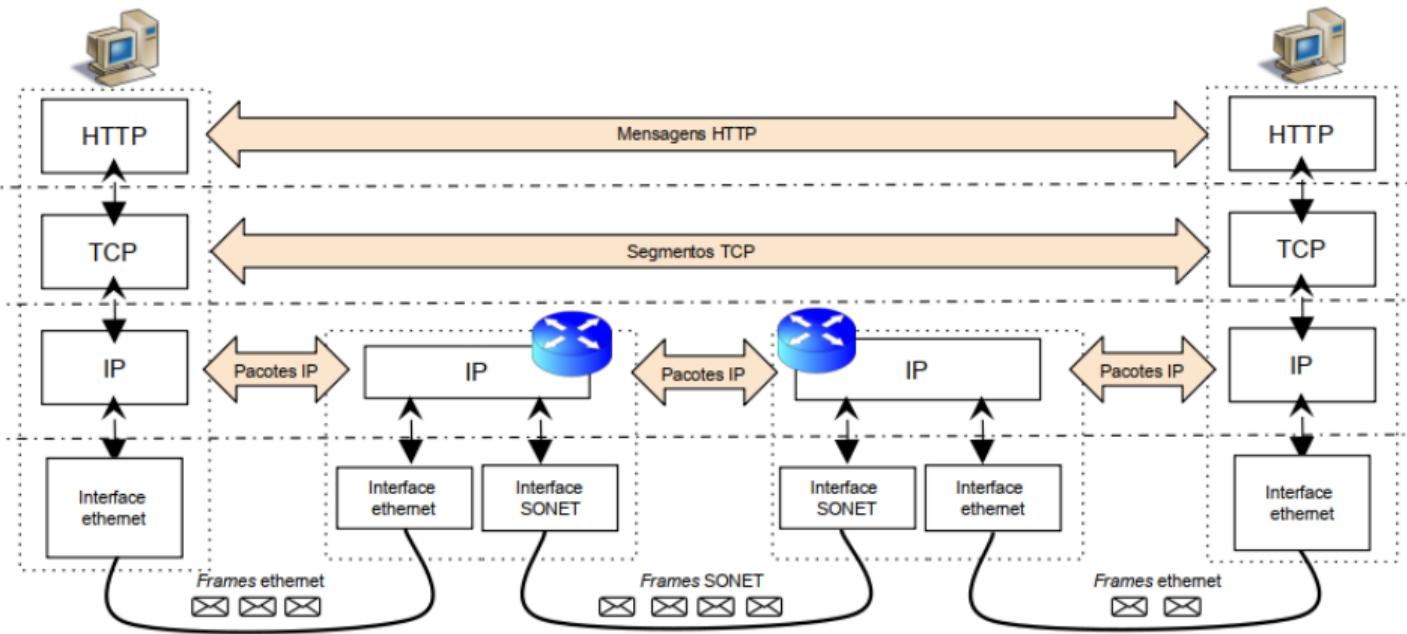
- Protocolo de aplicação (p.ex. HTTP) não necessita de saber
 - Como é implementado o TCP
 - Qual a configuração da rede em termos de encaminhamento (routing)
- O algoritmo de encaminhamento não é definido em função das características físicas dos canais nem do funcionamento do TCP, ou mesmo HTTP

CAMADAS E ENCAPSULAMENTO

CAMADAS/NÍVEIS EM TCP/IP



PROTOCOLOS E CAMADAS/NÍVEIS



TAMBÉM HÁ DESVANTAGENS

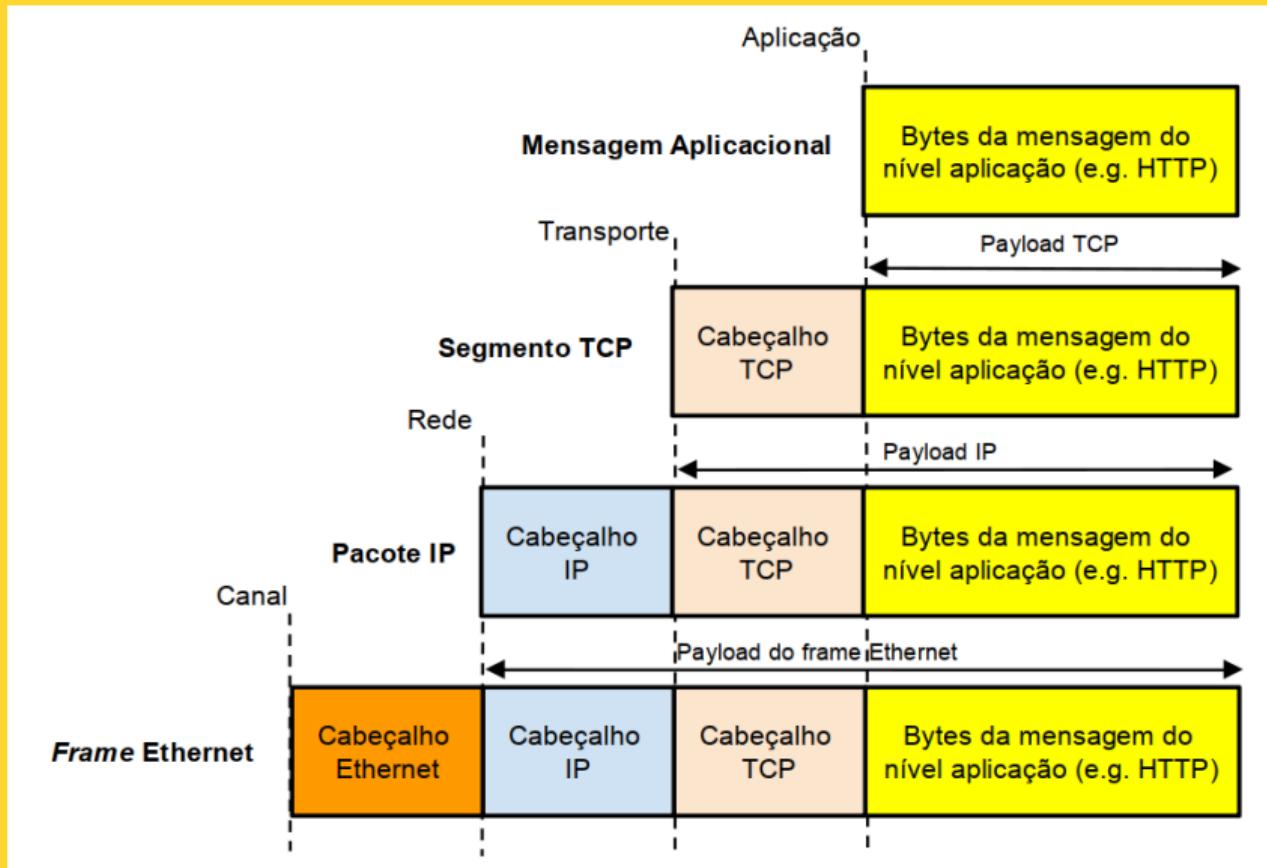
- Uma camada pode duplicar funcionalidades das inferiores
 - Tratamento de erros, retransmissões
- Algumas camadas necessitam de informação de outras
 - Timestamps, MTU
- Os cabeçalhos podem tornar-se maiores que os dados (*payload*)

ENCAPSULAMENTO

ENCAPSULAMENTO

- Uma mensagem pode ser vista em diferentes fases/níveis
- Em cada nível, há uma parte relevante (o cabeçalho) e uma parte opaca (o payload)
- Uma analogia possível é uma sequência de envelopes dentro de envelopes

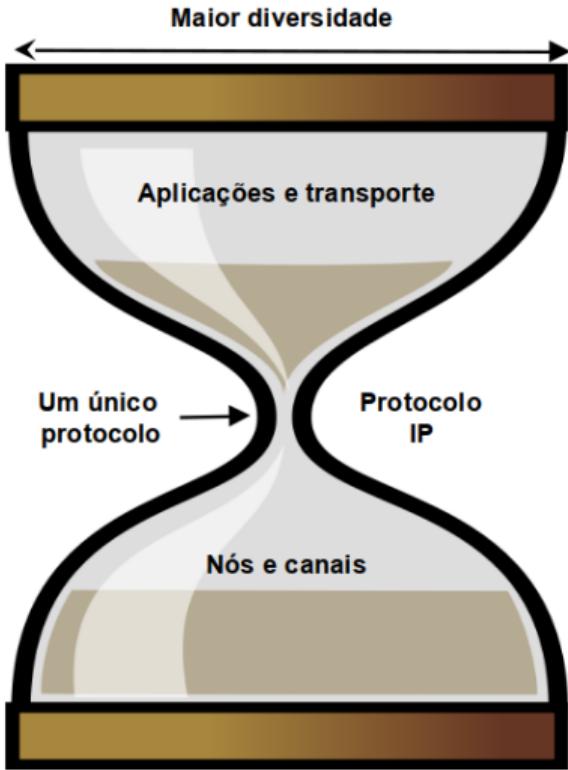
ENCAPSULAMENTO



BARALHANDO O "PUZZLE"

- Canais podem ser lógicos em vez de físicos
 - Pacotes IP encapsulados na parte dos dados de um pacote TCP ou UDP (e.g., VPNs)
 - Pacotes IP encapsulados em pacotes IP (virtual networks)
 - Pacotes TCP encapsulados em pacotes HTTP (alguns tipos de websockets)

MODELO DE PROTOCOLOS INTERNET

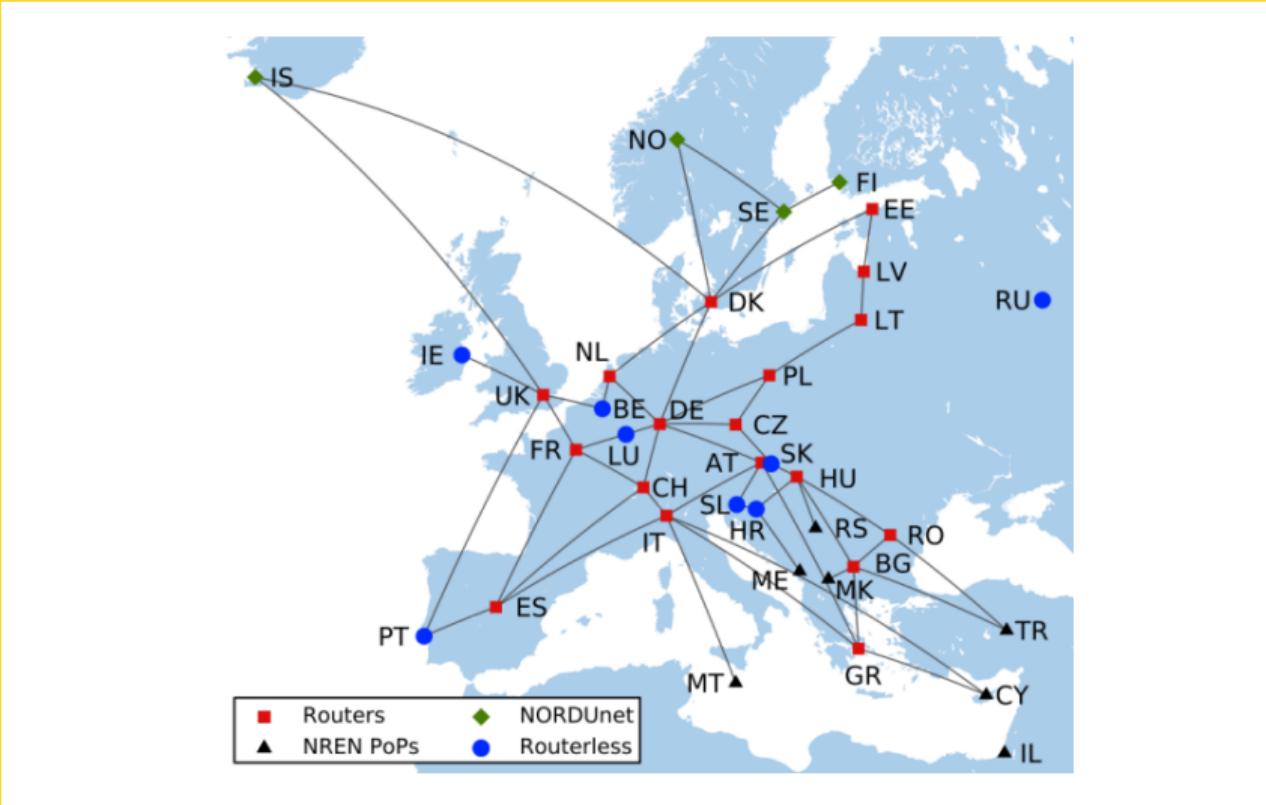


A INTERNET COMO REDE DE REDES

A INTERNET É UMA REDE DE REDES

- Os computadores estão ligados à Internet através de redes de acesso ISPs (Internet Service Providers)
- As redes de acesso têm de comunicar umas com as outras através de redes de trânsito de forma a que todos os computadores possam comunicar entre si
- Os ISPs de trânsito são nacionais, regionais ou mundiais
- A concorrência e a geografia introduz novos ISPs

EXEMPLO DE BACKBONE REGIONAL (GÉANT)



LIGAÇÕES INTERCONTINENTAIS

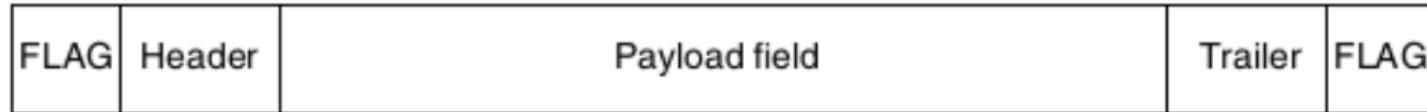


TRANSMISSÃO DIGITAL DE DADOS

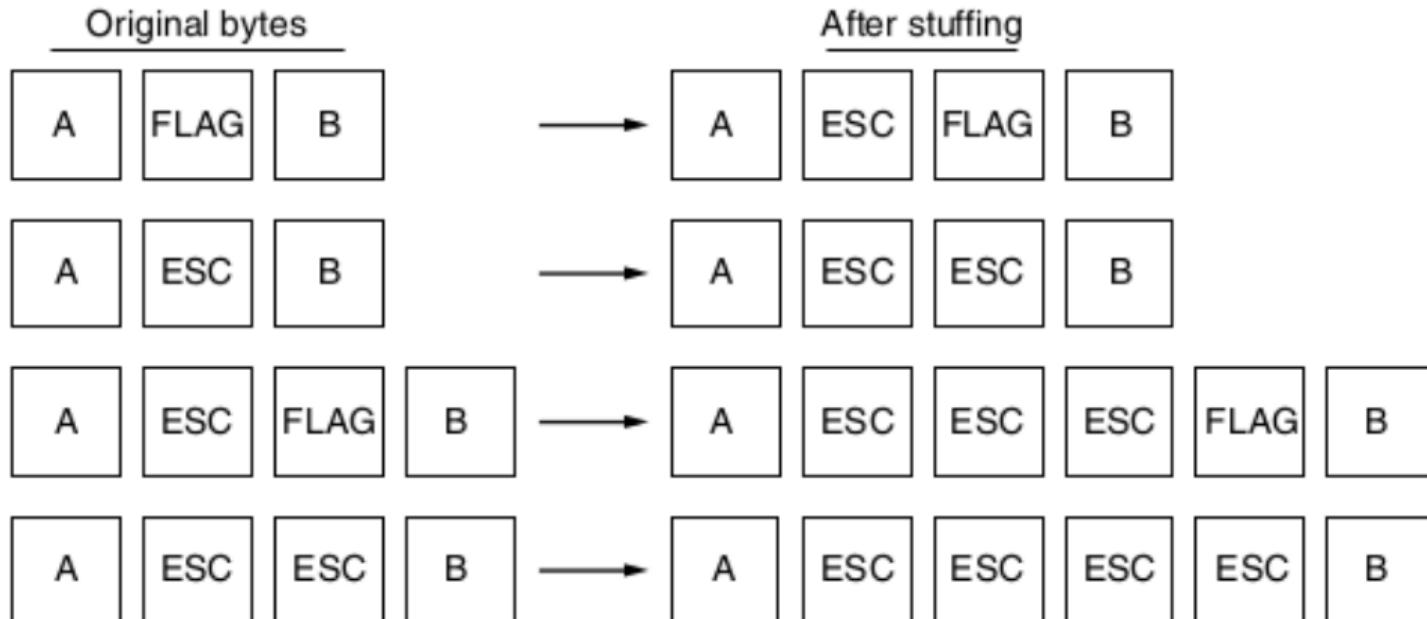
Framing (divisão dos dados em frames)

- Hipótese: espaço de tempo entre frames
- Risco de falhar a sincronização
- Hipótese: número de caracteres (indicado no header)
- Risco da contagem de caracteres chegar errada
- Hipótese: bytes (flag) no início e fim (com *stuffing*)
- Byte *escape* para quando a flag ocorre nos dados

BYTE STUFFING



(a)



FRAMING

- Problema: baseado em dados divididos em bytes (8 bits)
- Hipótese: flags de início e fim, com *bit stuffing*

EXEMPLO

- Flag é 0111110
- A camada física usa *bit stuffing*
- Sempre que ocorrem 5 uns (11111 é adicionado um zero)
- No receptor, sempre que ocorrem 5 uns e um zero, o zero é removido
- Se for a flag, não existe o zero

BIT STUFFING

(a) 0 1 1 0 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

↑
Stuffed bits

(c) 0 1 1 0 1 0 0 1 0

Figure 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

TRANSMISSÃO DE DADOS

CANAIS DE DADOS

- Asseguram transmissão de pacotes entre o emissor e o receptor
- O pacote passa por vários canais
 - de diferentes naturezas
 - suportados em diversas tecnologias
 - mas com primitivas básicas semelhantes
- Erros de transmissão (interferências, ruído)
- Capacidade (largura) limitada
- Velocidade limitada

CANAIS (LINKS) E NÓS DE COMUNICAÇÃO



TIPOS DE CANAIS DE DADOS

- Ponto-a-ponto (*point-to-point*)
 - Ligam 2 nós (cabos, wireless direccional)
- Multi-ponto (*multipoint*)
 - Ligam múltiplos nós (wifi, satélites, cabos com switch)
- *Guided media* e *Unguided media*
- *simplex*, *half-duplex* e *full-duplex*

TRANSMISSÃO DIGITAL DE DADOS

- Um pacote é uma sequência de bits
- Ao nível da comunicação física, utilizamos o termo *bit frame*
- Uma *frame* necessita de cabeçalhos específicos (delimitadores)
- A transmissão está sujeita a ruído e interferências diversas, pelo que requer alguma forma de detecção de erros

DÉBITO (BANDWIDTH)

Débito ou velocidade de transmissão

Quantidade de informação, medida em bits por segundo, que o canal é capaz de transmitir por unidade de tempo.

TEMPO DE TRANSMISSÃO

Tempo de transmissão

de uma frame com B bits através de um canal com um débito de D bits por segundo, i.e., o tempo desde que começa a ser emitido o primeiro bit até que acaba de ser emitido o último bit é B/D segundos.

$$\text{Tempo de transmissão} = \text{Dimensão da frame} / \text{Débito do canal}$$

TEOREMA DE NYQUIST (SEM RUÍDO) (1924)

Taxa máxima de transmissão

$$T_{max} = 2B \cdot \log_2(V) \text{ bits/s}$$

$B \rightarrow$ Bandwidth

$V \rightarrow$ Número de símbolos (2)

TEOREMA DE SHANNON (COM RUÍDO) (1948)

Taxa máxima de transmissão

$$T_{max} = B \cdot \log_2(1 + S/N) \text{ bits/s}$$

$B \rightarrow$ Bandwidth

$S/N \rightarrow$ Signal-to-noise ratio

ERROS DE TRANSMISSÃO

TAXA DE ERROS DE BITS

- Ocorrência de um erro num bit é um acontecimento independente de ocorrer outro bit (quase sempre)
- p é a probabilidade de um bit chegar errado (Bit Error Rate - BER)
- n é a dimensão da mensagem
 - A probabilidade da mensagem chegar sem erros é

$$(1 - p)^n$$

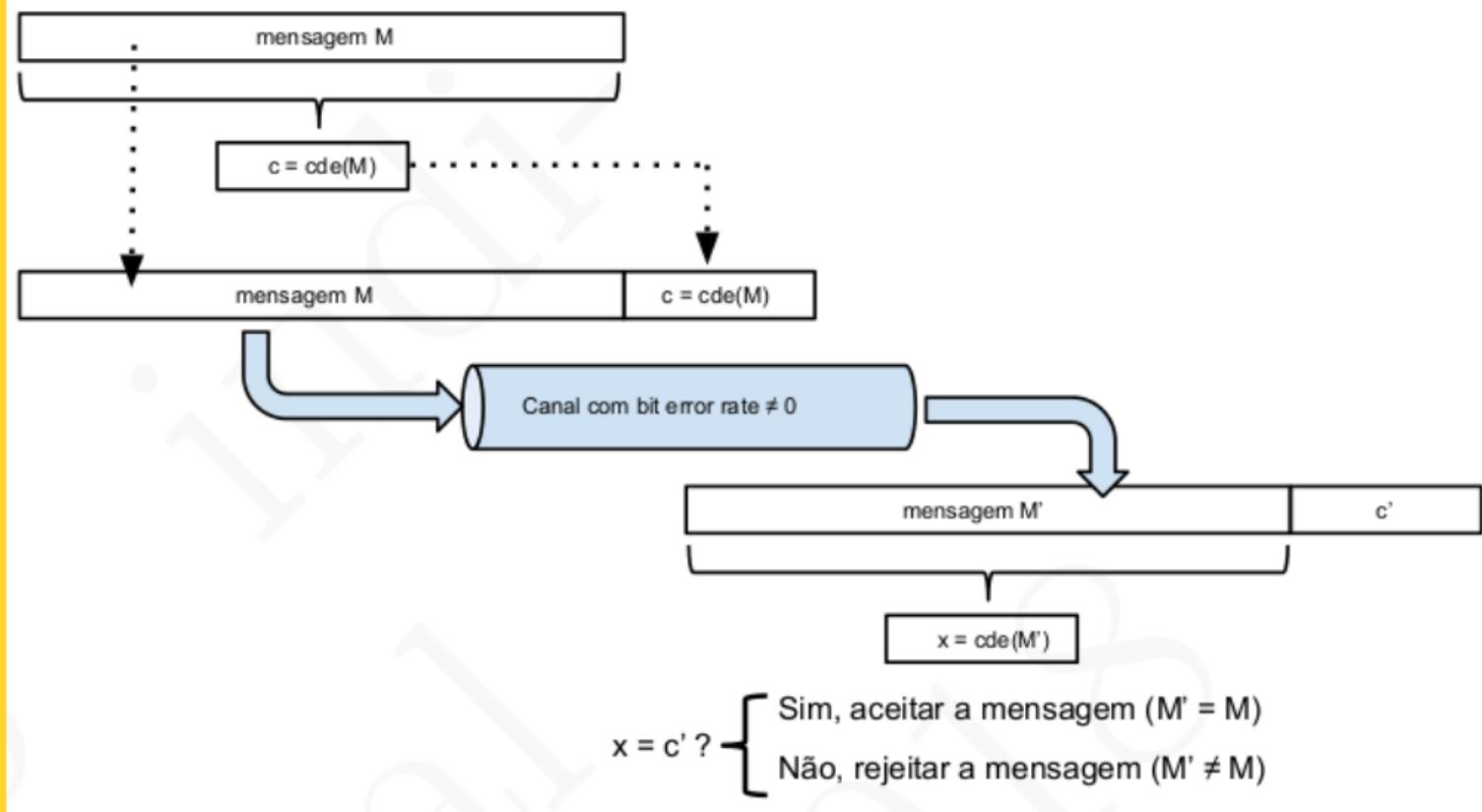
- A probabilidade da mensagem chegar com erros é

$$1 - (1 - p)^n$$

DETECÇÃO DE ERROS

- Função de detecção de erros ($cde(M)$)
- Pode ser a função identidade ($cde(M) = M$)
- Mas o objectivo é identificar o maior número de erros possível usando o menor número de bits possível
- E há erros que a função não detecta (quais???)

DETECÇÃO DE ERROS



TÉCNICAS DE DETECÇÃO DE ERROS

- Bits de paridade (usado nos modems)
- Soma de controlo (checksum)
 - Usado no IP para proteger o cabeçalho
 - Usado no TCP e UDP para proteger toda a mensagem (RFC 1071)
 - Custo de apenas 16 bits por pacote

TÉCNICAS DE DETECÇÃO DE ERROS (2)

Internet Checksum (java)

```
int checksum (byte[] data) {
    int sum = 0;
    int i = 0;
    for (;;) {
        sum = sum + byte[i]<<8 + byte[i+1];
        if ( sum & 0xFFFF000 > 0 ) { // a carry bit occurred
            sum &= 0xFFFF;
            sum++;
        }
        i += 2;
        // when finished return the ones complement of the sum
        if (i > data.length) return ~ (sum & 0xFFFF);
    }
}
```

TÉCNICAS DE DETECÇÃO DE ERROS (3)

- Bits de paridade (usado nos modems)
- Soma de controlo (checksum)
 - Usado no IP para proteger o cabeçalho
 - Usado no TCP e UDP para proteger toda a mensagem (RFC 1071)
 - Custo de apenas 16 bits por pacote
- Cyclic Redundancy Check (CRC)
 - Implementação banalizada em hardware
 - Usa a divisão módulo 2
 - Adequado para frames de comprimento significativo (> 1000 bytes)
 - Custo de apenas 32 bits por frame
 - Existem versões com menos bits (12, 16) para frames mais pequenas

CORRECÇÃO DE ERROS

- Permite não só detectar, como corrigir erros
- Usado actualmente em ligações sem fios
- Quando a taxa de erros é significativa

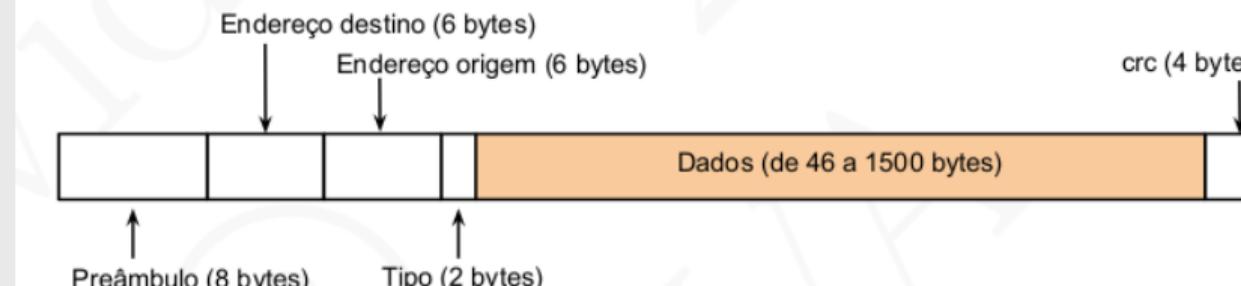
CASO PRÁTICO: FRAMES ETHERNET

FRAMES ETHERNET (IEEE 802.3)

- Preâmbulo com 8 bytes
- 7x 10101010
- 1x 10101011
- Permite sincronizar o relógio do receptor com o do emissor
- Normalmente o MTU (Maximum Transfer Unit) é de 1500 bytes
- Jumbo frames* têm 9000 bytes

FRAMES ETHERNET (IEEE 802.3)

Frame Ethernet



ENCAPSULAMENTO

