Addressing

Packet Format, Protocol Demultiplexing, IP classes and multicast, ARP

PEDRO MARTINS

April 11, 2018

Contents

1 Packet Format

1.1 Endereços

Um endereço é formado por 6 octetos [1], como se pode ver no diagrama da figura 1.

1º octeto	2º octeto	3º octeto	4º octeto	5º octeto	6º octeto
11011101	01110101	11001111	01011111	01000101	01111010

Figure 1: Exemplo de um endereço segundo o protocolo IEEE

No 1º octeto, existem dois bits com significados especiais

- **Último:** bit G/I (Grupo/Individual)
- **Penúltimo:** bit G/L (Global/Local)

O último bit do 1º octeto serve para identificar os tipos de endereços:

- `Unicast`: G/I = 0
- `Multicast`: G/I = 1
- `Broadcast`: todos os bits a 1

OUI: Organization Unique Identifier

1.2 Comum aos protocolos

- **Preamble:**
  - sequência alternada de '0's e '1's, para sincronização de clock
    - `01010101010101010101010...`
  - Utiliza-se **código de Manchester diferencial**
    - Produz exatamente a mesma sequência que os dados binários quando estes são uma sequência alternada de '0's e '1's
  - A **sincronização do clock** é crucial para decidir o **instante de amostragem**
  - Otimizar a escolha do instante de amostragem $\implies$ maximizar a abertura do diagrama de olho no instante de amostragem
  - O `preamble` possui 57 bits
    - No entanto, é preciso a indicação da terminação da trama, uma vez que estes bits apenas servem sincronismo, e "não podem ser contados antes de existir sincronismo"
- **SFD - Start of Frame Delimiter:**
  - 1 octeto

[1] octeto: conjunto de 8 bits. 1 byte.

  - Funcionalidade: permitir a detecção do início da `frame`
  - Pode existir *padding*
    * Para garantir a formatação correta do `frame` e alinhamento da informação
    * **Pad:** bytes de *padding*

- **Hardware Destination address**
  - 6 octetos (ver figura)
- **Source Address:**
  - 6 octetos (ver figura)
- **FCS - Frame Check Sequence:**
  - Permite detetar de erros na transmissão

1.3 Ethernet II

- Existem dois tipos de *standards* de Ethernet
- A proposta original foi submetida pelo IEEE

preamble	1 bytes SFD	6 bytes destination	6 bytes source	2 bytes protocol	46 - 1500 bytes data	4 bytes FCS	1 bytes EFD

Figure 2: Estrutura de um pacote de Ethernet

- **Protocol:**
  - 3º campo no header
  - superior a 1500 bytes
  - representa o protocolo à qual os dados pertencem.
- **EFD - End of frame Delimiter:**
  - Detetar o fim do `frame`
  - Possui um padrão específico
  - Utilizado porque não existe informação relativa ao tamanho do pacote na Ethernet II
- **Data:**
  - Dados a serem enviados
  - 46 a 1500 bytes de mensagem

1.4 IEEE 802.3

- **length:**
  - tamanho do pacote de dados (`MAC`)

preamble	1 bytes SFD	6 bytes destination	6 bytes source	2 bytes length	1 byte DSAP	1 byte SSAP	1 byte CTL	43 - 1497 bytes data	4 bytes FCS

Figure 3: Estrutura de um pacote de IEEE 802.3

- Os três próximos bits (`DSAP`, `SSAP` e `CTL`) referem-se à `LLC - Logical Link Control` *Protocol Layer*, e são usadas para representar o protocolo.
- **Data:**
  - Dados a serem enviados
  - 43 a 1497 bytes de mensagem

Uma das principais diferenças entre o protocolo Ethernet II e o protocolo IEEE 802.3 é que no IEEE 802.3 é feita explicitamente a identificação do protocolo. Entre o protocolo IEEE e Ethernet II existe uma identificação explicita na trama enviada. Além disso, o campo length (3º campo, possui dimensão inferior a 1500 bytes)

Contém ainda explicitamente:

- Designação do serviço de *access point*
- Quais são as "aplicações" da camada `Applications` que precisam do pacote
- *Control Data*
- *Frame Check Sequence*, com CRC (Cyclic Redundancy Check)

2 Protocol Demultiplexing

Usando o campo `protocol` de uma `frame` Ethernet, obtemos o diagrama de blocos representado abaixo, na figura 4

O `demultiplexing` é efetuado pelo `MAU` - *Media Access Unit*:

- Os pacotes são recebidos de um serviço e precisam de ser enviados para outro serviço
- Cada serviço possui um `grid number`
  - A camada 2 sabe a que entidade da camada 3 entregar o pacote
  - O protocolo, ao ser desmultiplexado, "revela" o endereço da entidade da camada 3

2.1 Classes de IP address

As classes IP servem para identificar os tipos de rede em relação ao seu tamanho

Inicialmente, no protocolo IEEE, 3 bytes são para o fabricante, 3 bytes para as placas de rede. Atualmente, são usados os 6 bytes para as redes.

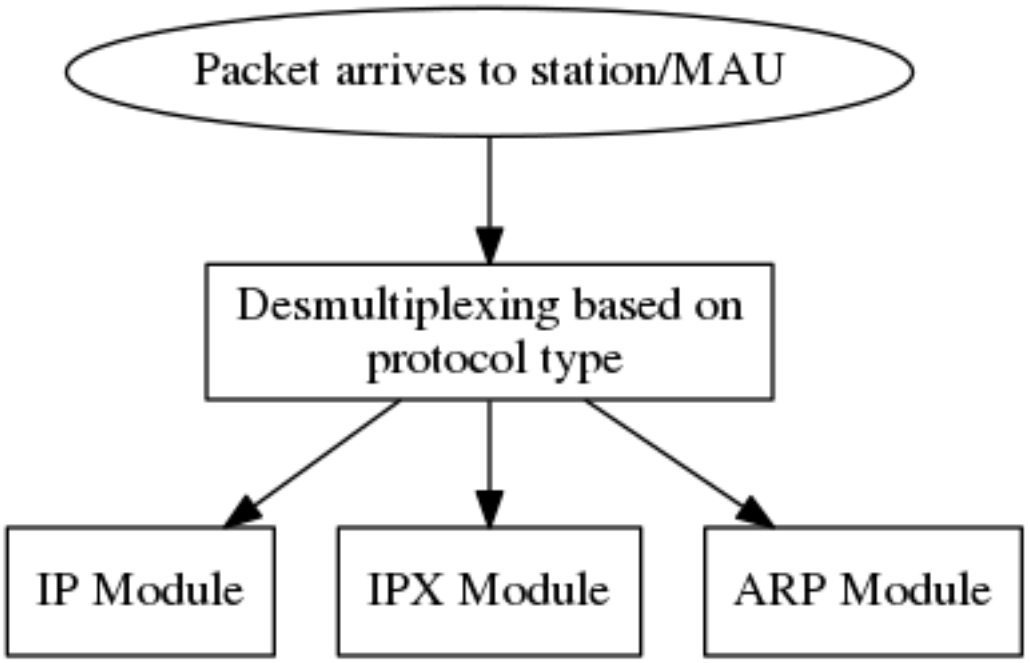

Figure 4: Diagrama de blocos para a operação de `protocol demultiplexing`. Na figura, MAU significa *Media Access Unit*

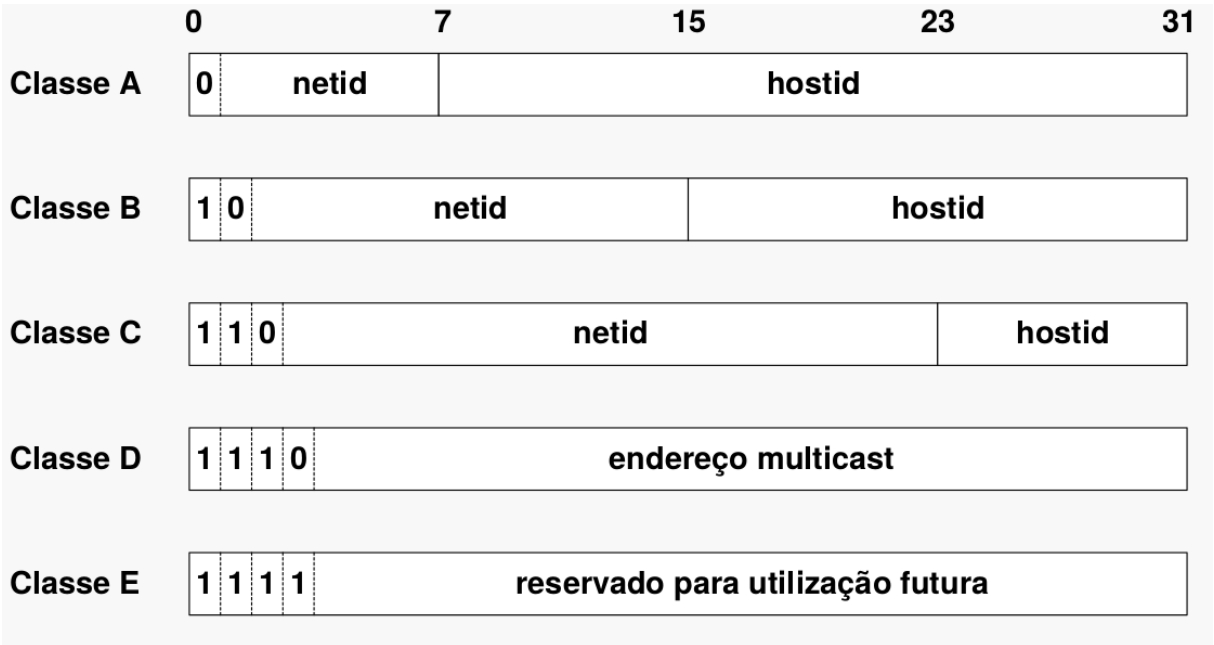

Figure 5: As diferentes classes de IP. A classe E não é usada atualmente

Table 1: Características dos 3 principais tipos de endereçamento usados. Note que nem todos os potenciais endereços são usados

Class	nº bits in prefix	nº max networks	nº bits in suffix	nº max hosts per network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Table 2: Organização dos bytes no endereço da classes de IP

Class	nº bytes network	nº bytes hosts
A	1	3
B	2	2
C	3	1
D	4	0

2.1.1 Classificação dos endereços nas classes

Class	Endereço mínimo possível	Endereço máximo possível
A	1.0.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.0.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.254

2.1.2 Problemas

As classes começaram a ser atribuídas nos primórdios da Internet. Isto significa que, por exemplo, a Boeing possua endereços classe A, e a China não sequer um endereço classe B.

2.1.3 Endereços IP especiais

- Um endereço todo a zeros identifica a rede atual

- Endereço todo a “1” é um broadcast local

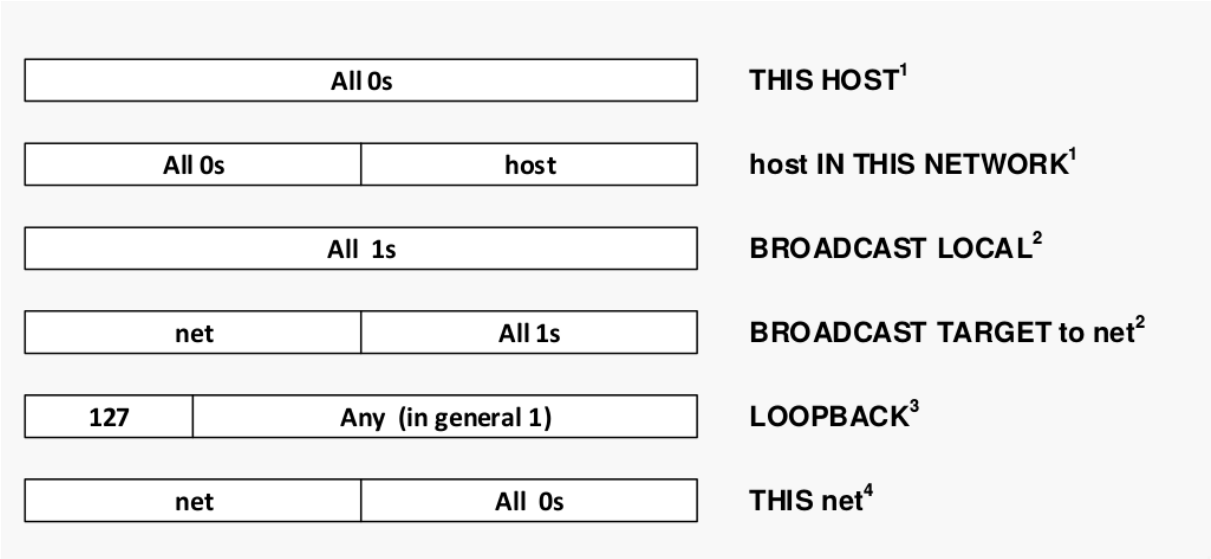

Figure 6: (1) - Apenas permitido na inicialização. Não representa um endereço válido e destino . (2) - Não é um endereço de origem válido. (3) Nunca deve aparecer na rede (No caso demonstrado, o LOOP BACK nunca deve sair para fora da placa de rede). O (4) indica um endereço usado para dar o nome à rede.

A razão porque não posso usar endereços “0” na rede é porque existe um programador que `hard-coded` o endereço “0” como sendo o endereço que identifica a máquina/host, para facilitar a escrita de um `mac-filter` . Desde aí, como algumas máquinas possuem este código, é preferível não arriscar a correr o risco de não conseguir comunicar com todas as máquinas

2.2 IP multicast

A classe D é uma classe usada para endereços `multicast`

`1110.<group ID>`

- Os pacotes são transmitidos a um grupo de máquinas,
- Cada máquina pode estar em mais do que um grupo em simultâneo
- É um tipo de endereçamento específico, que se comporta de forma diferente

IGMP: Internet Group Management Protocol

- Pode ser usado para efetuar a troca de informação entre os vários elementos/nós da rede
- Preferencialmente, devo ser usado `multicast` se o hardware tiver suporte para o mesmo. Caso contrário, é preferível usar `broadcast`

2.3 Máscaras de Rede

- As máscaras de rede são utilizadas para fazer `classless addresing`

	decimal		binário	
	rede	host	rede	hots
endereço IP	10.	0.0.1	00001010	00000000 00000000 00000001
máscara	255.	0.0.0	11111111	00000000 00000000 00000000

Table 4: Endereçamento *classless* e relação entre o endereço IP e a máscara da rede

- Inicialmente, os endereços IP serviam para **fixar e definir fronteiras** entre redes, usando os **primeiros bits do campo de endereço**, tal como no passado tinha sido feito para as classes A, B e C
- Mais tarde, as **fronteiras** entre redes passaram a ser **variáveis**
- Passou a ser usada uma máscara de rede para:
  - Definir o que pertence ou não à rede
  - Permite separar os **endereços** que pertencem à **rede** e os endereços que pertencem ao *host*
- É importante para definir aspetos como `broadcast` e `multicast`

2.4 Subnetting

O subnetting permite, entre outras coisas, organizar as redes em grupos, para *a posteriori* ser mais fácil agrupá-las e controlá-las em conjunto

3 ARP - Address Resolution Protocol

3.1 Porque é preciso?

Imaginemos a seguinte situação

> Sei o endereço de *hardware* e tenho pacotes de IP para entregar a um dado destinatário. Como é que mapeio um no outro, ou seja, como é que através do endereço IP do pacote recebido sei o *MAC address* para onde devo enviar?

Uma solução simples seria fazer o `broadcast` do pacote pela rede. Esta solução não é prática porque obriga a que:

- Todos os dispositivos na rede recebam o pacote
- Todos os dispositivos tenham de abrir o pacote
- Processá-lo
- Perceber se se destina ou não ao seu endereço IP
  - Se não, descartar o pacote
  - Se sim, continuar a processar o pacote

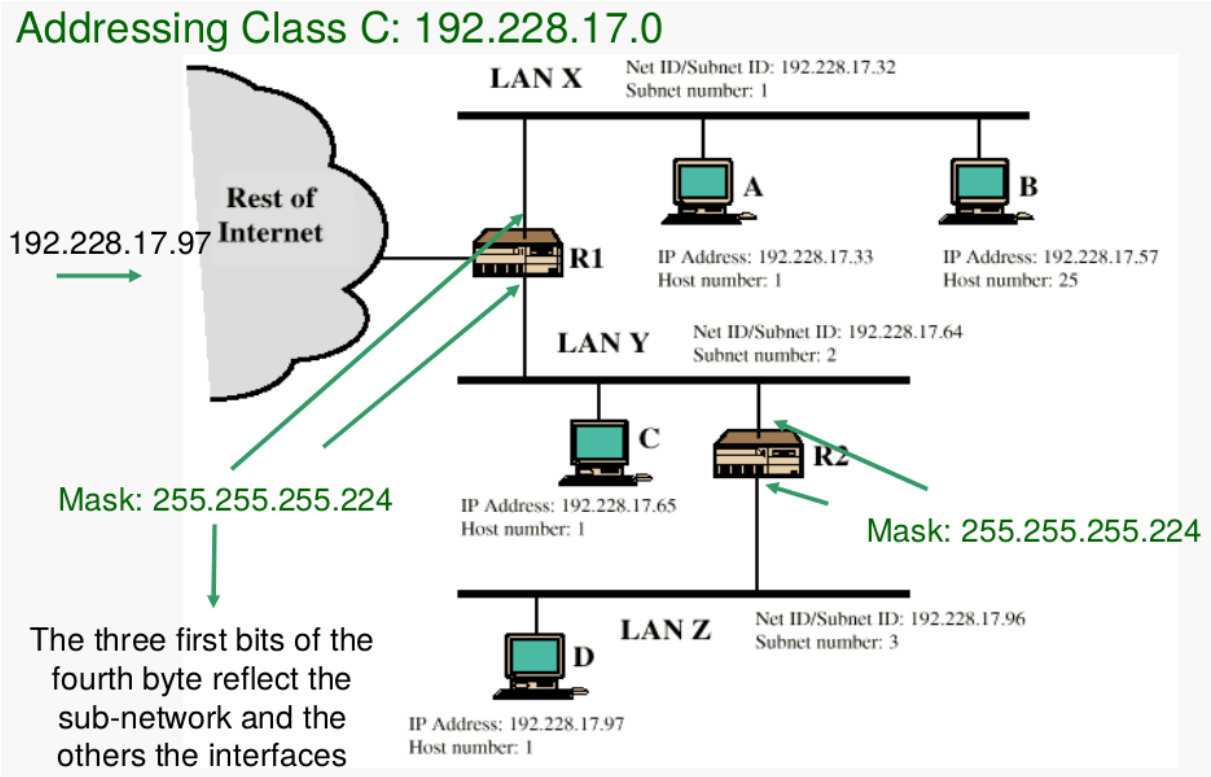

Figure 7: Exemplo de Subnetting

Esta metodologia apenas funciona para os `hubs`, porque estes apenas têm de efetuar o `broadcast` da informação. Não pode ser utilizada em dispositivos terminais, como computadores, porque obriga a que cada pacote da rede seja processado.

3.2 Solução

- `ARP`: *Address Resolution Protocol*
- É efetuado um pedido à rede, fazendo um `broadcast`, para saber quem sabe o `MAC address` de um dado endereço IP
  - **ARP Request**
- Se alguém na rede possuir na sua tabela de ARP, uma ligação entre o IP enviado no **ARP Request** e o `MAC address`:
  - envia uma **ARP Response** para o terminal/router que enviou o pedido, indicando o `MAC address` para o dado `IP`
- A estação que efetuou o `ARP Request` guarda a informação que recebeu na sua **tabela de ARP**

3.3 Objetivo do ARP:

- Descobrir se um terminal/router com um dado endereço de IP se encontra ligado na rede
- Permite a construção da frame de Ethernet com os endereços MAC de origem e destino corretos, usando a tabela de ARP

- Um **ARP Request** é sempre `broadcast`
- Uma **ARP Response não é** `broadcast`
- É identificado com o `Protocol Type 800`
- É inserido numa frame de `Ethernet`
- O `MAC address` representa o endereço físico
- O `IP address` representa o endereço lógico
- Sempre que existe uma comunicação entre duas máquinas, a tabela de ARP é atualizada

3.4 Problemas e Limitações

- Só pode ser usado em **redes locais**
  - Não é o mecanismo usado na Internet

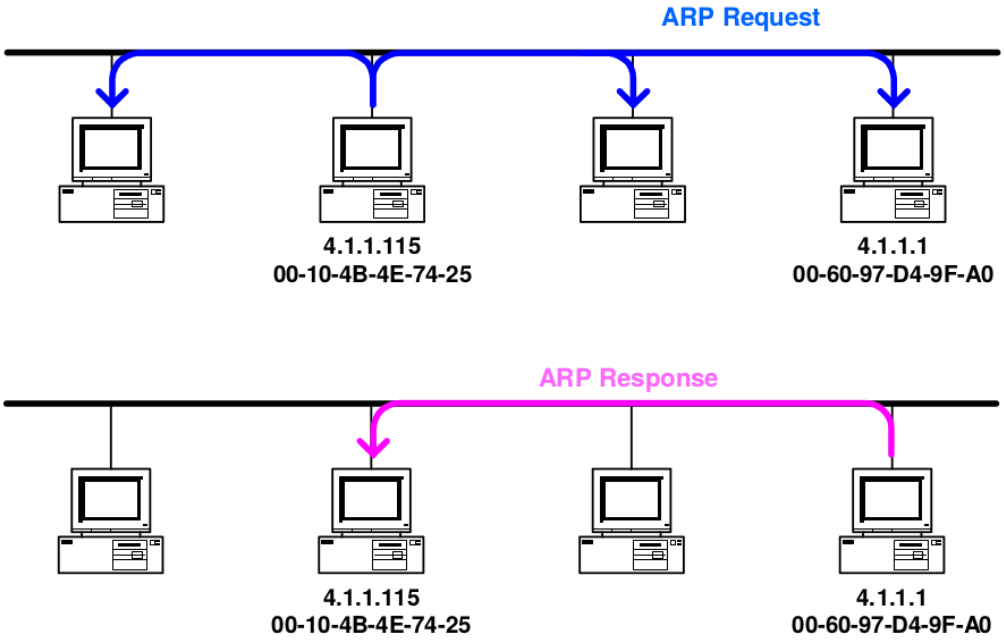

Figure 8: ARP Request and Response