# Redes de Computadores

Resumo

**Rafael Rodrigues**

LEIC
Instituto Superior Técnico
2022/2023

# Contents

# 1 Introduction

## 1.1 What is a Protocol?

A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.

- Hardware-implemented Protocol → Controls the flow of bits on the 'wire' between the 2 network cards.
- Congestion-Control Protocol → Controls the rate at which packets are transmitted between sender and receiver.
- Router Protocol → Determines a packet's path from source to destination.

## 1.2 Network Edge

## 1.3 Network Core

### 1.3.1 Packet Switching

- Comunicação dividida em pacotes, que partilham recursos com os restantes utilizadores.
- Cada um usa a largura de banda toda do canal. Recursos usados quando necessários.
- Não há alocação de canais.

### 1.3.2 Circuit Switching

- Reserva largura de banda em todo o percurso, somo se ligasse cabos fisicamente.
- Precisa de setup, não há partilha de recursos.
- Garantias de performance.

#### Packet Switching vs Circuit Switching

Packet Switching é excelente para comunicação bursty, cujas necessidade de recursos varia com o tempo.

## 1.4 Performance Metrics

## 1.5 Protocol Layers

To provide structure to the design of network protocols, network designers organize them in layers. Each layer provides its service by (1) performing certain actions within that layer and by (2) using the services of the layer directly below it.

Advantage → Layering provides a structured way to discuss system components.

Disadvantage $\rightarrow$ One layer may duplicate lower-layer functionality.

| |
|---|
| Application |
| Transport |
| Network |
| Link |
| Physical |

# 2 Application Layer

The application layer is where network applications and their application-layer protocols reside. The Internet's application layer includes many protocols, such as the HTTP protocol (which provides for Web document request and transfer), SMTP (which provides for the transfer of e-mail messages), and FTP (which provides for the transfer of files between two end systems). We'll see that certain network functions, such as the translation of human-friendly names for Internet end systems like www.ietf.org to a 32-bit network address, are also done with the help of a specific application-layer protocol, namely, the domain name system (DNS). We'll see in Chapter 2 that it is very easy to create and deploy our own new application-layer protocols.

An application-layer protocol is distributed over multiple end systems, with the application in one end system using the protocol to exchange packets of information with the application in another end system. We'll refer to this packet of information at the application layer as a message.

## 2.1 Principles of Network Applications

### 2.1.1 Network Application Architectures

- Client-Server Architecture:
  - One host (server) to many hosts (clients).
  - Clients do not directly communicate with each other.
  - The server has a fixed, well-known IP address, meaning a client can always contact the server.
- P2P Architecture:
  - Scalable
  - Hard to manage

### 2.1.2 Processes Communicating

## 2.2 The Web and HTTP (Port 80)

### 2.2.1 Overview of HTTP

### 2.2.2 Non-Persistent and Persistent Connections

### 2.2.3 HTTP Message Format

### 2.2.4 User-Client Interaction: Cookies

Cookies permitem que o protocolo HTTP mantenha estado. São guardadas pelo utilizador.
O servidor envia cookies (cookie header) na resposta HTTP.

### 2.2.5 Web Caching

**The Conditional GET**

Quando é solicitado um objeto que o browser já tem guardado em cache, o browser envia um GET condicional ao servidor, para verificar que a cópia guardada em cache está atualizada.

O servidor responde, mas caso não tenha sido modificado, o objeto não é enviado de modo a poupar bandwidth e melhorar o desempenho.

## 2.3 Electronic Mail in the Internet [Port 25]

### 2.3.1 SMTP

### 2.3.2 Comparison with HTTP

### 2.3.3 Mail Message Formats

### 2.3.4 Mail Access Protocols

## 2.4 Domain Name System (DNS) [Port 53]

- Associa nomes **(domínios)** a endereços IP.
- Base de dados distribuída por uma hierarquia de servidores de nomes **(name servers)**.
- Acedida através de um **resolver**.
- Utiliza maioritariamente o protocolo UDP, é usado TCP se a resposta for demasiado grande.

### 2.4.1 DNS Records

Resource Records (RR): (**name**, **value**, **type**, ttl)

The meaning of **name** and **value** depend on **type**:

- **type** = A:
  - **name**: hostname
  - **value**: IP address of the hostname
- **type** = NS:
  - **name**: domain
  - **value**: hostname of an authoritative DNS server knowing how to obtain IP address for the domain
- **type** = CNAME:
  - **name**: alias hostname
  - **value**: canonical hostname (real name)
- **type** = MX:
  - **name**: alias hostname of a **mail server**
  - **value**: canonical hostname (real name)

## 2.5 File Distribution

Distribution Time → Time it takes to get a copy of a file with size $F$ to all $N$ peers.

### 2.5.1 Client-Server Architecture

In the client-server architecture, none of the peers aid in distributing the file:

- The server must transmit one copy of the file to each of the $N$ peers. Thus, the server must transmit $N \cdot F$ bits. Since the server's upload rate is $u_s$, the time to distribute the time must be at least $\frac{N \cdot F}{u_s}$
- Let $d_{min}$ be the smallest download rate among the $N$ peers. The peer with the $d_{min}$ cannot obtain all $F$ bits of the file in less than $\frac{F}{d_{min}}$ seconds. Thus, this is the distribution time.

$$D_{client-server} = max\left\{\frac{N \cdot F}{u_s}, \frac{F}{d_{min}}\right\}$$

### 2.5.2 P2P Architecture

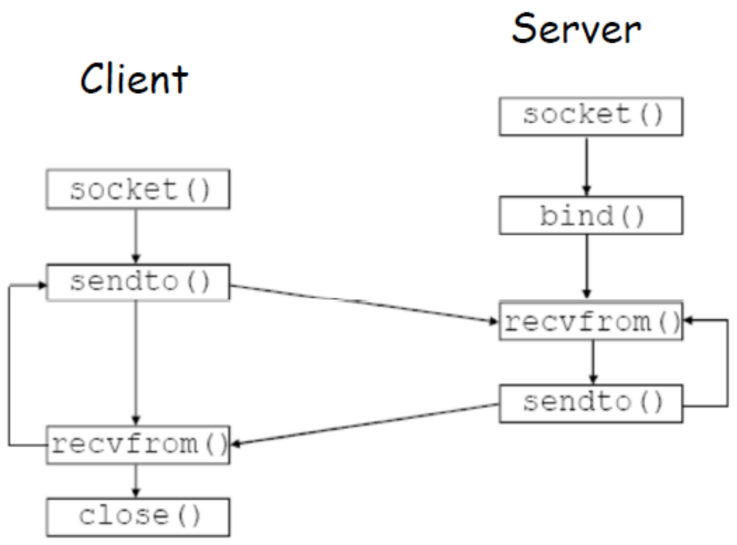
In the P2P architecture, when a peer receives some file data, it can use its own upload capacity to redistribute the data to other peers:

- The server must transmit $F$ bits of the file at least once into its access link. Since the server's upload rate is $u_s$, the minimum distribution time is at least $\frac{N \cdot F}{u_s}$
- Let $d_{min}$ be the smallest download rate among the $N$ peers. The peer with the $d_{min}$ cannot obtain all $F$ bits of the file in less than $\frac{F}{d_{min}}$ seconds. Thus, this is the distribution time.
- The total upload capacity of the system as a whole is equal to $u_{total} = u_s + u_1 + u_2 + ... + u_N$. The system must upload $F$ bits to each of the $N$ peers, thus delivering a total of $N \cdot F$ bits. This cannot be done at a rate faster than $u_{total}$. Hence, the minimum distribution time is also at least $\frac{N \cdot F}{u_{total}}$

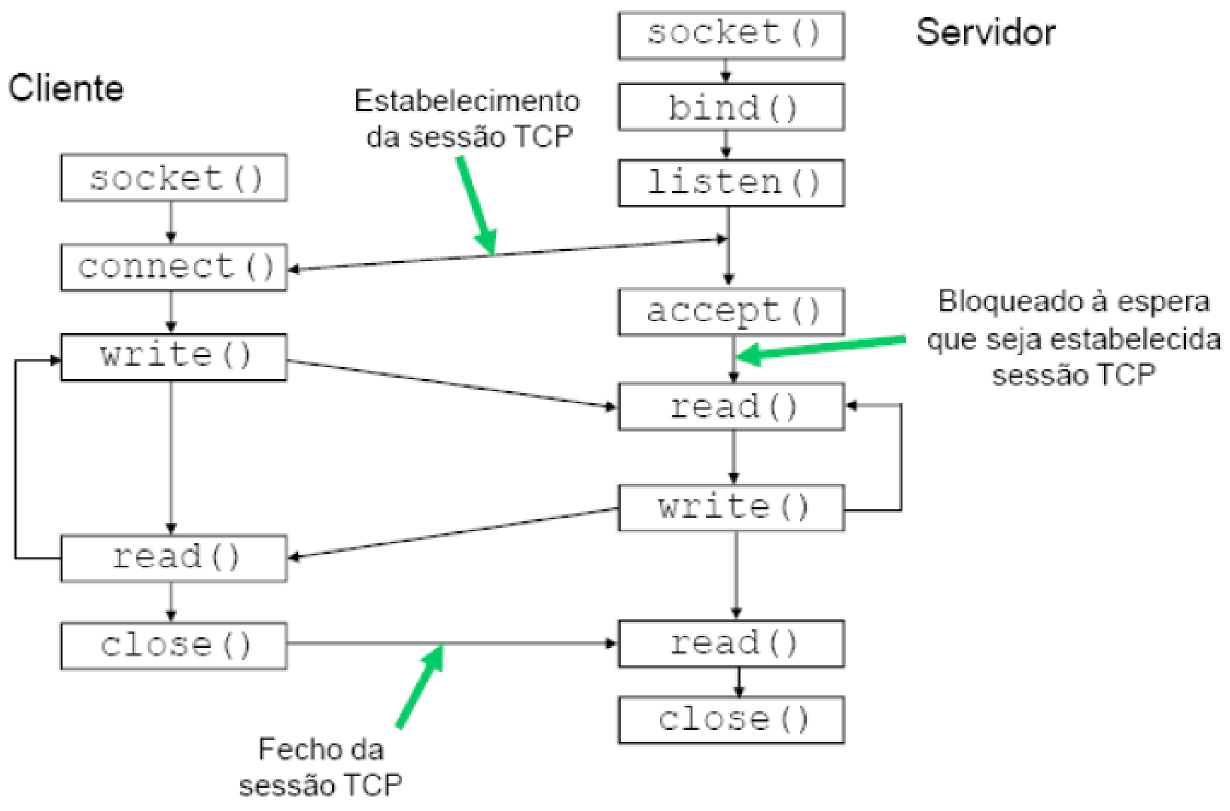
$$D_{client-server} = max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{N \cdot F}{u_{total}}\right\}$$

## 2.6 Socket Programming

**UDP**



**TCP**



**TCP vs UDP**

| TCP | UDP |
|---|---|
| read() and write() | sendto() and recvfrom() |
| Byte stream (no byte is lost) | Messages may be lost |
| Bytes read with read() may correspond to several write() | Preserves boundary between messages |
| Bytes written with write() may need to be read with several read() | Each message read with recvfrom() correspond to a single sendto() |

## 3 Transport Layer

The Internet's transport layer transports application-layer messages between application endpoints. In the Internet there are two transport protocols, TCP and UDP, either of which can transport application layer messages. TCP provides a connection-oriented service to its applications. This service includes guaranteed delivery of application-layer messages to the destination and flow control (that is, sender/receiver speed matching). TCP also breaks long messages into shorter segments and provides a congestion-control mechanism, so that a source throttles its transmission rate when the network is congested. The UDP protocol provides a connectionless service to its applications. This is a no-frills service that provides no reliability, no flow control, and no congestion control. In this book, we'll refer to a transport-layer packet as a segment.

### 3.1 User Datagram Protocol (UDP)

- Um socket UDP é identificado pelo tuplo: (dest IP, dest port)
- Quando recebe datagrama UDP, entrega ao socket com esse IP:porto
- Sem ligações nem garantias de ordem ou entrega
- Sem controlo de congestão
- Checksum (opcional):

Headers pequenos, é simples e rápido. Usado para streaming, DNS e SNMP.

### 3.2 Transmission Control Protocol (TCP)

- Um socket TCP é identificado pelo tuplo: (src IP, src port, dest IP, dest port)
- Há 1 socket por ligação (cliente), cada uma abre um socket novo.
- Entrega fiável e ordenada.
-

# 4 Network Layer

The Internet's network layer is responsible for moving network-layer packets known as datagrams from one host to another. The Internet transport-layer protocol (TCP or UDP) in a source host passes a transport-layer segment and a destination address to the network layer, just as you would give the postal service a letter with a destination address. The network layer then provides the service of delivering the segment to the transport layer in the destination host.

The Internet's network layer includes the celebrated IP protocol, which defines the fields in the datagram as well as how the end systems and routers act on these fields. There is only one IP protocol, and all Internet components that have a network layer must run the IP protocol. The Internet's network layer also contains routing protocols that determine the routes that datagrams take between sources and destinations. The Internet has many routing protocols. As we saw in Section 1.3, the Internet is a network of networks, and within a network, the network administrator can run any routing protocol desired. Although the network layer contains both the IP protocol and numerous routing protocols, it is often simply referred to as the IP layer, reflecting the fact that IP is the glue that binds the Internet together.

## 4.1 Overview of Network Layer

The primary role of the **data plane** is to forward datagrams from its input links to its output links.

The primary role of the **control plane** is to coordinate these local, per-route forwarding actions so that datagrams are ultimately transferred end-to-end, along paths of routers between source and destination.

### 4.1.1 Forwarding and Routing: The Data and Control Planes

The primary role of the network layer is to move packets from the sending host to a receiving host. Thus, there are 2 important network-layer function:

1. Forwarding → When a packet arrives at a router's input link, it must move the packet to the appropriate output link. A packet might also be blocked from exiting a router or duplicated and sent over multiple outgoing links.
   **Forwarding is router-local**, and is the key function performed by the data-plane functionality.

2. Routing → The network layer must determine the path taken by packets as they flow from a sender to a receiver. To do so, the network layer uses multiple routing algorithms. Routing refers to the network-wide. **Routing is a network-wide process**.

## 4.2

## 4.3 The Internet Protocol (IP)

### 4.3.1 Fragmentation

### 4.3.2 Adressing

**Dynamic Host Configuration Protocol (DHCP) [Server Port 67, Client Port 68]**

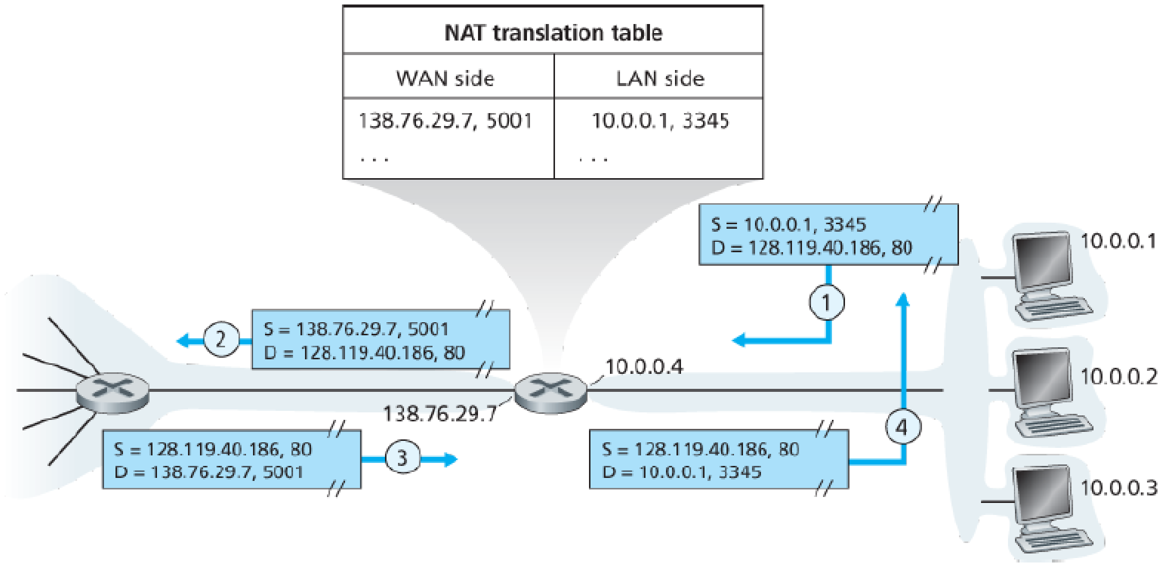
- Mensagens de DHCP são em UDP.
- Endereços são emprestados por um tempo finito (lease time).
- Após 50% do lease time, o cliente tenta renovar o IP (**DHCP Request**).
  Se não conseguir, tenta de novo a 85%.

Outras mensagens:

- Decline: cliente rejeita oferta
- NACK: servidor não consegue satisfazer pedido
- Release: cliente informa que já não quer usar endereço
- Inform: cliente pede parâmetros extra

**Network Address Translation (NAT)**

1. Quando um utilizador da rede privada envia um pacote, o NAT substitui o endereço IP interno do remetente pelo externo, e memoriza o correspondente.
2. Quando recebe uma resposta o NAT restaura o IP interno.



## 4.4 Software Defined Network (SDN)

A abordagem usando SDN para a implementação da camada de rede considera que apenas o plano de dados (**Data Plane**) é implementado em cada encaminhador.

# 5 Link Layer

The Internet's network layer routes a datagram through a series of routers between the source and destination. To move a packet from one node (host or router) to the next node in the route, the network layer relies on the services of the link layer. In particular, at each node, the network layer passes the datagram down to the link layer, which delivers the datagram to the next node along the route. At this next node, the link layer passes the datagram up to the network layer.

The services provided by the link layer depend on the specific link-layer protocol that is employed over the link. For example, some link-layer protocols provide reliable delivery, from transmitting node, over one link, to receiving node. Note that this reliable delivery service is different from the reliable delivery service of TCP, which provides reliable delivery from one end system to another. Examples of link-layer protocols include Ethernet, WiFi, and the cable access network's DOCSIS protocol. As datagrams typically need to traverse several links to travel from source to destination, a datagram may be handled by different link-layer protocols at different links along its route. For example, a datagram may be handled by Ethernet on one link and by PPP on the next link. The network layer will receive a different service from each of the different link-layer protocols. In this book, we'll refer to the link-layer packets as frames.

## 5.1

## 5.2 Error Detection and Correction Techniques

### 5.2.1 Parity Checks

### 5.2.2 Cyclic Redundancy Check (CRC)

## 5.3 Multiple Access Links and Protocols

### 5.3.1 Channel Partitioning Protocols

### 5.3.2 Random Access Protocols

### 5.3.3 Dynamic Allocation Protocols (Taking-Turns)

## Summary of MAC Protocols

| | Dynamic Allocation | Random Access |
|---|---|---|
| High Loads | Efficient | Collision overhead |
| Low Loads | Active nodes have to wait for idle nodes | Efficient |

## 5.4 Switched Local Area Networks

### 5.4.1 Link-Layer Addressing and ARP

#### MAC Addresses

#### Address Resolution Protocol (ARP)

### 5.4.2 Ethernet

### 5.4.3 Link-Layer Switches

#### Self-Learning

Switches are self-learning, meaning the switch table is built automatically, dynamically and autonomously.

1. The switch table is initially empty.
2. For each incoming frame received on an interface, the switch stores in its table:

| MAC Address | Interface | Current time |
|---|---|---|
| E6-E9-00-17-BB-4B | 1 | 00:01 |

3. The switch deletes an address in the table if no frames are received with that address as the source address after some time (**aging time**).

#### Spanning-Tree Protocol (STP)

- Establishes a root node called the root bridge.
- Constructs a topology that has one path from/to every network node.
- Resulting tree originates from the root bridge.
- Redundant links that are not part of the shortest path are blocked.

#### Bridge Protocol Data Unit (BPDU)

Convention for Conf-BPDUs: Root ID . Root Path Cost . Bridge ID

A configuration $C_1$ is said to be better than $C_2$ if (ordered by decreasing importance):

1. $C_1$ Root ID is lower than that of $C_2$.
2. $C_1$ Root Path Cost is lower than that of $C_2$.
3. $C_1$ Bridge ID is lower than that of $C_2$.
4. $C_1$ Port ID is lower than that of $C_2$.

### 5.4.4 Virtual Local Area Networks (VLANs)

# 6 Wireless and Mobile Networks

## 6.1 WiFi [802.11]

### 6.1.1 Architecture

### 6.1.2 MAC Protocol

**Multiple Access**

**Collision Avoidance**