Computer Networks - Xarxes de Computadors

Teacher: Llorenç Cerdà

Slides: http://studies.ac.upc.edu/FIB/grau/XC

Outline

- Course Syllabus
- Unit 1: Introduction
- Unit 2. IP Networks
- Unit 3. TCP
- Unit 4. LANs
- Unit 5. Network applications
Course Syllabus

Course Organization

- 2+1h lectures/week: theoretical + problems
  - Print the problems manual (available in the racó).
  - Try to do the problems beforehand.
  - Find textbooks and related links at the web page.
- Laboratory sessions of 2h on selected weeks + 2 non classroom labs
  - Buy laboratory manual in Repography.
  - Study and prepare sessions before hand.
  - Minicontrol held at the end of each session.
  - Required submitting a report at the beginning of the session. Otherwise, the minicontrol cannot be done.

web page: http://studies.ac.upc.edu/FIB/grau/XC
Course Syllabus

Evaluation:

\[ NF = 0.25 \times NL + 0.75 \times NT \]

Where:
- NF = Final grade.
- NL = Laboratory: 25% average of lab sessions, and 75% a final lab. exam.
- NT = Theory grade.

NT can be obtained:
- Continuous evaluation: \( NC = 0.4 \times C1 + 0.4 \times C2 + 0.2 \times C3 \). If \( NC \geq 5 \) then \( NT = NC \).
  - \( C1 \): Units 1,2 (introduction+IP),
  - \( C2 \): Unit 3 (TCP+LANs),
  - \( C3 \): Units 4,5 (Apps)
- Final Exam (EF). \( NT = \max(NC, EF) \).
  - If with NC it is \( NF \geq 5 \), you must send an email to the coordinator if you want to do the EF (to increase grade).
Course Syllabus

Incentive to study:
The final grade (NF) will be increased 1 point to students who meet the following conditions:

- Deliver on time the tracking problems (exercicis de seguiment) that will be proposed during the course.
- Obtain a grade $\geq 5$ at least 1 of the Controls.
- Have a theory grade (NT) and lab (NL) greater than or equal to 5: NT,NL $\geq 5$.

Autonomous learning (transversal competence):
- Two non classroom labs (home labs) will be proposed in the Racó.
- Evaluated with a specific final lab exam.
Computer Networks - Xarxes de Computadors

Outline

- Course Syllabus
- Unit 1: Introduction
- Unit 2. IP Networks
- Unit 3. TCP
- Unit 4. LANs
- Unit 5. Network applications
Unit 1: Introduction

Outline

- Brief history of Computer Networks and Internet
- Introduction to the Internet
- Standardization Organizations and OSI Reference Model
- Client-Server Paradigm
Unit 1: Introduction

Brief history of Computer Networks

- 1830: Telegraph
- 1866: First transatlantic telegraph cable
- 1875: Alexander Graham Bell invented the telephone
- 1951: First commercial computer
- 1960: Concept of Packet Switching.
- 1960s: ARPANET project, origins of the Internet.
- 1990s: The Internet is opened to the general public.

Source: wikipedia
Unit 1: Introduction

Brief History of the Internet


- ARPANET connected Universities, research labs and military centers. Military portion separated in 1983.

- 1970s: End-to-end reliability was moved to hosts, developing TCP/IP. TCP/IP was ported to UNIX Berkeley distribution, BSD.

- 1990s: The Internet is opened to commerce and the general public by the Internet Service Providers, ISP.
Unit 1: Introduction

Outline

- Brief history of Computer Networks and Internet
- Introduction to the Internet
- Standardization Organizations and OSI Reference Model
- Client-Server Paradigm
Unit 1: Introduction

- Host
- Access Network
- LAN
- WAN
- Telephone company, telco, or carrier.
- Router
- Line Bitrate
- Bits per second, bps.
Unit 1: Introduction

Bitrate

$t_b$ is the transmission time of 1 bit.

- $v_t = 1/t_b$ is the line bitrate in bits per second (bps)
- typical bitrate prefixes:
  - k, kilo: $10^3$
  - M, Mega: $10^6$
  - G, Giga: $10^9$
  - T, Tera: $10^{12}$
  - P, Peta: $10^{15}$
- Examples:
  - Public Switched Telephone Network (PSTN) modem: 56 kbps
  - ADSL: 4 Mbps
  - LAN Ethernet: 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps.
  - Carrier lines E3: 34 Mbps, OC-192: 9,9 Gpbs, ...
Unit 1: Introduction

Types of Switching

- **Circuit switching**, e.g. PSTN (Public Switched Telephone Network)
- **Packet switching**:  
  - Virtual Circuit, e.g. X.25, ATM.  
  - Datagram: Internet.

Datagram packet switching
Unit 1: Introduction

Outline

- Brief history of Computer Networks and Internet
- Introduction to the Internet
- Standardization Organizations and OSI Reference Model
- Client-Server Paradigm
Unit 1: Introduction

Standardization Bodies

# Unit 1: Introduction

## ISO Open Systems Interconnection (OSI) Reference Model

- **Layers or Levels**: Physical or Layer 1 (L1), ...
- Peer layers communicate using a *protocol*.
- Protocols from different layers are independent.
- Layer *i* offers services (e.g. send a datagram to a given address) to layer *i+1*: *Service Access Points (SAP)*.
- Peer layers exchange *Protocol Data Unit (PDU)*, which consists of a *header* and *payload*.

### Brief description of Layers:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Application</td>
<td>Processes using network services (web, email...)</td>
</tr>
<tr>
<td>6. Presentation</td>
<td>Encoding of text, numbers...</td>
</tr>
<tr>
<td>5. Session</td>
<td>“Login” type service.</td>
</tr>
<tr>
<td>4. Transport</td>
<td>End to end data transfer.</td>
</tr>
<tr>
<td>2. Data link</td>
<td>Structured transport of bits.</td>
</tr>
<tr>
<td>1. Physical</td>
<td>Electric and mechanical.</td>
</tr>
</tbody>
</table>

*Internet jargon: Layer 8: the user.*
Unit 1: Introduction

TCP/IP Architecture

- No RFC specifies the TCP/IP model.
- Networking literature usually identifies the layer model:

Physical network (Internet jargon): Any network that transport datagrams (not the OSI physical layer!)
Unit 1: Introduction

Encapsulation

- Each layer adds/remove the PDU header.
Unit 1: Introduction

TCP/IP Implementation

- **TCP/IP networking code** is part of the Operating System kernel.
- **Socket interface**: Is the Unix networking interface for the processes. It was first implemented in Berkeley Software Distribution, BSD.
- The **socket system call** creates a **socket descriptor** used to store all information associated with a network connection, similarly as an inode descriptor for a file.
Unit 1: Introduction

Outline

- Brief history of Computer Networks and Internet
- Introduction to the Internet
- Standardization Organizations and OSI Reference Model
- Client-Server Paradigm
Unit 1: Introduction

Client Server Paradigm: Processes, messages, sockets, segments and IP datagrams

User space
Process
message to send
(e.g. request a web page)

Transport layer
Socket interface
Segments encapsulated into IP datagrams

Operating System

Socket Interface:
TCP/IP API: socket(), bind(), listen(), accept(), connect(), read(), write()...

User space
Process
message to send
(e.g. a web page)

Transport layer
Socket interface
Segments encapsulated into IP datagrams

Operating System

Internet

client

server

ISP

ISP

Llorenç Cerdà-Alabern
Unit 1: Introduction

Client Server Paradigm: The Internet Transport Layer

- Two protocols are used at the TCP/IP transport layer: User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).
- UDP offers a datagram service (non reliable). It is connectionless.
- TCP offers a reliable service (correct segments are acknowledged, ack, lost segments are retransmitted). It is connection oriented (covered in detail in Unit 3).

TCP connection:

![TCP connection diagram]
Unit 1: Introduction

Client Server Paradigm

- How connection is established among processes?
- The client always initiates the connection towards a known IP address, in the IP header, and a *well known port* (< 1024), in the TCP/UDP header.
- Well known ports are standardized by IANA in RFC-1700 (*Assigned Numbers*). In a unix machine can be found in /etc/services.
- The server is a *daemon* waiting for client requests.

![Diagram of Client Server Paradigm](image)

**Well known port**

<1024

**Ephemeral port**

(≥1024)

**TCP/UDP**

**Socket interface**

**Processes**

**Operating System**

**Host A**

Port x1

**Host B**

Port y1

**TCP/UDP header**

**dst port = y1**

**src port = x1**

**dst port = x1**

**src port = y1**

**Server**

**Client**
### Unit 1: Introduction

**Client Server Paradigm – UNIX /etc/services File**

- Enables server and client programs to convert service names to well known ports.

```bash
linux> cat /etc/services
# Network services, Internet style
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# This list could be found on:
#                  http://www.iana.org/assignments/port-numbers
# **************************************************************************
# WELL KNOWN PORT NUMBERS
# The Well Known Ports are assigned by the IANA and on most systems can
# only be used by system (or root) processes or by programs executed by
# privileged users.
#
#  # Keyword  Decimal  Description
#    ------  -------  -----------
echo     7/tcp  Echo
#          7/udp  Echo
discard  9/tcp  # Discard
#          9/udp  # Discard
daytime  13/tcp # Daytime (RFC 867)
#          13/udp # Daytime (RFC 867)
chargen  19/tcp # Character Generator
#          19/udp # Character Generator
ftp-data 20/tcp # File Transfer [Default Data]
#          20/udp # File Transfer [Default Data]
ftp      21/tcp # File Transfer [Control]
ssh      22/tcp # SSH Remote Login Protocol
#          22/udp # SSH Remote Login Protocol
telnet   23/tcp # Telnet
#          23/udp # Telnet
...
Unit 1: Introduction

Client Server Paradigm – Network applications

- Remote commands
  - telnet
  - ssh
- Exchange of documents
  - ftp, sftp
  - peer-to-peer
- Web based applications
- Email
- Network management
- Real time
  - Voice over IP
  - Video streaming
- ...

Grau en enginyeria informàtica - Xarxes de Computadors (XC-grau)

Llorenç Cerdà-Alabern