

# Ayudantía 10

## Networking

Antonio Ossa    Vicente Domínguez

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile

IIC2233, 2015-1

# Tabla de contenidos

- 1 Introducción
  - IP
  - Puertos
  - Protocolos
- 2 Networking en Python
  - Sockets

# Introducción

- En el desarrollo de software cada vez es más necesario que los programas que creamos se puedan comunicar entre sus instancias.
- Para ello se hace uso de la **Internet**, pero primero es necesario entender como está estructurada. En sí debemos aprender 3 conceptos clave:
  - Dirección IP.
  - Puertos.
  - Protocolos de conexión.

# Dirección IP

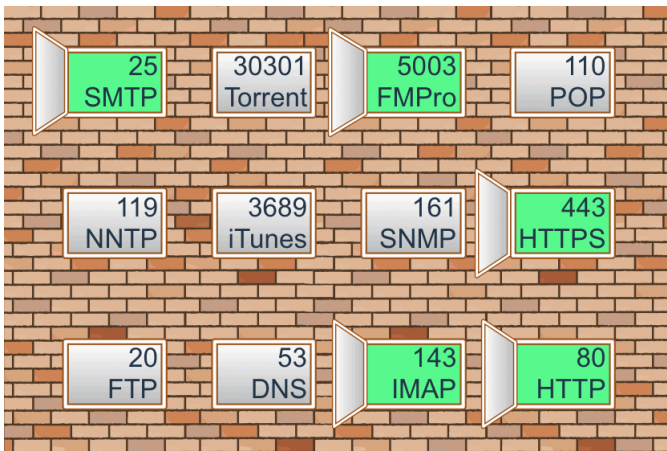
- **Etiqueta** que identifica a un dispositivo dentro de una red. Para los humanos es más simple manejarse con dominios.
- Esta dirección es **dinámica**, pues varía cada vez que se apague/encienda/conecte el router. Pero se puede hacer **fija** pagando.
- La IP **local** (o privada) identifica a un dispositivo dentro de una red. La IP **pública** identifica a nuestro equipo en Internet, pero no la podemos conocer directamente.

# Puertos

- Son puntos de comunicación dentro de una dirección IP.
- Aunque podemos encontrar una IP, es necesario diferenciar con qué proceso en particular nos queremos comunicar.
- Un programa puede hacer uso desde 0 a la cantidad de puertos que quiera.
- Un problema recurrente es cuando un programa intenta usar el puerto que está en uso por otro programa. Basta con cambiar el puerto para solucionarlo.

# Puertos

- Para simplificarlo, una forma de verlo es pensar en la IP como la dirección de un edificio y cada departamento como si fuera un puerto que está enviando información.



# Protocolos

## TCP v/s UDP

- Definen cómo se debe realizar el intercambio de información entre dos puntos de una red.



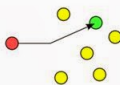
**TCP**

- Slower but reliable transfers**
- Typical applications:**
  - Email
  - Web browsing

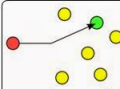


**UDP**

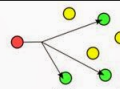
- Fast but non-guaranteed transfers ("best effort")**
- Typical applications:**
  - VoIP
  - Music streaming



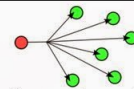
**unicast**



**unicast**



**multicast**



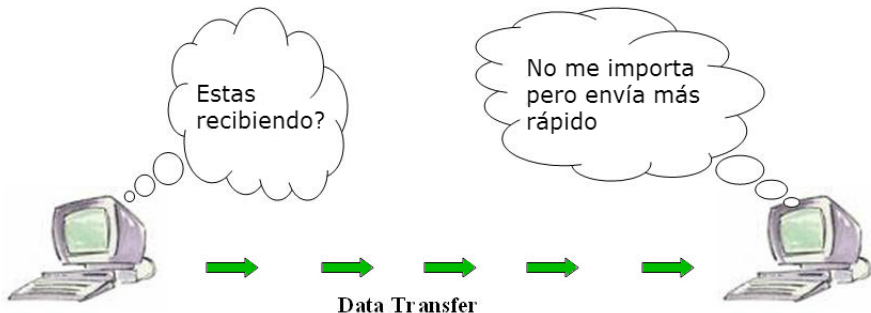
**broadcast**

# Protocolos

## UDP

- **UDP** no está orientado a la conexión. Una máquina A envía paquetes a una máquina B sin realizar una conexión previa y sin confirmar el recibo de los datos.

### UDP



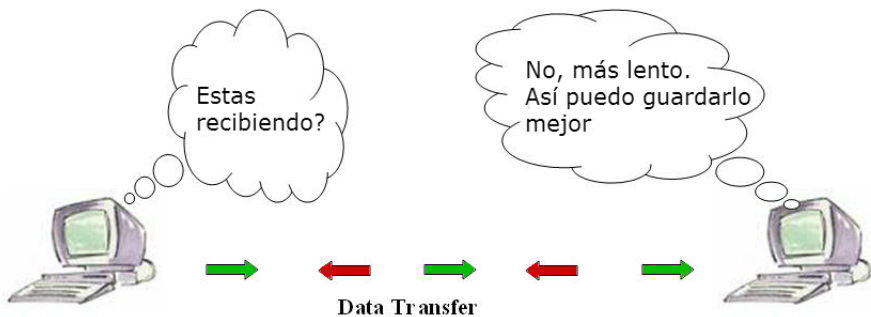


# Protocolos

## TCP

- **TCP** está orientado a la conexión. Una máquina A envía paquetes a una máquina B confirmando la recepción correcta de estos. Si los datos están corrompidos vuelve a solicitar estos datos.

### TCP



# Sockets

- En **Python** se utiliza el sistema de sockets.
- Se envían **bytes** entre ellos. Esto significa que podemos enviar objetos serializando con json y pickle.



$$\underbrace{203 \cdot 43 \cdot 12 \cdot 234}_{\text{IP address}} : \underbrace{80}_{\text{Port}}$$
  
$$\underbrace{\hspace{15em}}_{\text{Socket}}$$

# Sockets

```
1  import socket
2  un_socket_TCP = socket.socket(socket.AF_INET,
    socket.SOCK_STREAM)
3  un_socket_UDP = socket.socket(socket.AF_INET,
    socket.SOCK_DGRAM)
4  # metodos cliente
5  socket_cliente.connect(address)
6  # metodos servidor
7  socket_servidor.bind(address)
8  socket_servidor.listen(backlog)
9  socket_servidor.accept()
10 # metodos generales
11 un_socket.send(b'Hola mundo')
12 un_socket.recv(1024)
13 un_socket.sendto(bytes, address)
14 un_socket.recvfrom(bufsize[, flags])
15 un_socket.close()
16 un_socket.gethostname()
```

# Extra

```
"Hi, I'd like to hear a TCP joke."  
"Hello, would you like to hear a TCP joke?"  
"Yes, I'd like to hear a TCP joke."  
"OK, I'll tell you a TCP joke."  
"Ok, I will hear a TCP joke."  
"Are you ready to hear a TCP joke?"  
"Yes, I am ready to hear a TCP joke."  
"Ok, I am about to send the TCP joke. It will last 10  
seconds, it has two characters, it does not have a  
setting, it ends with a punchline."  
"Ok, I am ready to get your TCP joke that will last 10  
seconds, has two characters, does not have an explicit  
setting, and ends with a punchline."  
"I'm sorry, your connection has timed out."  
...Hello, would you like to hear a TCP joke?"
```