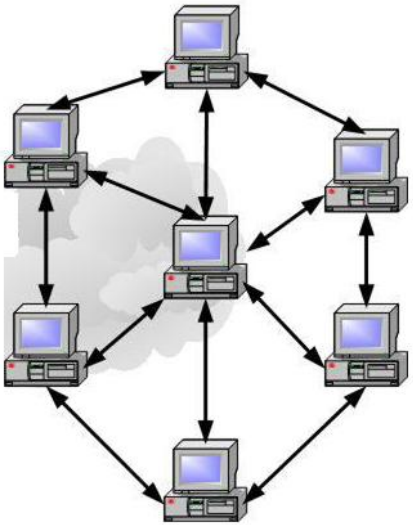
Networking - Part 1

Michel FACERIAS

Polytech - Université de Montpellier

16 décembre 2022

A *data communication* network or *DCN*[1] is a set of equipment linked together to allow computers to share resources located on or provided by network nodes.

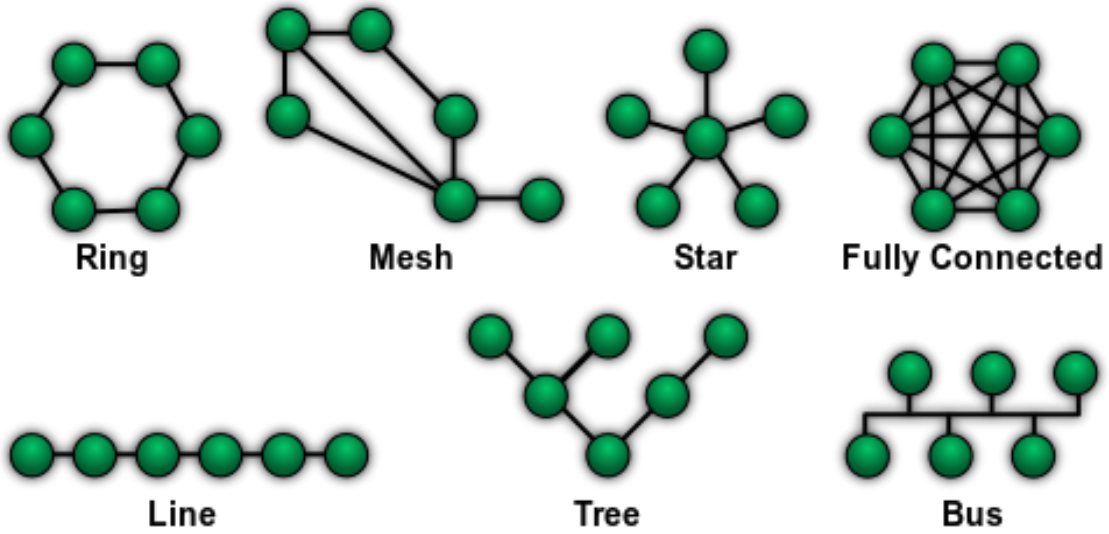


It is composed by two sorts of equipment :

- Data Communication Equipment (DCE) structure the smallest part of the network (like hubs, bridge and switches). They can't invisible in the *userland* ;
- Data Terminal Equipment, that include computer and other hosts (like routers). They should be visible in the *userland*.

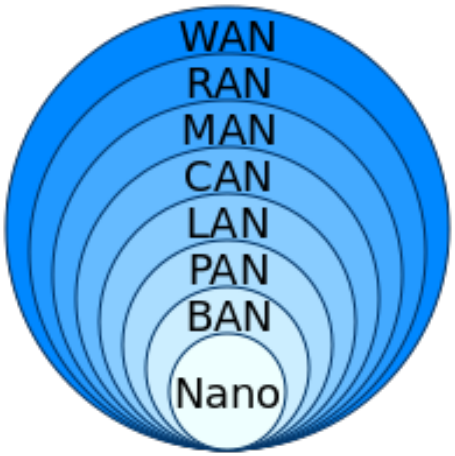
1. https://en.wikipedia.org/wiki/Computer_network

Through the *userland* view, the physical or geographic locations of network nodes and links have relatively little effect on a network. Here are some examples.



With many technologies, such as bus or star networks, the topology can significantly affect its reliability. A single failure can cause the network to fail entirely. In general, the more interconnections there are, the more robust the network is ; but the more expensive it is to install and maintain.

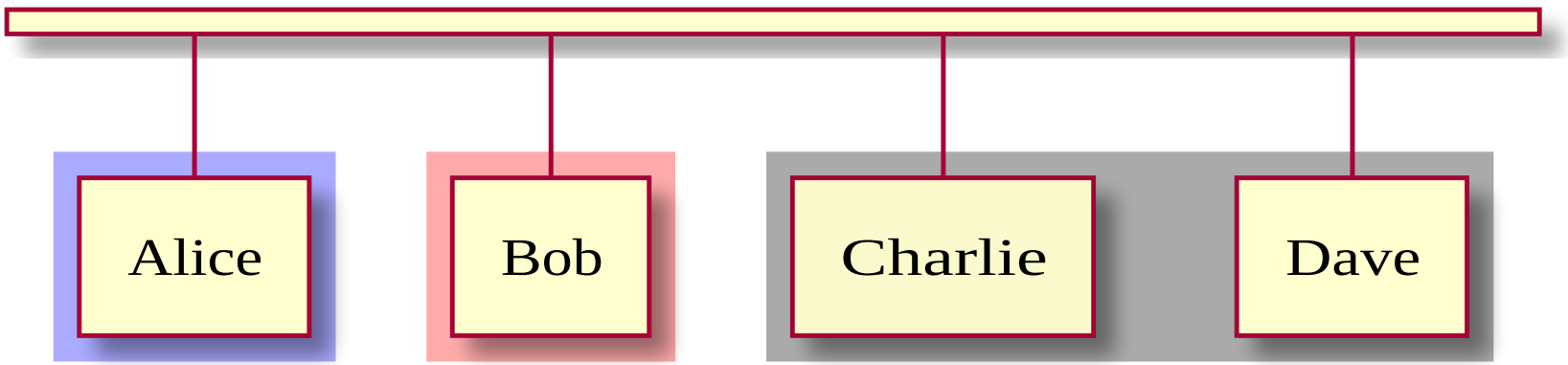
Networks may be characterized by many properties or features. Another distinct classification method is that of the physical extent or geographic scale.



The main scale factors are :

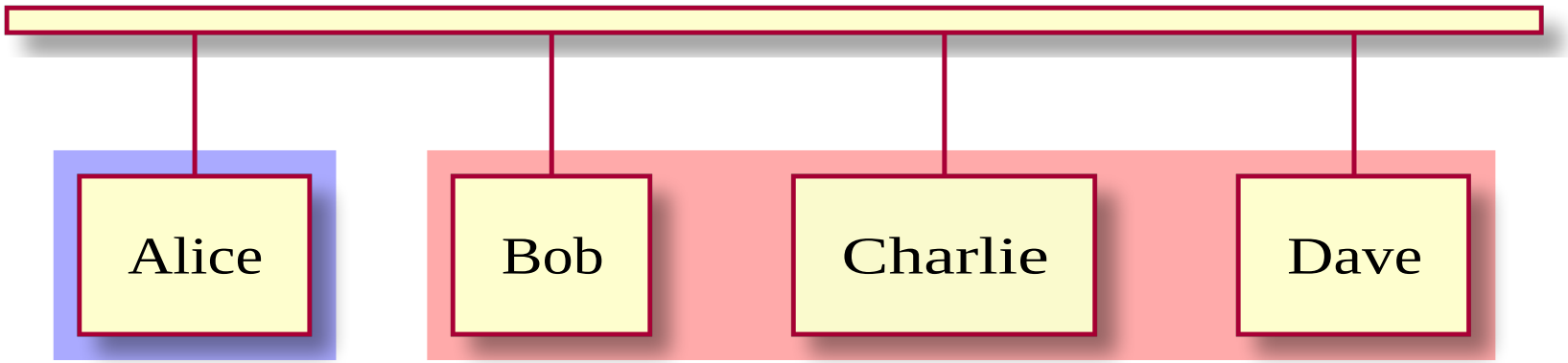
- Body Area Network (inside a computer motherboard)
- Personal Area Network (Bluetooth, ...)
- Local Area Network (the most known)
- Campus Area Network
- Metropolitan Area Network
- Regional Area Network
- Wide Area Network

This diagram shows an **unicast** connection.



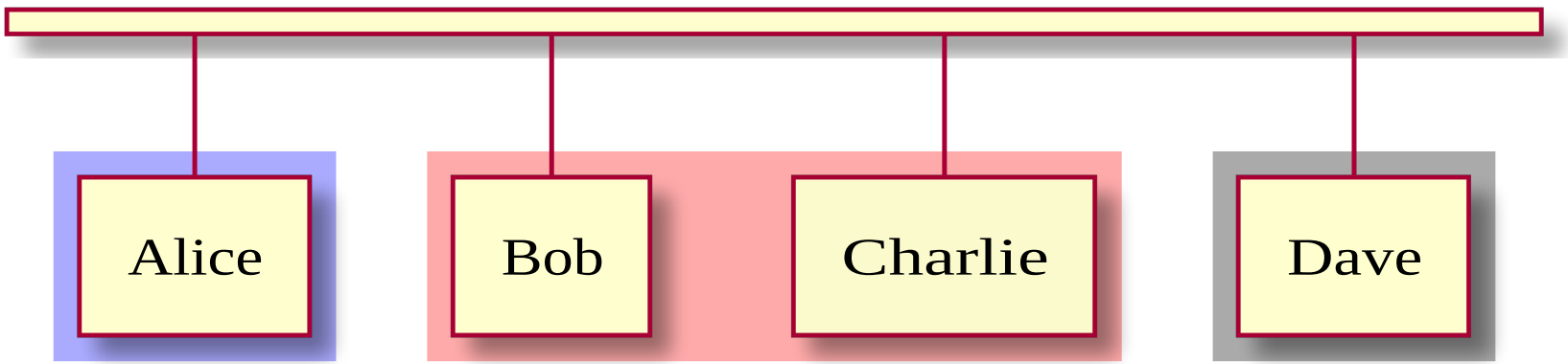
Alice is sending a message to Bob. Charlie and Dave are not concerned.
This is the default mode. It uses unicast addresses to identify the recipient.

This diagram shows an **brodcast** connection.



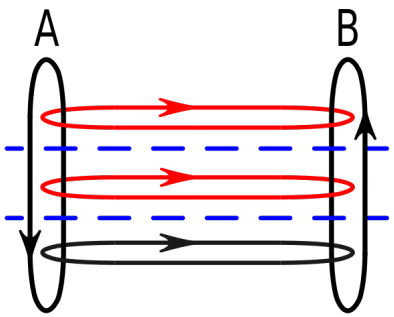
Alice is sending message to everybody.
This mode is used less often. It uses a special broadcast address.
We will come back to this in the next lesson ...

This diagram shows an **multicast** connection.



Alice is sending a message to Bob and Charlie.
Dave is not concerned.
This mode is used for special applications. It uses a special group address. We will come back to this in the next lesson too ...

A communication protocol is a system of rules that allows two or more entities to transmit information.
The protocol defines the rules : syntax, semantics, chronology and eventually recovery methods.
Message flows using a protocol suite use a layered model, with real and virtual communication ways :



- Black loops show the real messaging loops ;
- Red loops are the effective virtual communication between equivalent layers ;
- In *userland* mode, you can just act with the upper layer, without worrying about what happens on the other layers.

The Open Systems Interconnection (OSI) [2] model is a theoretical reference tool for understanding data communications between any two networked systems, even if differents.
It divides the communications processes into seven layers :

- Each layer both performs specific functions to support the layers below it ;
- Each layer offers services to the layers above it.

None of the real network protocols fully implements the OSI model.

2. https://en.wikipedia.org/wiki/OSI_model

This table show the seven [3] OSI model layers. [4]

Layer			Protocol data unit (PDU)	Function
Host layers	7	Application	Data	High-level APIs, including resource sharing, remote file access
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

3. Some says that OSI models contains two more layers, 0 is the physical medium, and 8 is the human in front of the keyboard.
4. `https://en.wikipedia.org/wiki/OSI_model`

The Internet protocol suite, commonly known as TCP/IP, is the set of communications protocols used in the Internet and similar computer networks.
The historical protocols of the suite are :

- a network layer protocol, the Internet Protocol itself (IP), providing an identification address ;
- a connection-oriented transport layer protocol, the Transmission Control Protocol (TCP) ;
- a connection-less transport layer protocol, the User Datagram Protocol (UDP) was implemented later ;
- a session layer protocol, embedded in TCP and UDP, providing process-specific separated transmission channels using the concept of the network port.

This table show the four TCP/IP model layers.

	Layer	Payload name	Function
4	Application	Data	The application layer includes the protocols used by most applications for providing user services or exchanging application data over the network. At the application layer, the TCP/IP model distinguishes between user protocols, like HTTP, and support protocols, like DNS.
3	Transport	Segment / User datagram	The transport layer establishes basic data channels that applications use for task-specif c data exchange in the form of end-to-end message transfer services.
2	Internet	Datagram	The Internet layer provides an unreliable datagram transmission facility between hosts. It has the responsibility of sending packets across potentially multiple networks.
1	Link	Frame	The protocols of the link layer operate within the scope of the local network connection to which a host is attached. This includes not only hardware implementations, but also virtual link layers such as networking tunnels (Virtual Private Networks).

This table show OSI vs TCP/IP model matching.

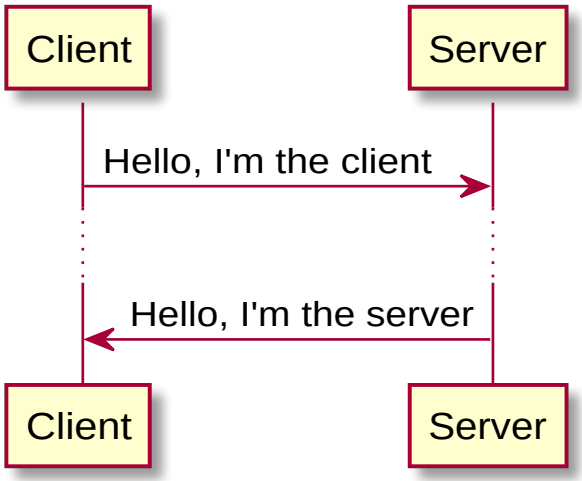
OSI Layer		TCP/IP Layer	Id
5	Session	Transport	Socket Port number TCP or UDP
4	Transport		
3	Network	Internet	IP address
2	Data link	Link	*Depends of implementation*

A sever is a host on which a service application is running.
Each service is mapped to a port. Most of them have *well known* values identifying a *well known* service.
This is partial view of /etc/services file from a Linux host.

```
echo        7/tcp
echo        7/udp
...
ftp-data    20/tcp
ftp         21/tcp
ssh         22/tcp   # SSH Remote Login Protocol
telnet      23/tcp
smtp        25/tcp   # Mail
...
domain      53/tcp   # Domain Name Server
domain      53/udp
...
http        80/tcp   # WorldWideWeb HTTP
...
```

Some service exist in UDP, or in TCP, or both... but not exactly for the same usage.

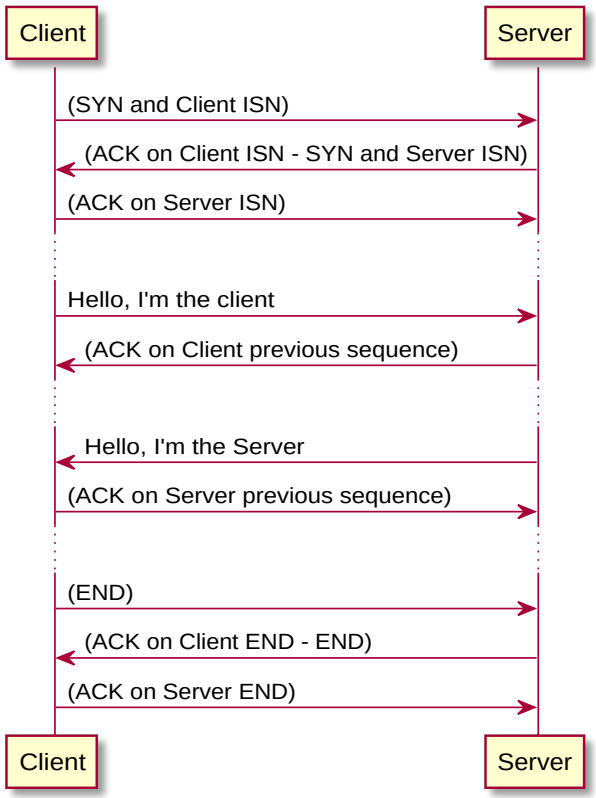
In UDP transport mode, messages are sent as **throwing a bottle overboard**. Here is an example :



- The client send his message to the server ;
- The server could answer to the client, if it must.

There is **no guarantee** that the message will get to the server.
There is **no guarantee** that the client get an answer (if it must !).
There is **no explicit start nor end**, it's a **pseudo-session** !

In TCP transport mode, messages are sent after a session is opened. Here is an example :

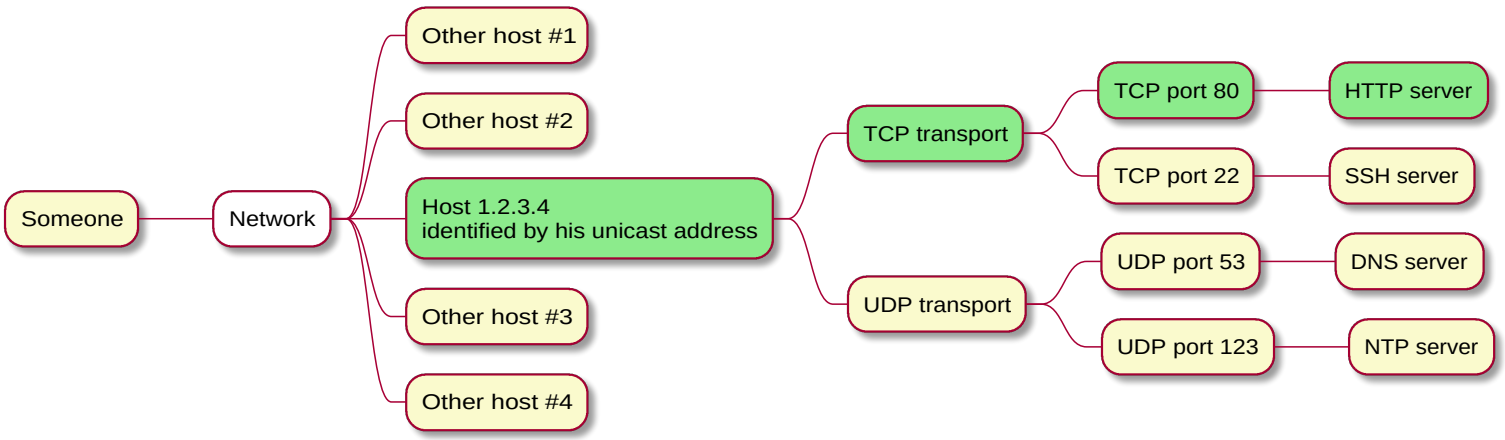


- The client initiate a **3 way start handshake** with the server ;
- The client send his message to the server ;
- The server **must** acknowledge it ;
- The server could answer to the client ;
- The client **must** acknowledge it ;
- The client initiate a **3 way end handshake** with the server ;

Acknoledge is a guarantee that each had received the message. There is an **explicit start and end**, it's a **session**, hanged by the client in this case study.

This picture show the decision steps, when someone use Firefox, to connect to :

`http:\\1.2.3.4\`



It shows how the destination port number is used to link to a service.

Example

Perl server source code

This is an example of a perl source code to build a minimal *syslog* server.

```
use IO::Socket;

my $buffer;

$sock = IO::Socket::INET->new(LocalPort => '514', Proto => '
    udp')||die("Socket: $@");

do{
  $sock->recv($buffer, 1524);
  print "$buffer\n";
}while(1);
```

End