

Software Platforms

From Transport to Application and Session

LM in Computer Engineering

Prof. Massimo Maresca

From Network Protocols to Sockets and Connections

Recollections (1)

- Network Architecture: Layers and Planes (Data, Control e Management)
- OSI Reference Model, TCP/IP, Ethernet
- Bridging/Routing
 - Mapping from Layer 3 to Layer 2
- Basic Internet philosophy and technology
 - Origin and evolution of the Internet experience (non managed network)
 - Standard based on authoritative people.
 - Autonomous System and routing technology (Intra/Inter domain routing)

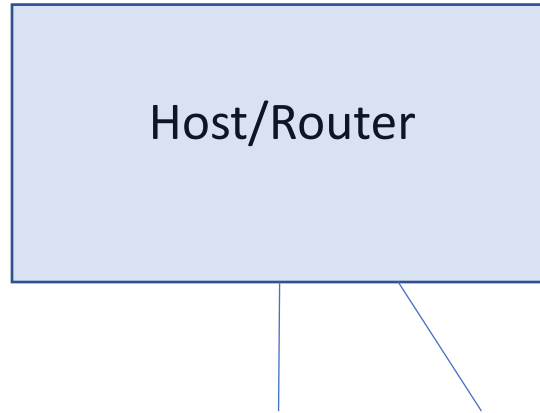
References: A. Tanenbaum, Computer Networks

Recollections (2)

- DNS
 - Mapping from the Name Space to the Address Space
- NAT
 - From protection of private address to Load Balancing and High Availability
- Network Virtualization
 - VLAN, VPN, Software Defined Networks

References: IETF, W3C, IEEE

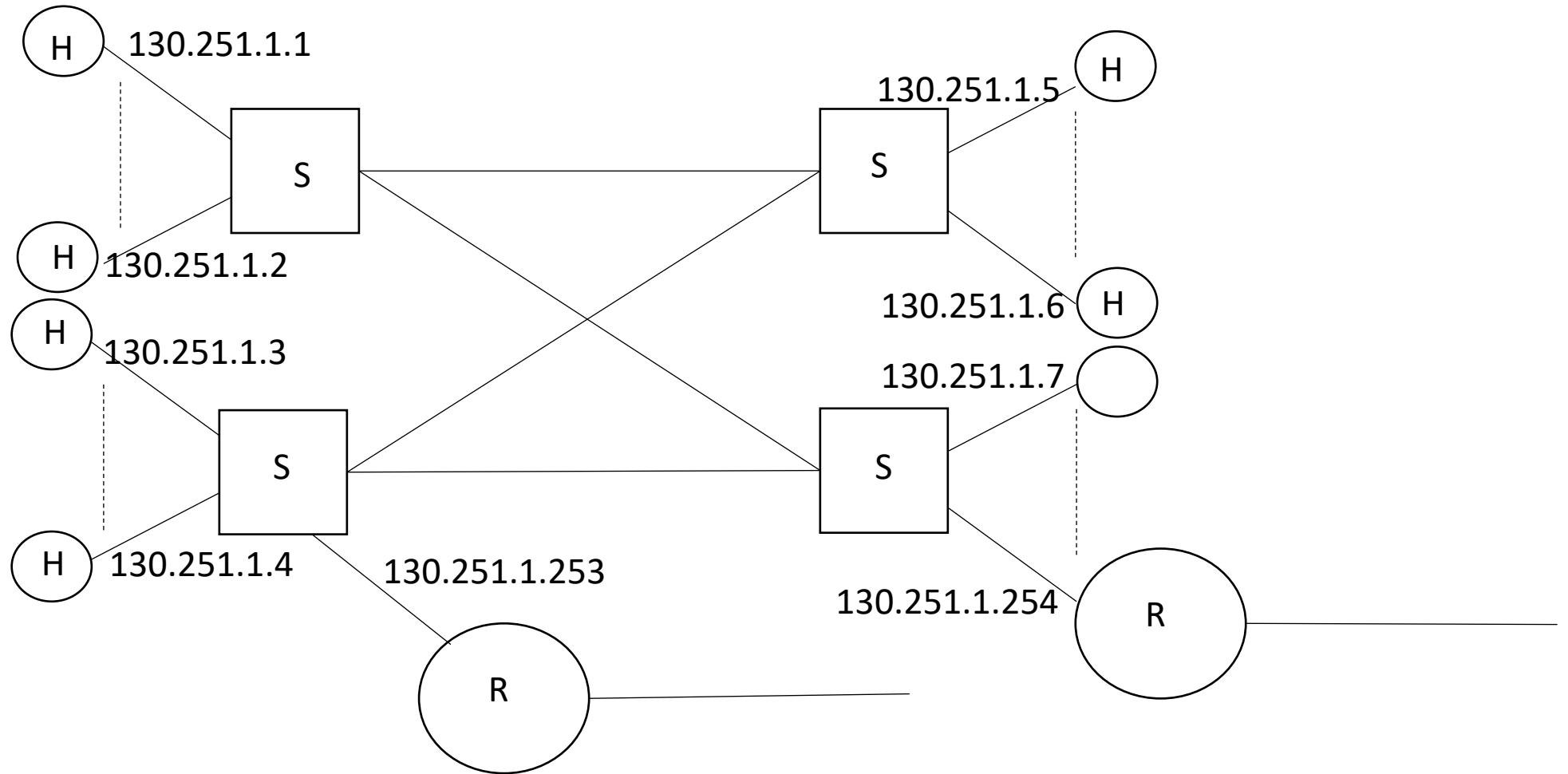
Host/Router Configuration



Connection-specific DNS Suffix: m3s.it
Description: Qualcomm QCA61x4A 802.11ac
Physical Address: 9C-30-5B-BD-5D-8D
DHCP Enabled: Yes
IPv4 Address: 172.25.1.95
IPv4 Subnet Mask: 255.255.255.0
Lease Obtained: Tuesday, September 18,
2018 9:11:37 AM
Lease Expires: Wednesday, September 19,
2018 1:46:51 PM
IPv4 Default Gateway: 172.25.1.254
IPv4 DHCP Server: 172.25.1.254
IPv4 DNS Servers: 8.8.8.8, 8.8.4.4

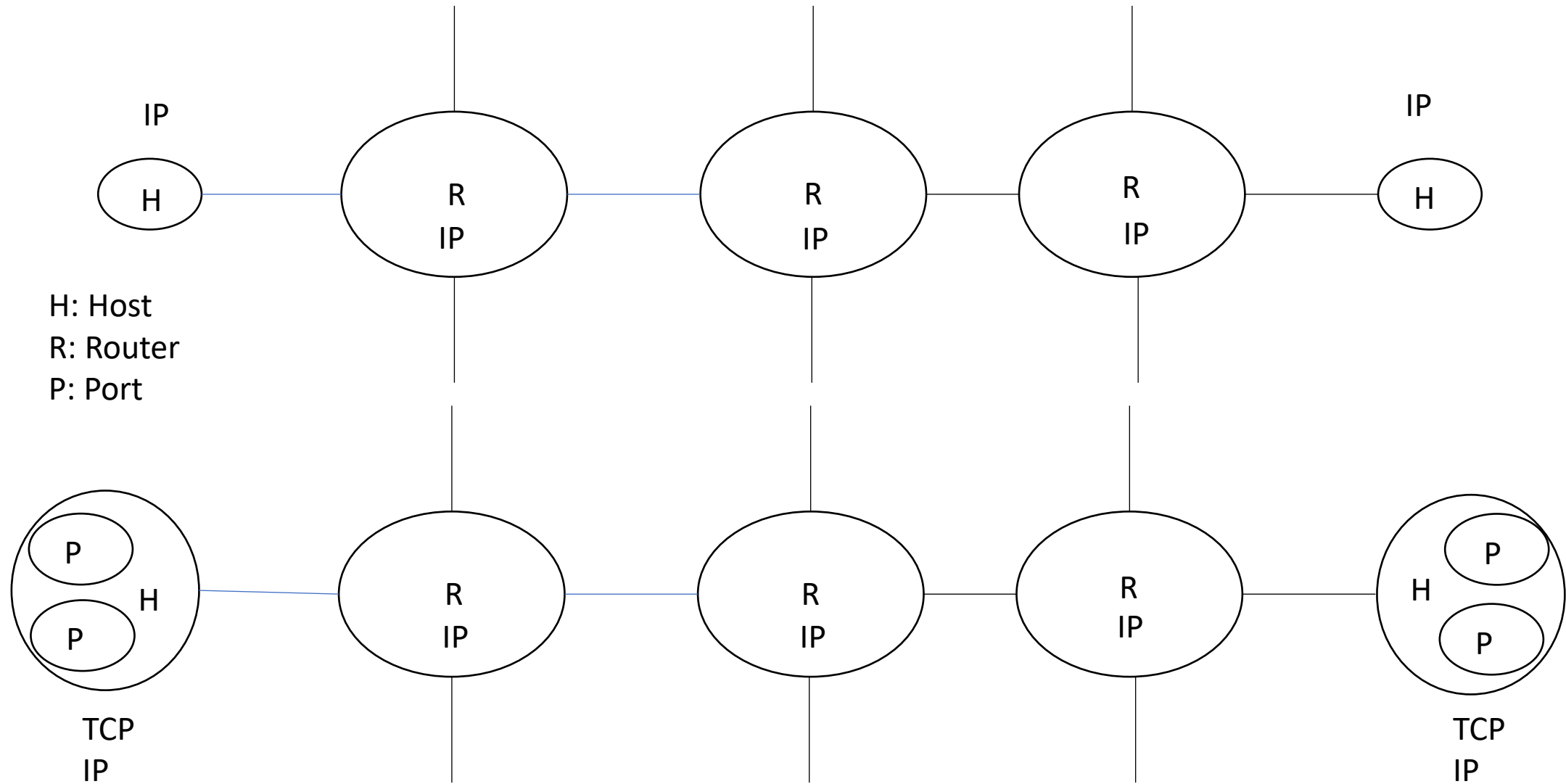
Every NIC is associated to a set of information

From MAC to IP



Netmask = 255.255.255.0

From IP to TCP



TCP

- Remind that TCP is:
 - stream oriented
 - reliable,
 - end-to-end,
 - connection oriented.
- TCP controls end-to-end transmission (error and congestion) as it assumes to work over an “unreliable “ layer, such as IP.
- As a consequence, a TCP socket must be “Connection Oriented” and refer to Communication Endpoints.
- IP Communication Endpoints correspond to IP addresses whereas TCP Connection Endpoints correspond to (IP Address, Port) pairs.

Basic Communication Mechanism: TCP Connection

Socket Abstraction

- In spite of the fact that it appears natural, it took years to come up with the Socket Abstraction after the introduction of TCP/IP;
- The Socket Abstraction can be compared to the File Abstraction
- Sockets natural refers to the underlying protocols (TCP, UDP, IP).

Ref. <https://docs.oracle.com/javase/tutorial/networking/sockets/index.html>

TCP Socket Programs

Servers:

MyTCPServer.java

Client:

MyTCPClient.java

- MyTCPServer: a socket based TCP Server.
- MyTCPClient: a socket based TCP Client.

From the Transport Layer to the Upper Layers

Stack TCP/IP including HTTP

HTTP
TCP
IP
MAC

FROM THE INTERNET TO THE WEB

- The **WEB** was born in 1989 at CERN in Switzerland as an **application of the Hypertext concept** to the global community of nodes provided by the Internet.
- The original idea is that of a set of “pages” (hyper)-linked to each other.
- The main components are:
 - A syntax (URL - Uniform Resource Locator) to locate resources over the Internet: e.g., <http://www.ietf.org/how/wgs/>
 - An application layer protocol (HTTP – HyperText Transfer Protocol) to connect client and servers.
 - A language (HTML – HyperText Markup Language) to code documents in such a way to support links to other documents: <a href=<http://...>> String

testo che linca
altro testo

URL, HTTP and HTML

- AN URL (example <http://www.ietf.org/how/wgs/>) includes:
 - Protocol (http)
 - Name (www.ietf.org)
 - Path (how/wgs), with reference to a home directory.
- HTTP (as well as its secure version HTTPS) is the protocol used to connect clients and servers over the Web.
- HTTP evolved from HTTP 1.0 to HTTP 1.1 to HTTP 2. From single HTTP Connections mapped to single TCP Connections, to multiple HTTP Connections mapped to single TCP Connections.
- HTML evolution from simple hypertext to from, equations graphics, XML, Video/Audio etc. (HTML5).

HTTP

- HTTP was introduced to complement the existing application protocols, which at that time at which the Web was conceived were ftp, telnet and smtp.
- HTTP is stateless and includes the following methods:
"GET" , "HEAD" , "POST" , "PUT" , "DELETE" , "TRACE" , "CONNECT".
- Requests are acknowledged. Examples of acknowledgements include:
 - 200 Success
 - 301 Page moved
 - 404 Page not found
 - 500 Server Error
- Data is typed to enable the activation of plug-ins/application (example: *text/html* is directly processed by the browser)

More on HTTP

- Originally HTTP Connections mapped on TCP Connections
- Then evolution to HTTP Persistent Connections with Sequential Requests mapped on TCP Connections
- Benefits:
 - Save TCP 3-way handshaking
 - Save TCP Congestion Control Slow-Start
- Evolution to Persistent Connections with Pipelined Requests mapped on TCP Connections
 - Improve Server performance thanks to parallelization

Static Web vs. Dynamic Web

- Static Web: Content Retrieval from Storage
- Dynamic Web: Content Generation by software components activated at request time.
- Server Side Dynamic Page Generation
 - CGI – Common Gateway Interface
 - JSP - Java Server Pages
 - Servlet in java i dynamic si possono fare con i servlet
- Client Side Dynamic Page Generation
 - Javascript
 - Applet/ActiveX component
 - VBscript

HTTP Programs

Servers:

MyHttpServerOverTCP
MyHttpServer

Client:

GetTest.java

- MyHttpServer: a Http Server built on top of the Http library.
- MyHttpServerOverTCP: a HTTP Server built over a TCP socket. No Http Library is used. The Http behavior on the server side is obtained through a response message compliant with the Http standard.
- GetTest: a Http client to be used to test the programs.

How about Security ?

- Two issues:
 - Authentication
 - Encryption
- Authentication :
 - Preliminary Message Exchange based on Server Public Key (Server Authentication)
 - Symmetric Key Negotiation
 - Client Authentication
 - Data Exchange

HTTP
HTTP
TCP
IP
MAC

How about the Session Layer ?

- The OSI RM includes a Session Layer which the Web seems to lack. Is this really the case ?
- The stateless nature of HTTP prevents from creating client-server pairings. Each HTTP data exchange is autonomous.
- The cookie mechanism was introduced to support such a feature. When a client requests a page, the server may provide additional information under the form of a cookie. A cookie is a string (typically shorter than 4KBytes) and may include an expiration date.
- What can a cookie be used for ?
 - Client identification: Every time a client connects to a server the server recognizes him/her.
 - Session Management in complex interactions: e.g., in e-commerce sites
 - User behavior tracing: e.g., to track web site access (3rd party cookies)

COOKIE EXAMPLES

SESSION COOKIES

- In the HTTP Response:

HTTP/1.0 200 OK

Content-type: text/html

Set-Cookie: cookie-name=cookie-value

- In the HTTP Request

GET /page.html HTTP/1.1

Host: www.myserver.org

Cookie: cookie-name=cookie-value

PERMANENT COOKIES

- In the HTTP Response:

HTTP/1.0 200 OK

Content-type: text/html

Set-Cookie: cookie-name=cookie-value; Expires=Wed, 21 Oct 2015 07:28:00 GMT;

COOKIE USAGE

- Client Identification inside sessions
 - The server maintain a Session
- Client Identification in successive connections
 - The server maintain profiles and automatically connect client requests to existing profiles
- Web Activity tracking
 - Client Identification through third party cookies