

# Web Technologies

## Lecture 1

### The Internet and HTTP

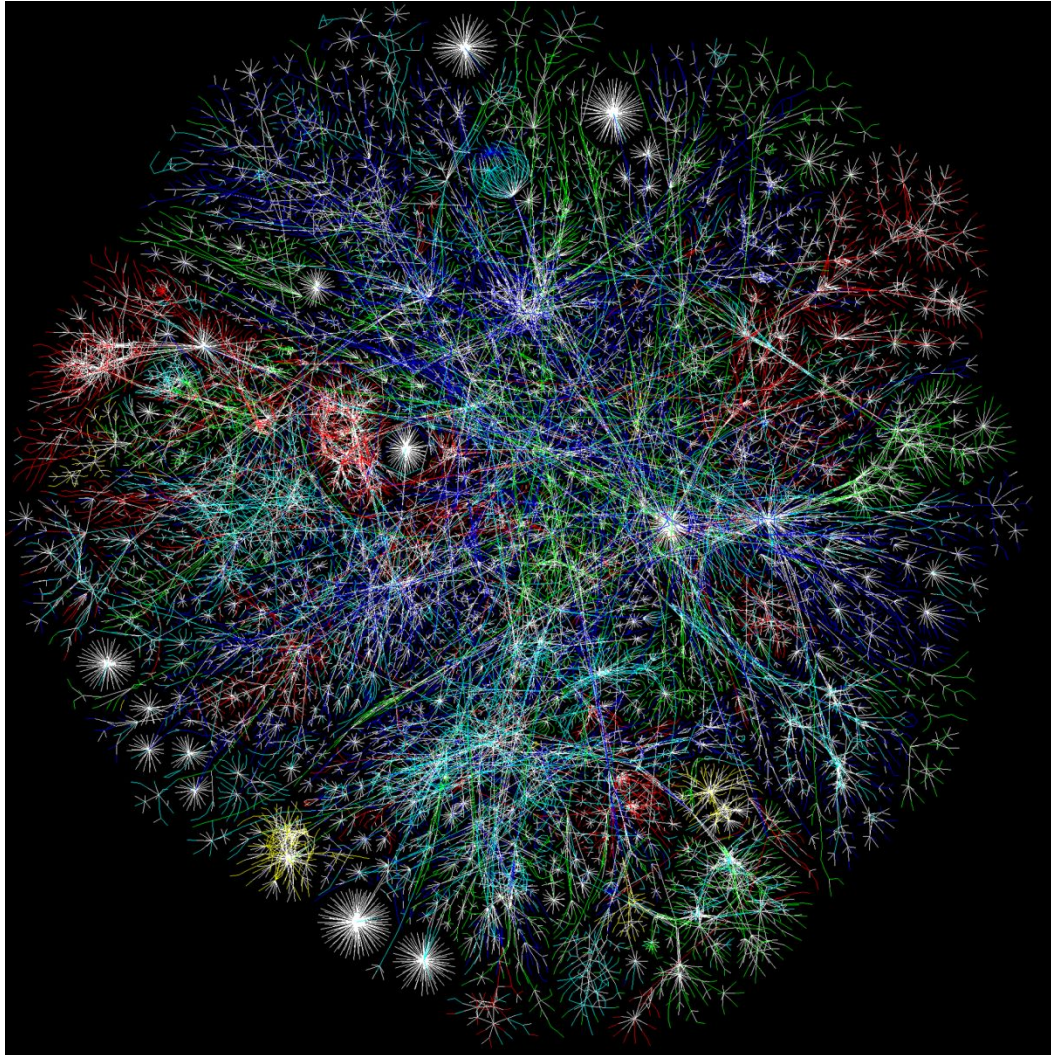
# The Internet

- From *internettted*, meaning interconnected
- Global system of interconnected computer networks
- 2014 – more than 38% of the world is connected
- Relies on the TCP/IP protocol

# A bit of history

- 1960's ARPANET is created
  - Early packet switching protocol
- 1981 - ARPANET is expanded by the National Science Foundation (NSF) - NSFNET
- 1982 - TCP/IP is standardized
- Late 80's & early 90's ISPs begin to emerge
- 1990 - ARPANET is decommissioned
- 1990 - 1.5 mbs over the Atlantic between Cornell University and CERN
- 1990 - 1<sup>st</sup> web browser called WorldWideWeb
- 1995 - the Internet is fully commercialized
- 2007 - over 97% of the world telecom is over the Internet

# Connected World

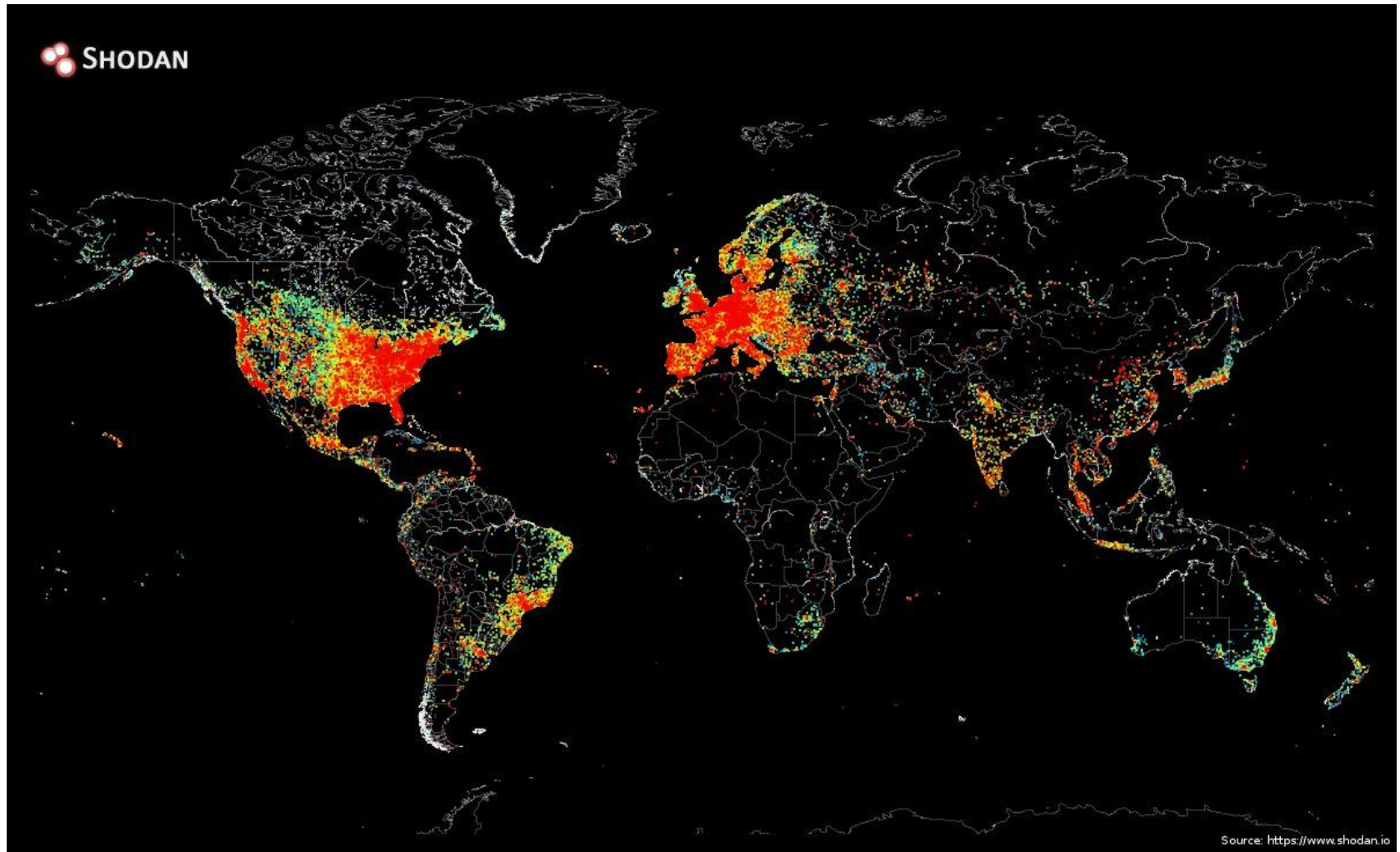


# The Internet of Things

- The Internet is a complex beast
  - Mobile devices
  - Smart sensors
  - Social networks
  - Etc.



# The Internet today





# Towards a networked society

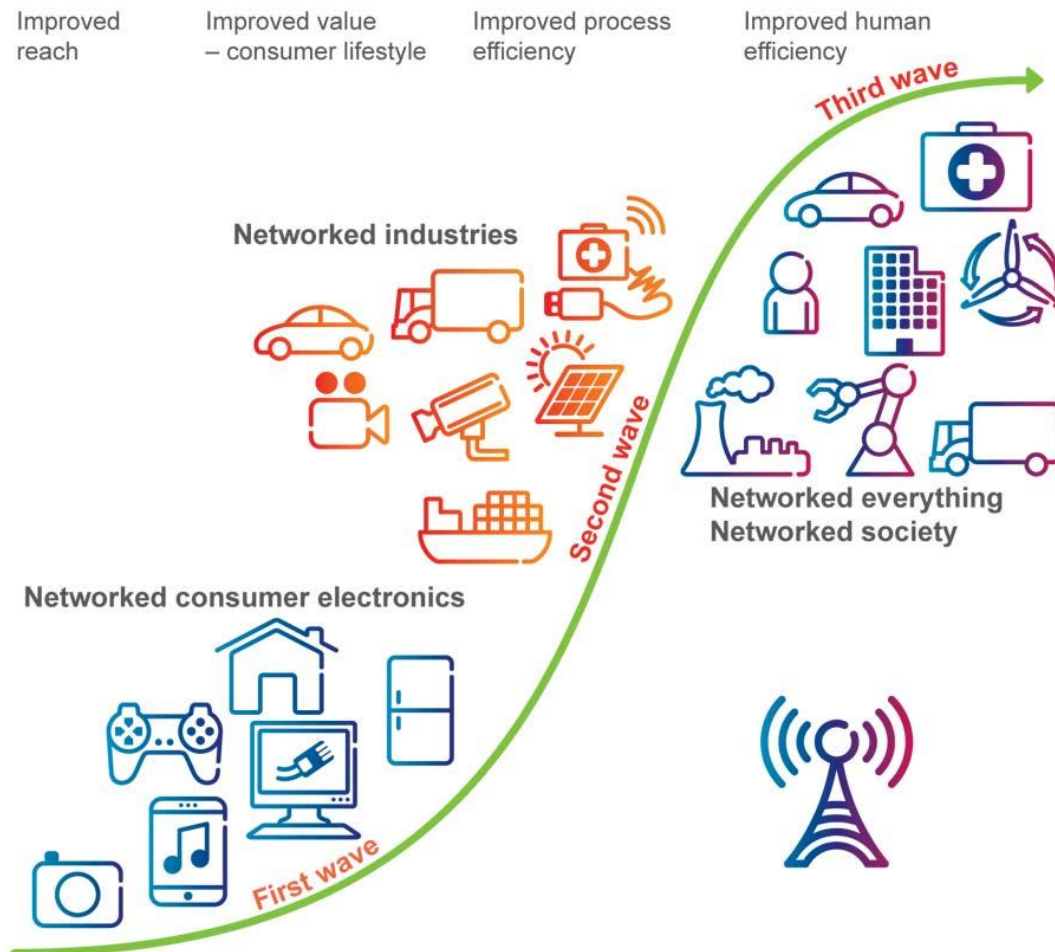
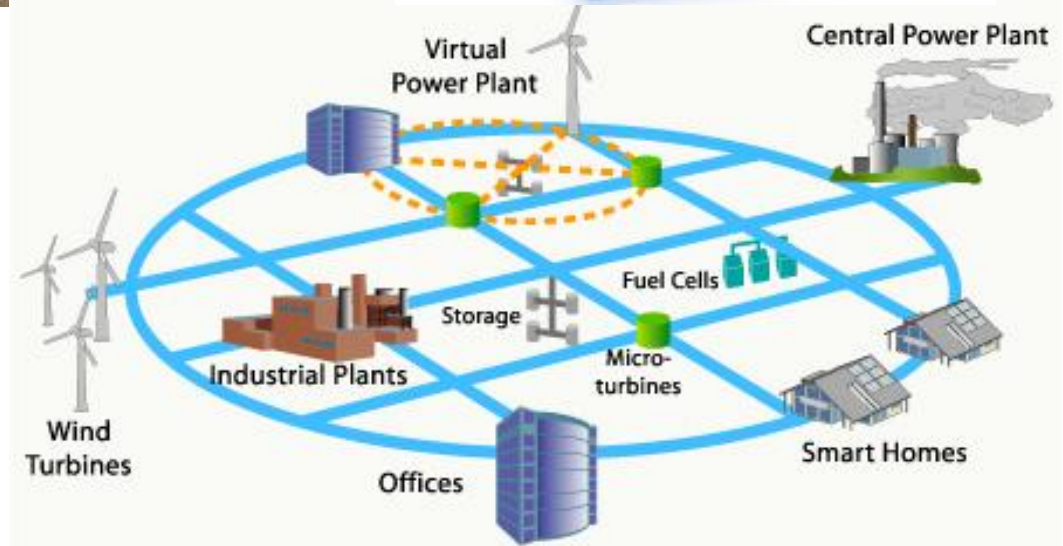
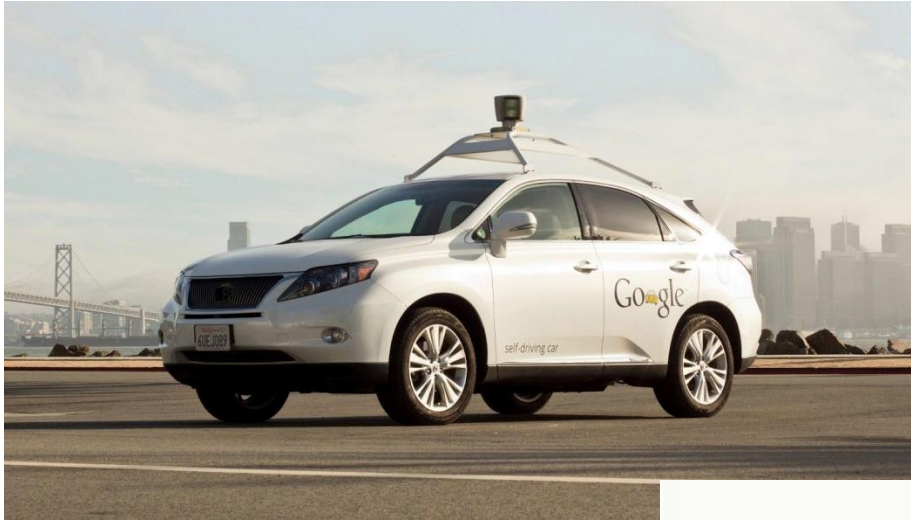


Figure 2. The three waves of connected device development.

# Examples



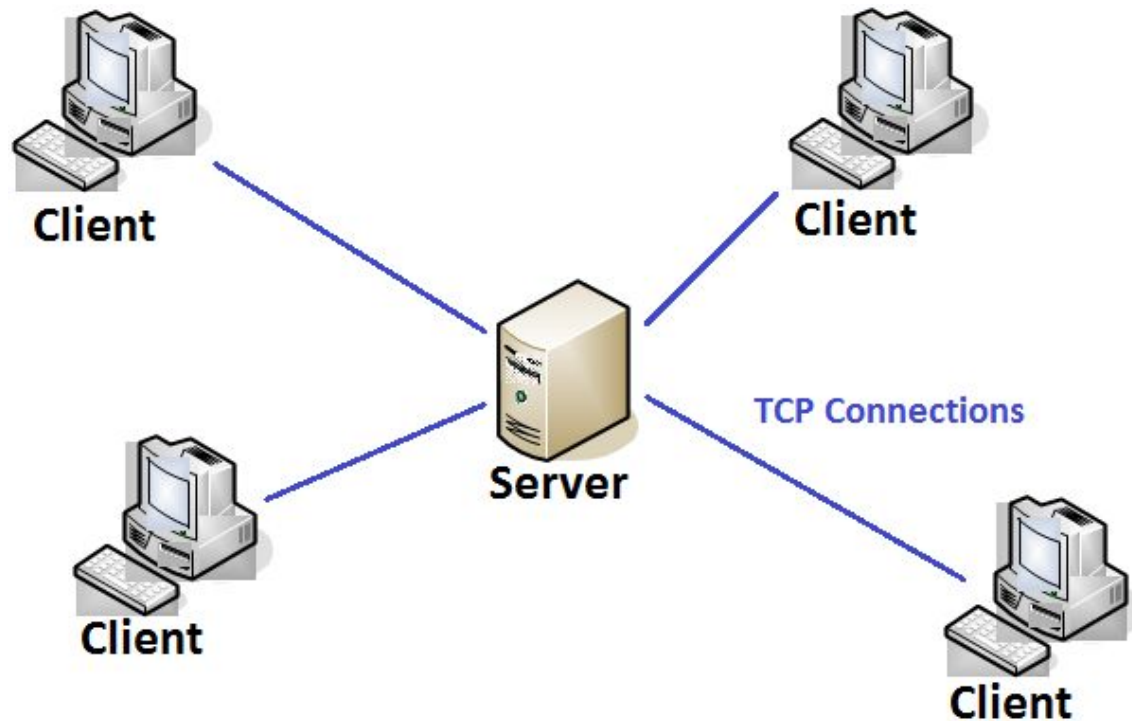


# Fundamentals

- Client-server model
  - E.g., web browsers
- P2P
  - E.g., file sharing software
- TCP/IP for communication
- Many application level protocols
  - HTTP for web traffic
  - SSH for file transfer
  - POP3 for email
  - Etc.

# Client-server architecture

- Many clients talk to one or more (via a proxy) servers



# World Wide Web

- A collection of documents and other resources
- URI – Uniform Resource Identifier
  - 2 subclasses:
    - **URL** – Uniform Resource Locator
      - Format: *http://<host>[:<port>]/[<resource>][?<query>]*
      - Example: `http://www.test.ro:8080/get/animals?type=mamal`
    - **URN** – Uniform Resource Name
      - Format: *<scheme>:<hierarchy>[?<query>][#<fragment>]*
      - Example: `urn:animals:mamal:ferret:nose`

# HTTP

- Hyper Text Processing Protocol
  - 1999: HTTP/1.1
    - Reuse the same connection for multiple resource requests
    - Less latency
  - May 2015: HTTP/2 (proposed)
    - Header field compression
    - Concurrent exchanges on the same connection
  - HTTP/1.1 format:
    - Request
      - [method] [resource] [version]<CRLF> [header]: [value]<CRLF> <CRLF>*
    - Response
      - [version] [status] [message]<CRLF>*
      - [header]: [value]<CRLF> <CRLF>*
      - [body]*

# HTTP sessions

- Sequence of network request-responses
  - Establish TCP connection to a port (usually 80)
  - Server listens for requests from clients
    - GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH
  - Server sends back status and message plus the body which represents the requested resource
    - Status codes:
      - Informational (1xx)
      - Successful (2xx): 200
      - Redirection (3xx): 301, 302
      - Client error (4xx): 404
      - Server error (5xx): 500

# Example

- **Request**

GET /index.html HTTP/1.1<CRLF> Host: www.example.com<CRLF> <CRLF>

- **Response**

HTTP/1.1 200 OK<CRLF> Date: Mon, 23 May 2005 22:38:34 GMT<CRLF> Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)<CRLF> Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT<CRLF> Etag: "3f80f-1b6-3e1cb03b"<CRLF> Accept-Ranges: bytes<CRLF> Content-Length: 438<CRLF> Connection: close<CRLF> Content-Type: text/html; charset=UTF-8<CRLF> <CRLF> <Content ...>



# HTTP session state

- HTTP is stateless
  - Does not retain information or status about requests
  - Can be bypassed with cookies, server side sessions, hidden variables, web forms.

# Synchronous vs. asynchronous

- Synchronous = blocking
  - A complete page must be loaded with every client request
  - AJAX (2005) changed the game by making asynchronous communication possible
- Asynchronous
  - Perform background tasks without blocking the interface
    - Example: load page content gradually (Google maps), autocompletion (Google search)

# What to expect from this lecture

- Basic information about Internet technologies
  - State vs stateless
  - Synchronous vs. asynchronous
  - Client side technologies
    - HTML, CSS
    - XML, DOM, AJAX, JQuery, JSON
  - Server side technologies
    - Web servers, templating engines, PHP, databases
  - Advanced topics on
    - Web services
    - RSS
    - Cloud fundamentals
    - Message queues

# Grading

- Written exam (60%)
  - Multiple choice
- Lab assignments (40%)
- Documentation
  - Personal page: [www.info.uvt.ro/mfrincu](http://www.info.uvt.ro/mfrincu)
  - Google