# Internet, Principes et Protocoles (IPP)

# TCP

Transmission Control Protocol

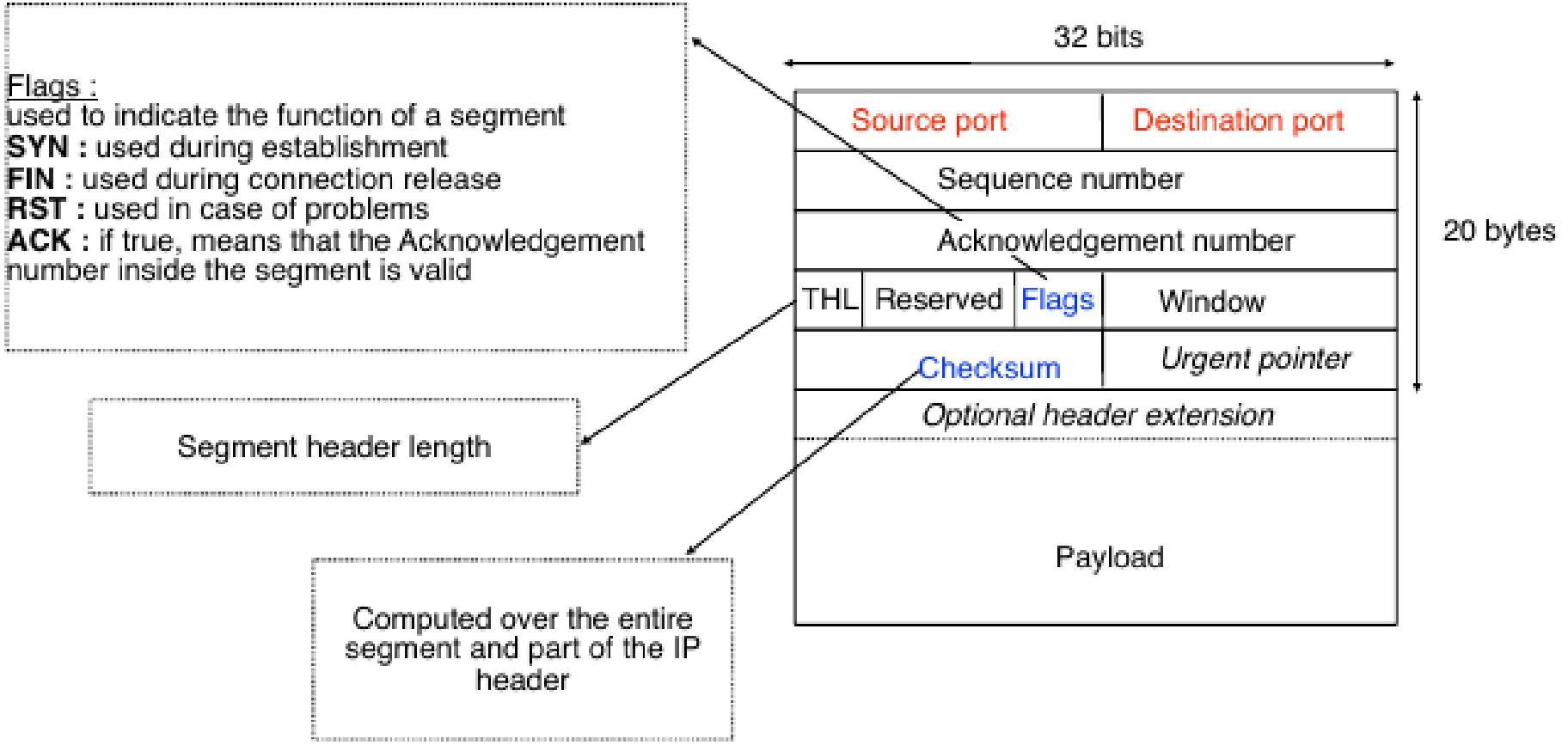
Provides a reliable byte stream service

Characteristics of the TCP service

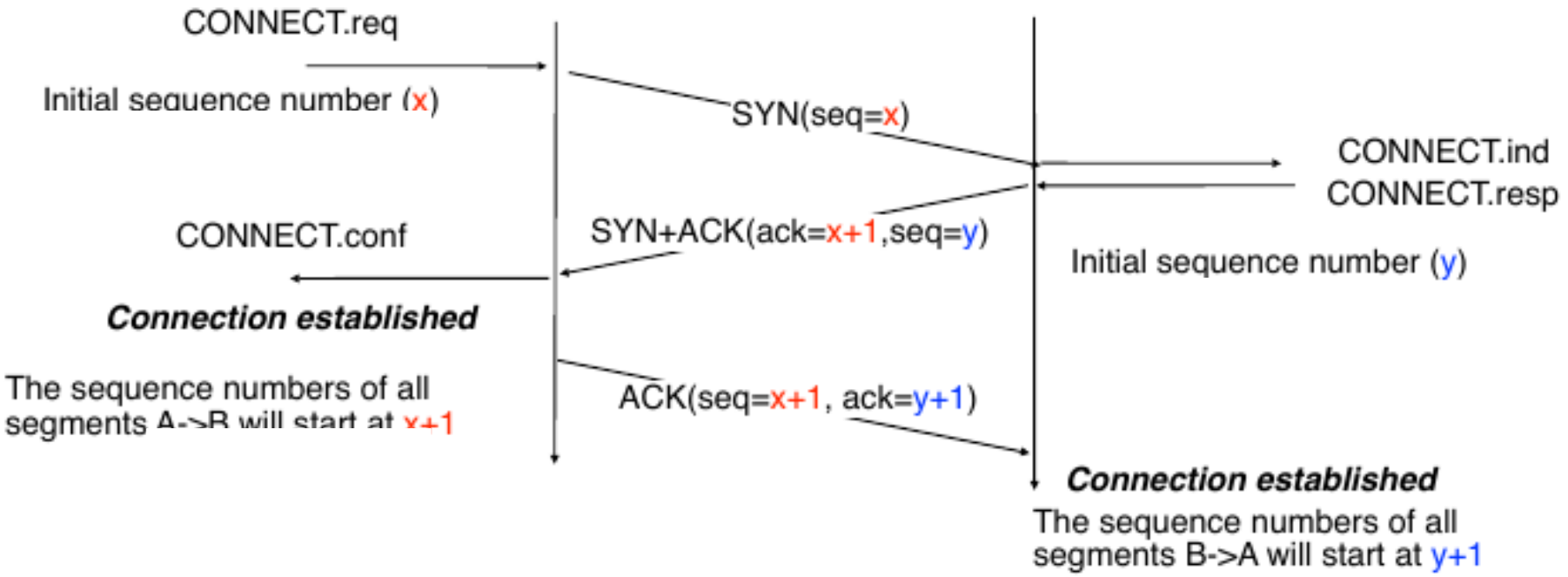
- TCP connections
- Data transfer is reliable
  - no loss
  - no errors
  - no duplications
- Data transfer is bidirectional
- TCP relies on the IP service
- TCP only supports unicast

# TCP Paquet

## Single segment format



# TCP 3-way handshake



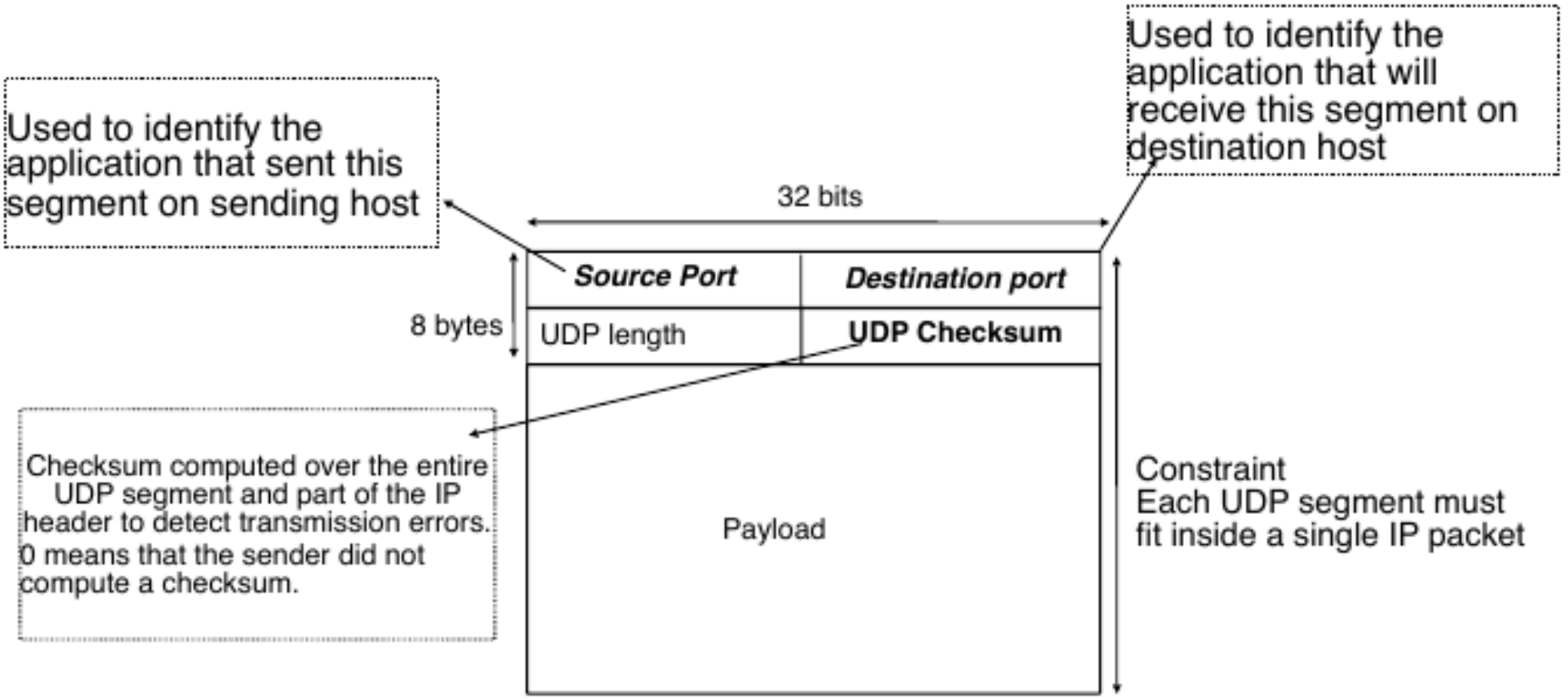
# UDP

User Datagram Protocol (UDP)
- The simplest transport protocol

Goal
- Allow applications to exchange small SDUs by relying on the IP service
  - on most operating systems, sending raw IP packets requires special privileges while any application can use directly the transport service

Constraint
- The implementation of the UDP transport entity should remain as simple as possible

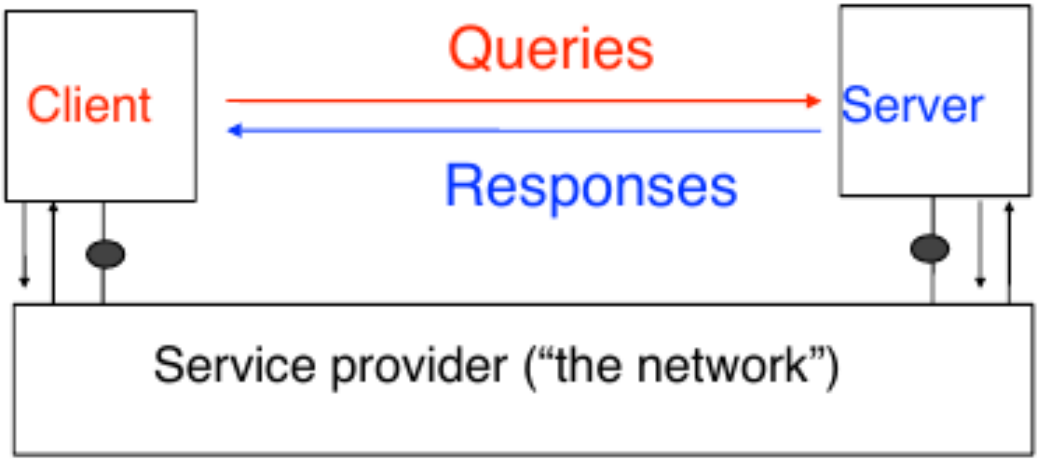
# UDP Segment


Used to identify the application that sent this segment on sending host
Used to identify the application that will receive this segment on destination host
32 bits
Source Port
Destination port
8 bytes
UDP length
UDP Checksum
Payload
Checksum computed over the entire UDP segment and part of the IP header to detect transmission errors. 0 means that the sender did not compute a checksum.
Constraint
Each UDP segment must fit inside a single IP packet

# Application Layer

- Highest layer, allowing hosts to exchange different kind of information (sending emails, ssh, serving websites, REST APIs,...)
- Relying on the lower protocols for the transmit ion and management of the information.
- Often using on the client-server model

# Client-Server Model




Client
interacts with server through transport layer
sends queries or commands

Server
Answers the queries received from clients
Executes the commands from clients
Many clients can use the same server

Example : email, www, ...

# Client-Server Model

On the Internet, applications can use two different transport services

The service provided by the User Datagram Protocol (UDP)

unreliable connectionless service with error detection

The service provided by the Transmission Control Protocol (TCP)

reliable bytestream connection-oriented service

# Names and addresses

Address of a server
- IP Address of the host on which the server is running
- port number (TCP or UDP)
  - usually well known port number

Drawback
- Difficult to remember an IP address for a human

Idea
- Replace IP address by a hostname
- Easier for humans
  - but IP address is necessary to contact server

How to translate a hostname in an IP address ?

# Names and addresses

`hosts.txt` file

contains the name-address table

must be updated regularly

```
#
# Internet host table
#
127.0.0.1       localhost
138.48.32.99    babbage
138.48.32.100   leibniz
138.48.32.1     routeur
138.48.32.92    corneille
138.48.32.107   backus
138.48.20.152   arzach
138.48.32.137   almin01
138.48.32.170   duke
```

cannot be used in a large network

# Domain names

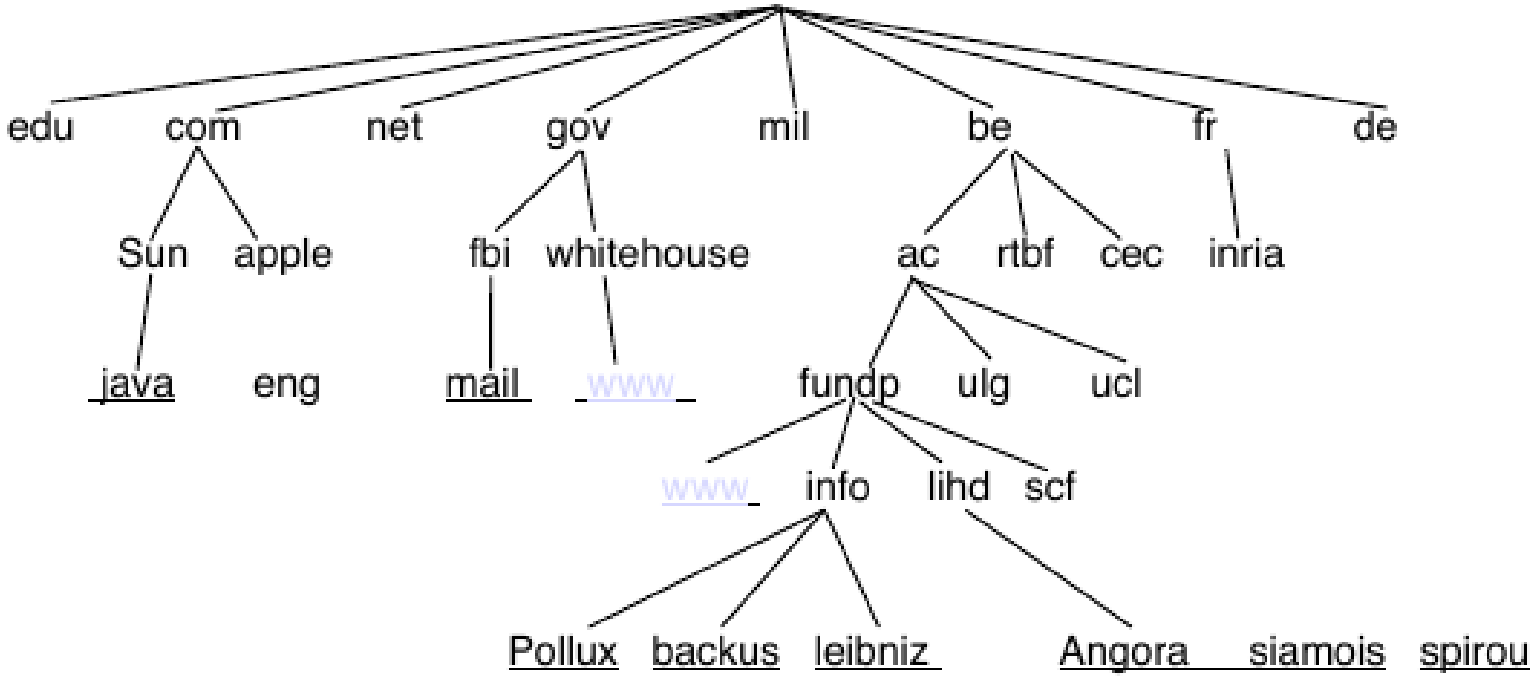
Requirement
- Host names should be unique

How to achieve this in a scalable manner ?
- Introduce hierarchy
- Each hostname is composed of two parts
  - domain name (globally unique)
  - hostname (unique within a given domain)

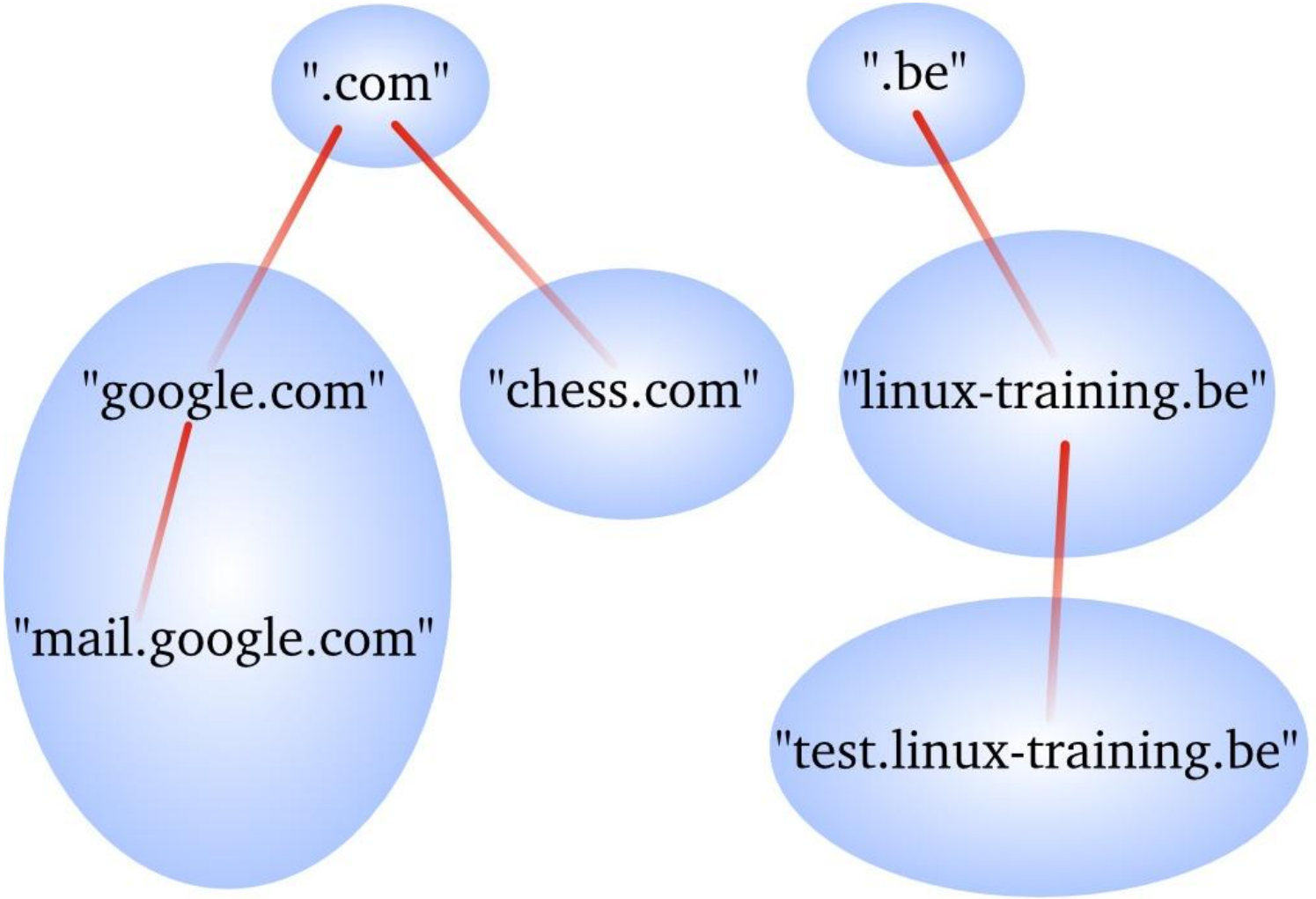
How to uniquely distribute domain names ?
- Introduce hierarchy
- A small number of top-level domain names
- Inside each top-level domain, allocate uniquely second level domain names
- Inside each seconde-level domain, allocate uniquely either third-level domain names or host names,

# Domains

Tree of all host names



# Domains and sub-domains



# Domains

How to efficiently translate a host name ?

- By using a centralised database
  - there are more than 1 billion host names today
- By using a distributed database
  - DNS : Domain Name System
  - relies on the hierarchy of domain names
  - there is one server responsible for each domain and this server must be queried to translate host names inside this domain

# DNS

- From Wikipedia:

  *The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers [...] connected to the Internet or a private network. I*

  *t translates more domain names to the numerical IP addresses needed for locating and identifying computer services and devices.*

  *By providing a worldwide, distributed directory service, the Domain Name System has been an essential component of the functionality of the Internet since 1985.*

- The ‘yellow pages’ of the Internet: give a name, receive an address.

# DNS Example

- When requesting duckduckgo.com in your browser, your browser will silently make a DNS request to get the ip.
- The request will go to a DNS Server (DNS Recursor), that will answer the DNS Query with the IP’s of the target.
- The browser will communicate with one of/the IP found to get the web-page.

# DNS

- DNS components:
  - The **DNS Recursor**: The DNS recursor is a server designed to receive queries from client machines. Typically the recursor is then responsible for making additional requests in order to satisfy the client's DNS query.
  - The DNS **Root Server**: the first step in translating (resolving) host names into IP addresses.
  - **TLD (Top Level Domain) Server**: This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is "com").
  - **Authoritative Nameserver**: The authoritative nameserver is the last stop in the query. If the authoritative name server has access to the requested record, it will return the requested IP address.

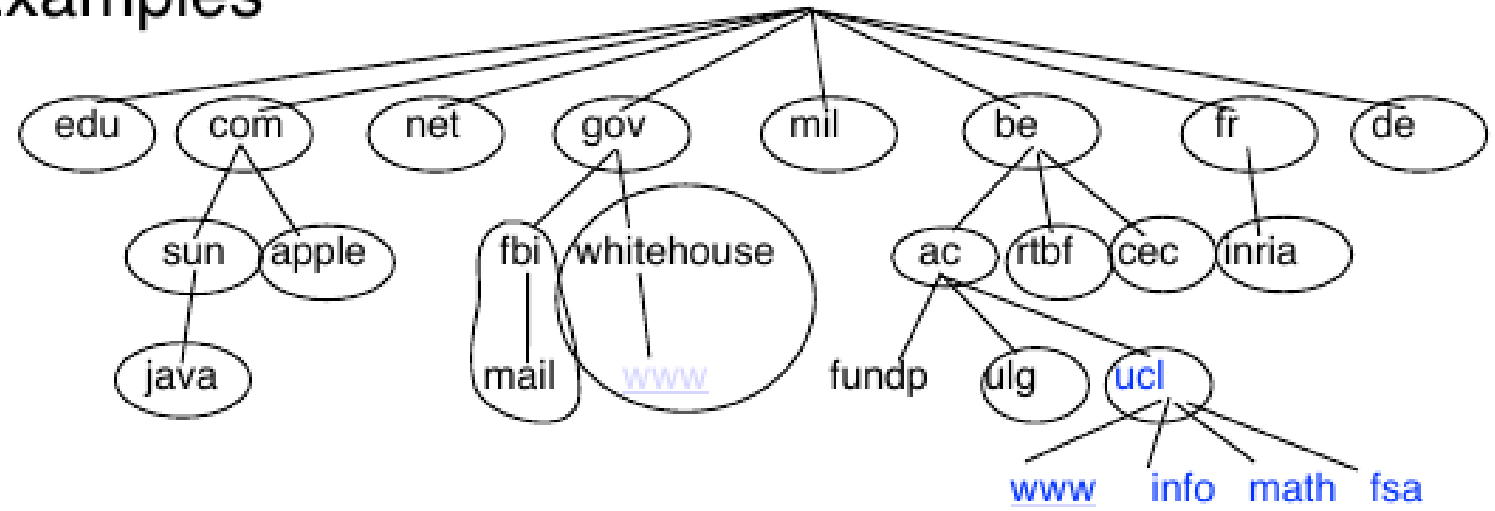
# DNS and DNS servers

Domain Name Service (DNS)

Each DNS server is responsible for a domain and knows

- The IP addresses of all host names in this domain
- The IP addresses of the DNS servers responsible for subdomains

Examples



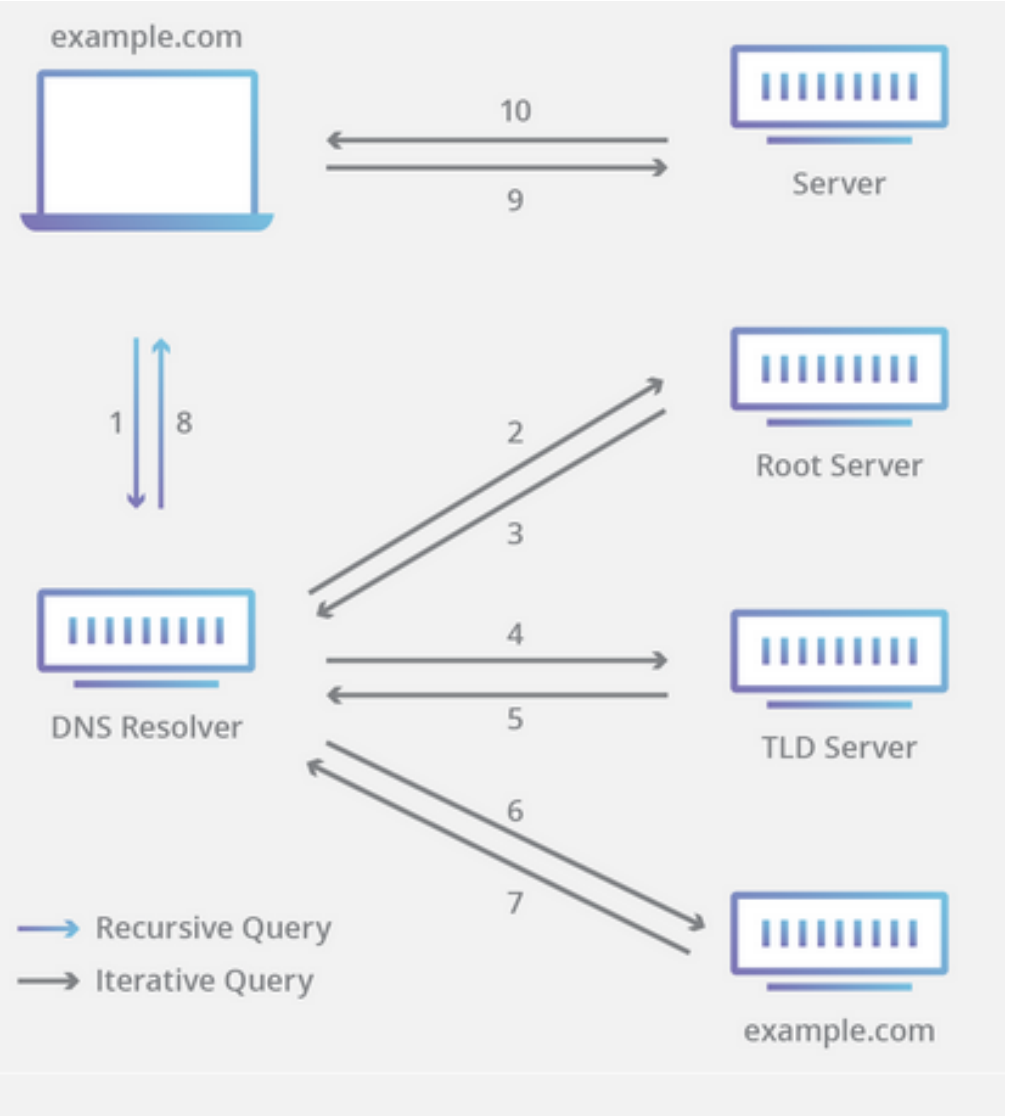

java.sun.com

www.ucl.ac.be

# DNS In Practice

The 8 steps in an iterative DNS lookup:

- A user types ‘example.com’ into a web browser which sends the query to a DNS recursive resolver.
- The resolver then queries a DNS root nameserver.
- The root server then responds to the resolver with the address of a TLD DNS server (such as .com or .net), which stores the information for its domains.
- The resolver then makes a request to the .com TLD.
- The TLD server then responds with the IP address of the domain’s nameserver, example.com.
- The recursive resolver sends a query to the domain’s nameserver.
- The domain nameserver answers with the IP address for example.com
- The DNS resolver then responds to the web browser with the IP address



# Iterative vs Recursive Query

- **Iterative Query**

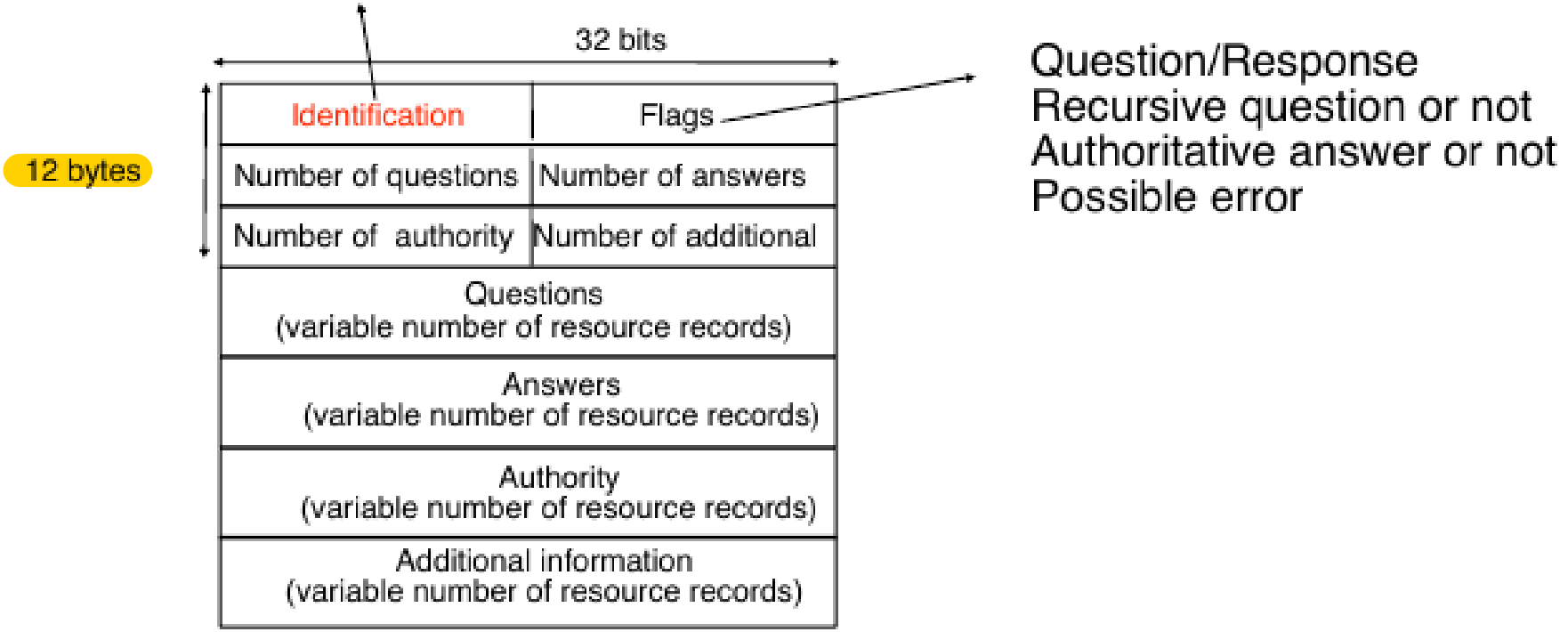
  The resolver sends a query for each domain, until it get to the Authoritative Server (most common)

- **Recursive Query**

  The resolver sends one query to the root server, who will send the query to the TLD server, who will... The answer is then sent back up to the chain to the Root server, who will answer the request of the client.

# DNS Packet

Each DNS request contains a number that will be returned in the response by the server to allow the client to match the request.



# Record types

- The answers are given in the form of a *resource record.* A record is a pair domain-IP. There are different types of records:
  - A record: IPv4 address
  - AAAA record: IPv6 address
  - MX record: mail server to be contacted to send email to this domain
  - NS record: NameServer responsible for this domain
  - ...

# DNS

Reduce risk of failures
- several root-servers
- server DNS servers authoritative for each domain
- each endhost can send queries to multiple resolvers

Improved performance
- avoid sending several times the same query
- cache memory on DNS resolvers containing
  - recent name-addresses translations
  - addresses of DNS servers recently contacted

DNS protocol
- usually runs over UDP
- sometimes is also used over TCP

# DNS problems and solutions

**Problems**

- DNS poisoning
  - *DNSpionage*
- DDOS on DNS servers
  - 2016 Dyn cyberattack
- Use the DNS servers to DDOS others
  - Github, 1,3 TB/sec
- DNS Tunneling

**Solutions**

- DNS-over-TLS
- DNSSEC