16CSCN01: Introduction to Computer Networks

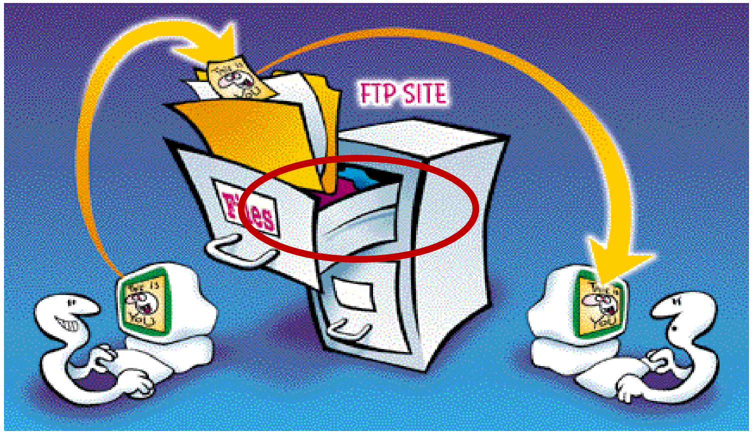
Lecture 4: DNS

Transport Layer

Dr. Amal ElNahas

Application Layer
What Will We Study?

√



√

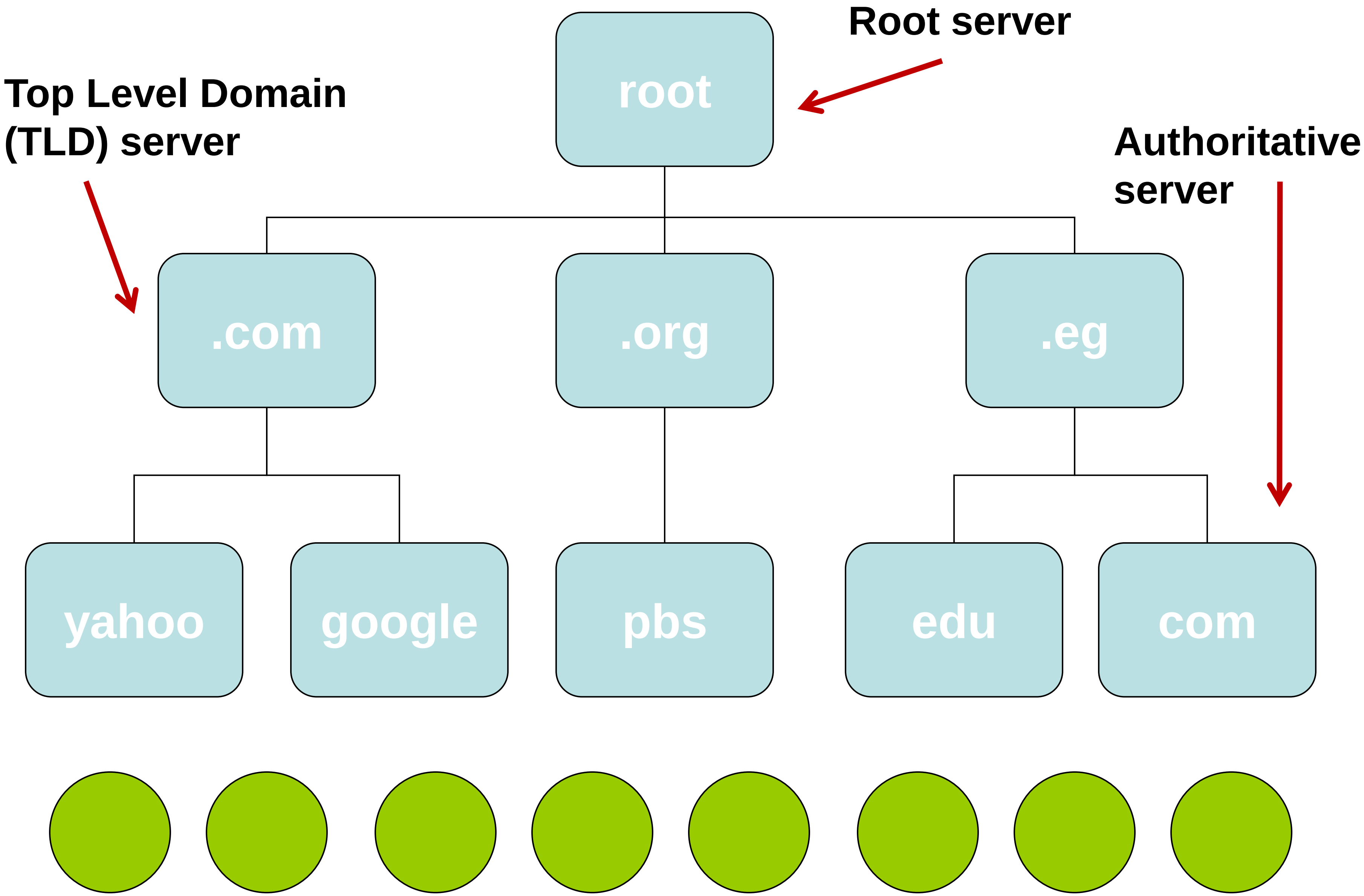
DNS Servers

- **Root DNS server:**
  - There are 13 root DNS servers worldwide
  - When local name server fails, it contacts the root name server
  - Root either knows the mapping (reply directly) or knows IP of an “authoritative” name server that has the mapping

- **Top-Level Domain DNS server:**
  - Responsible for each of the TLD, e.g .com, .edu,.eg,…

- **Authoritative DNS server:**
  - Holds the mapping (names-IP) of all hosts within same organization

- Local DNS server:
  - Doesn’t belong to the DNS hierarchy
  - Acts as a default DNS server (contacted first)



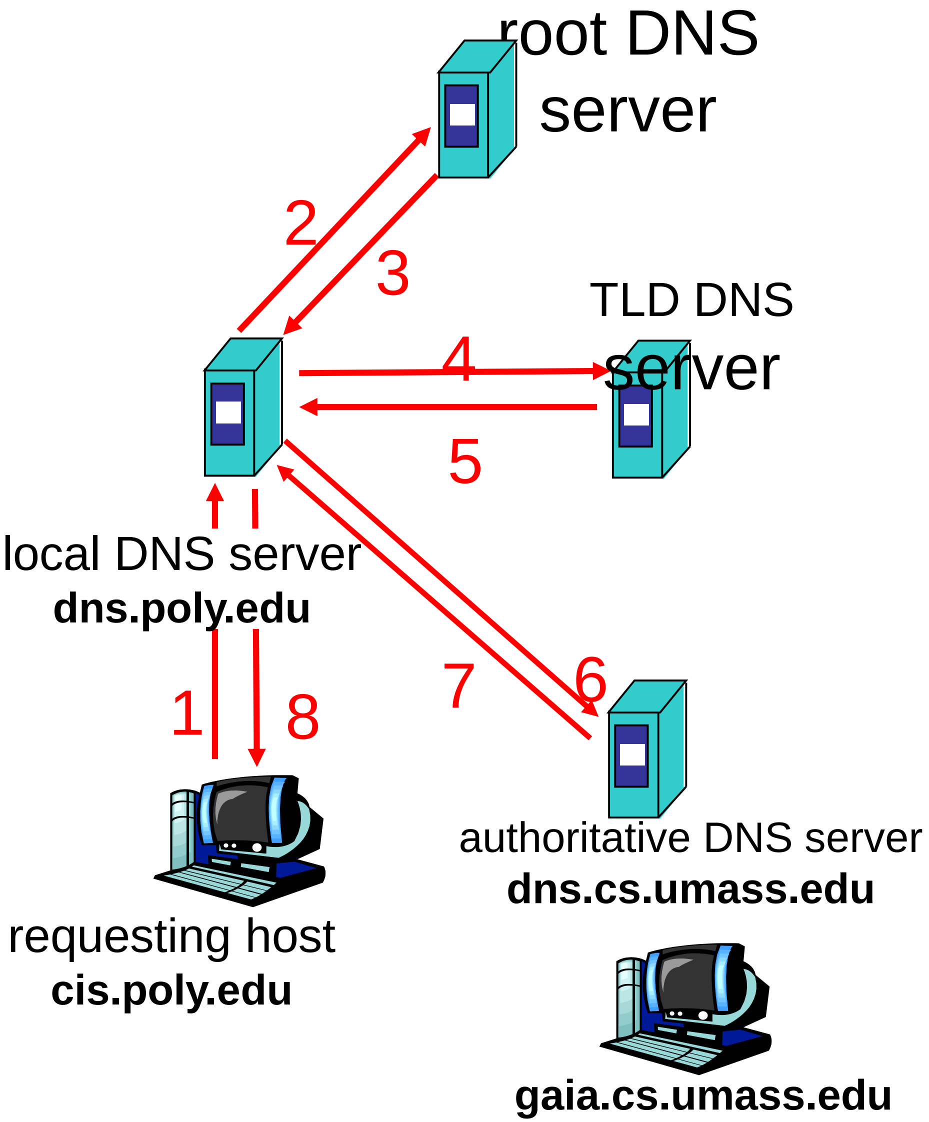
Root server
Top Level Domain (TLD) server
Authoritative server
root
.com
.org
.eg
yahoo
google
pbs
edu
com

DNS Queries

Recursive vs Iterative

DNS Query: Iterative

Host at cis.poly.edu wants IP address for gaia.cs.umass.edu



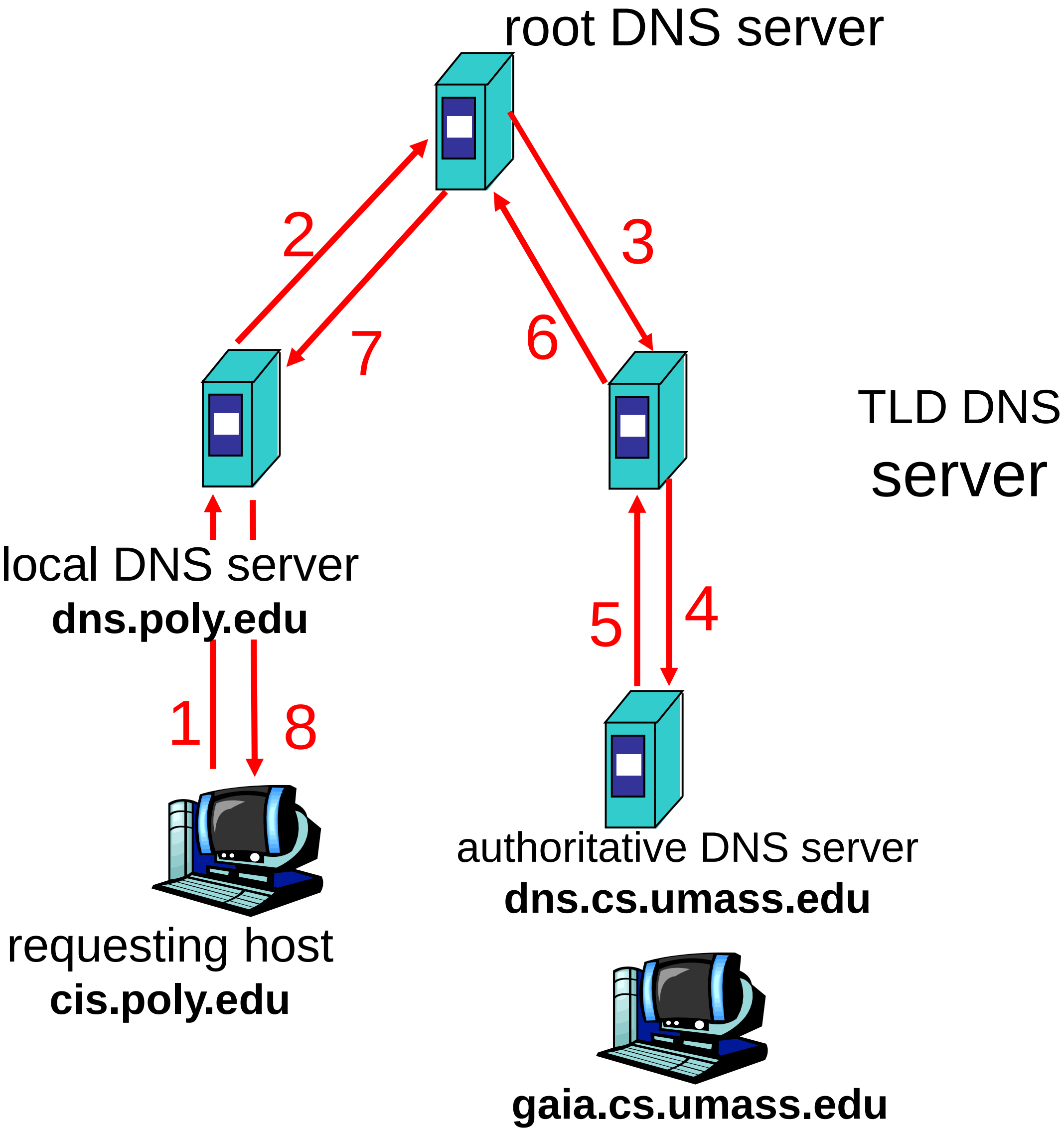
iterated query:

- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"

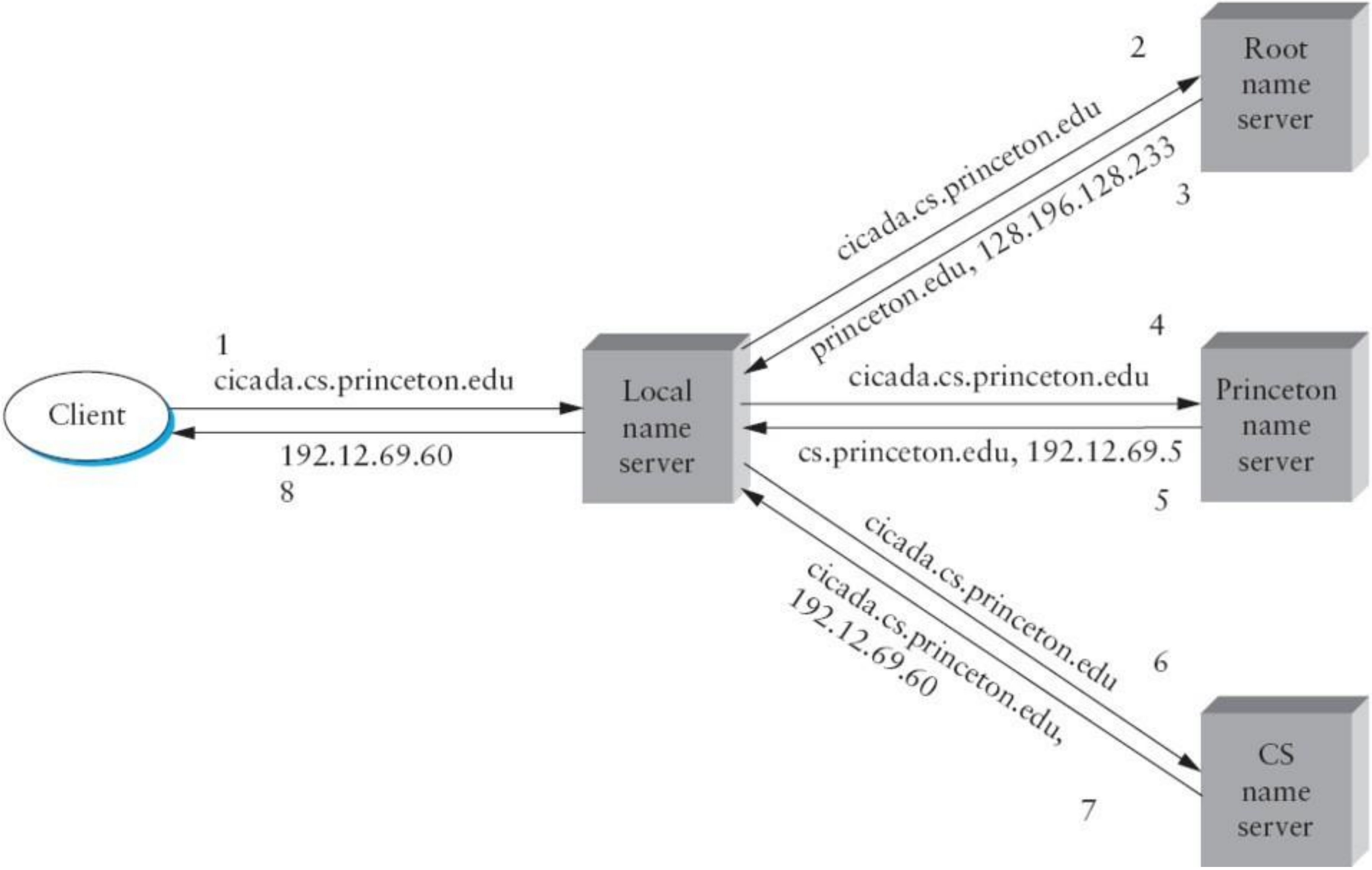
DNS Query: Recursive

recursive query:

- puts burden of name resolution on contacted name server
- heavy load?



How DNS works?



DNS Records

- **Format:**

  (name, value, type, ttl)

- **Types:**

  A, MX, CNAME, NS

DNS Records

- **Type A**

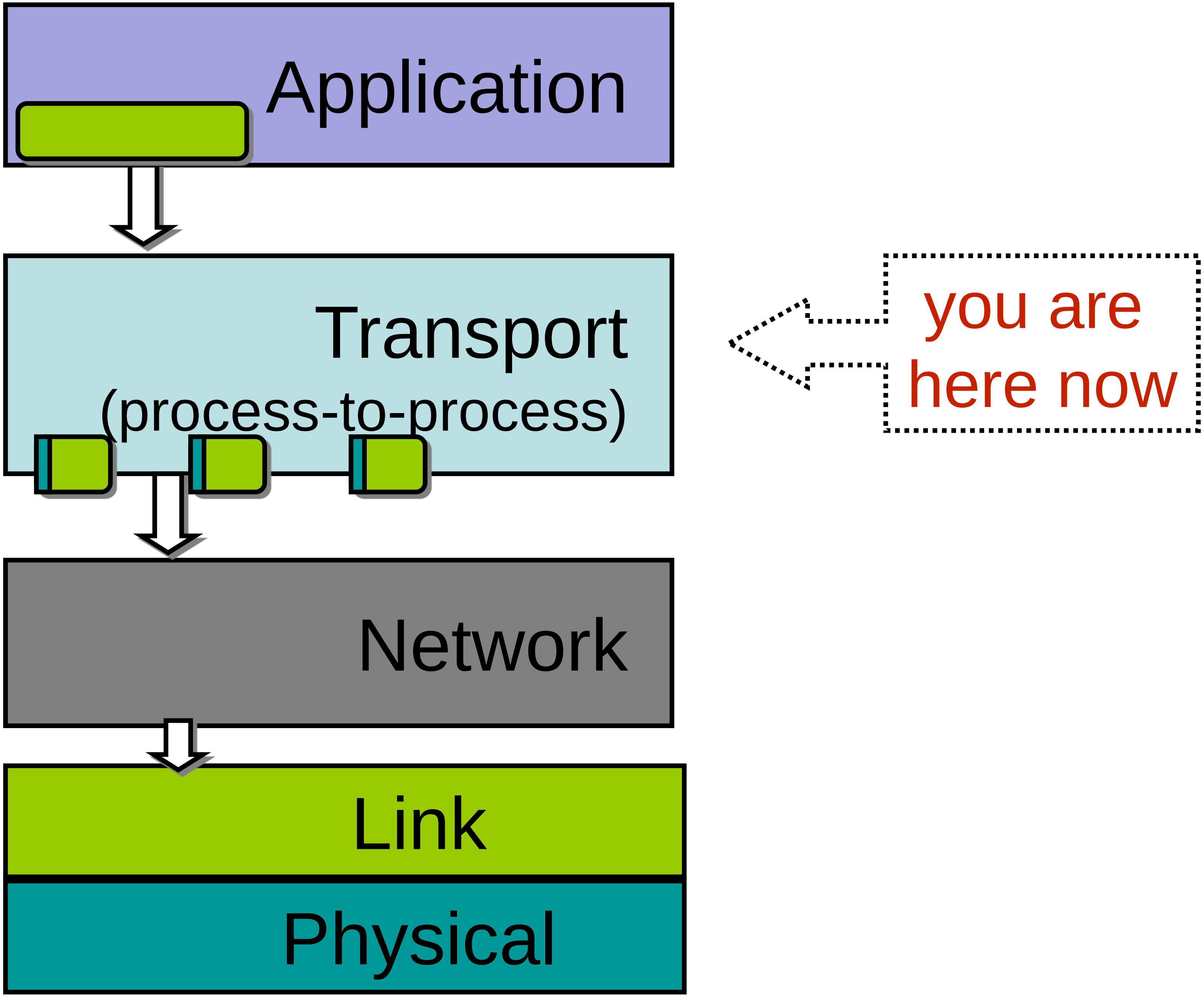
name is hostname, **value** is IP address.

(machine1.foo.com, 145.37.2.126, A)

- **Type=MX**

value is name of mailserver associated with alias name **name**

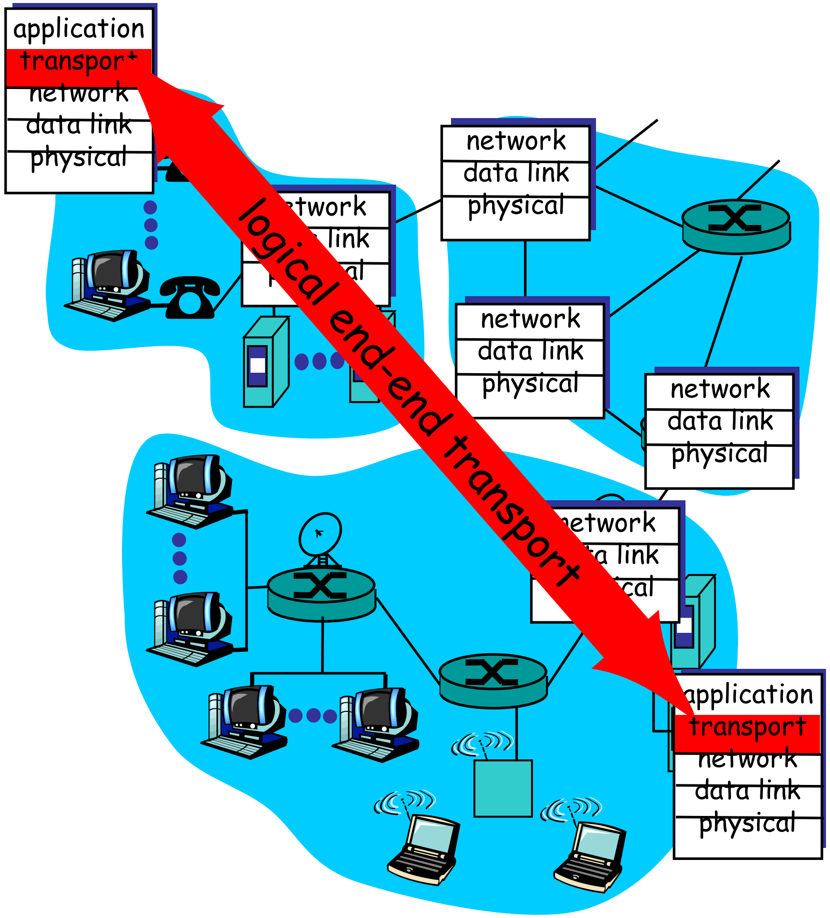
(foo.com, mail.foo.com, MX)


Application
Transport
(process-to-process)
you are here now
Network
Link
Physical

Transport Layer

- Logical communication between applications on 2 different machines (end-to-end protocols)

- Network layer delivers packets to a specific destination host, while transport protocol distinguishes between different destinations (applications) within the same host

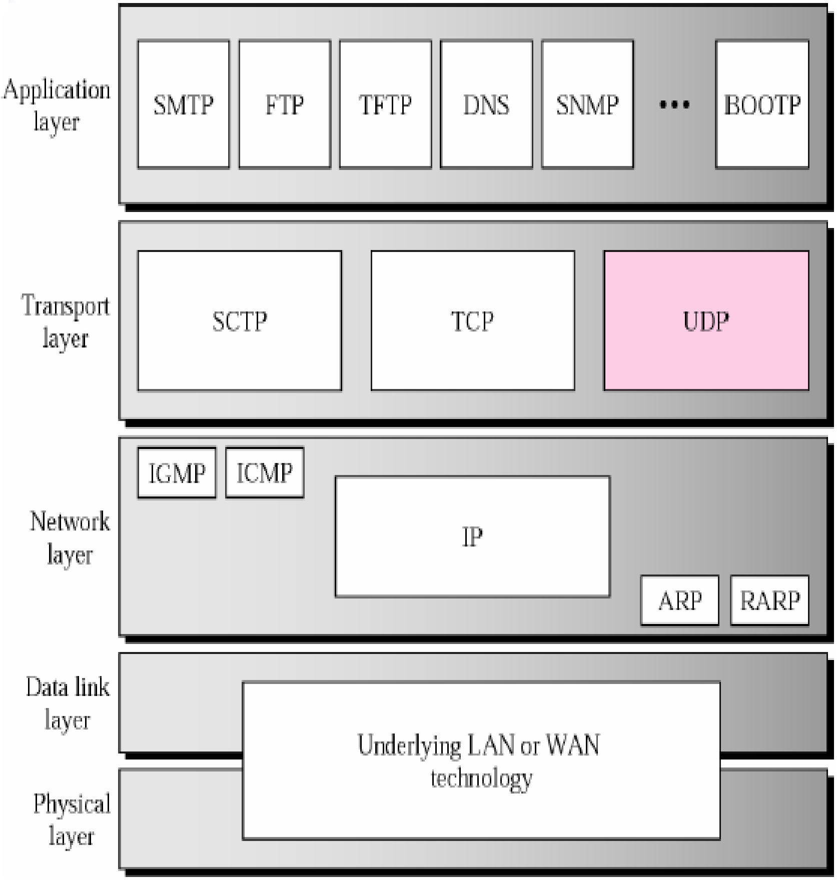


Internet Transport Layer

Internet supports two protocols:

- Transport Control Protocol (TCP)
  - Reliable
  - In order delivery
  - Flow control
  - Congestion control
  - Needs connection set up

- User Datagram Protocol (UDP)
  - Unreliable
  - Unordered delivery
  - Best effort service



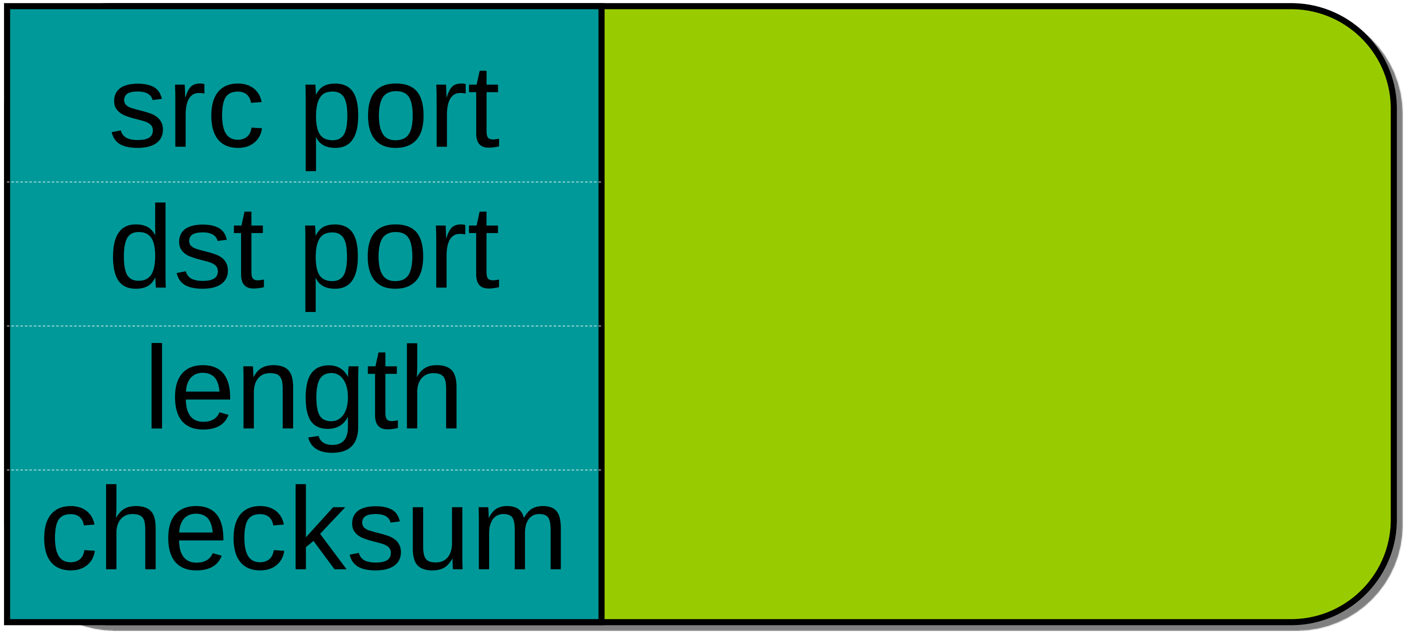
UDP: User Datagram Protocol

- “best effort” service, UDP segments may be:
  - lost
  - delivered out of order to app
- ***connectionless:***
  - No handshaking between UDP sender, receiver
  - Each UDP segment handled independently of others

Why is there a UDP?

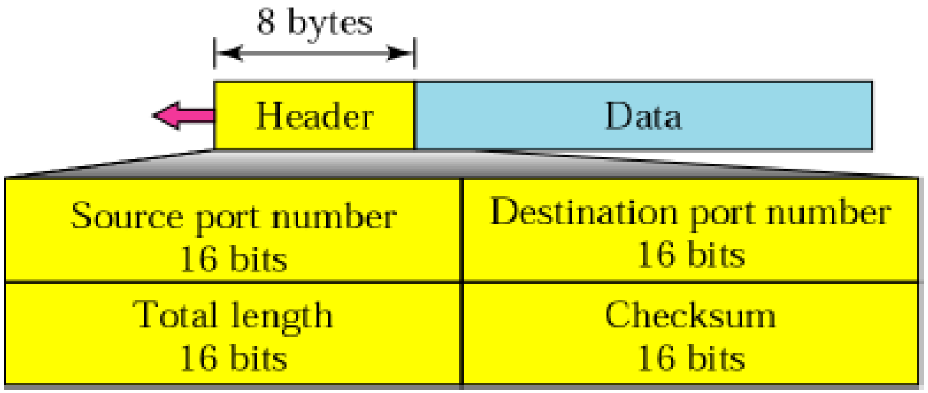
- No connection establishment (which can add delay)
- Simple: no connection state at sender, receiver
- Small segment header
- No congestion control: UDP can blast away as fast as desired
- Best for DNS queries, real-time applications (VoIP,…)

UDP Segment Header



UDP Packet

- **UDP checksum:**
  - Goal: detect “errors” (e.g., flipped bits) in transmitted segment
  - **Sender:**
    - Calculate checksum and adds it to the header
  - **Receiver:**
    - Recalculate the checksum and compare it with the one stored in header.
      - – NO: error detected
      - – YES: no error detected.



Source port number 16 bits	Destination port number 16 bits
Total length 16 bits	Checksum 16 bits