

# CS 640 Introduction to Computer Networks

## Lecture 3

CS 640

---

---

---

---

---

---

---

---

## Today's lecture

- Domain Name System
  - Overview
  - The hierarchy of domain names
  - DNS records
  - The resolution process

CS 640

---

---

---

---

---

---

---

---

## Domain Name System Overview

- What are names used for in general?
  - identify objects
  - locate objects
  - define membership in a group
  - ...
- Basic Terminology
  - *Name space*
    - defines set of possible names
    - consists of a set of name to value *bindings*
  - *Resolution mechanism*
    - when invoked with a name returns corresponding value

CS 640

---

---

---

---

---

---

---

---

## DNS Properties

- **Size of Internet demands robust naming mechanism**
  - Specified in RFC 1034, 1035 (Mockapetris '87)
  - Scalability through caching and hierarchy
  - Reliability through caching and redundancy
- **Names versus addresses**
  - Human readable versus router readable
  - Location transparent versus location-dependent
- **Hierarchical**
  - Names are divided into components
- **Global versus local**
  - What is the scope of naming?

CS 640

---

---

---

---

---

---

---

---

## Examples of Mappings

- **Hosts**  
`pluto.cs.wisc.edu` → `192.12.69.17`  
`192.12.69.17` → `80:23:A8:33:5B:9F`
- **Files**  
`/usr/llp/tmp/foo` → `(server, fileid)`
- **Users**  
`Suman Banerjee` → `suman@cs.wisc.edu`

CS 640

---

---

---

---

---

---

---

---

## Examples (cont)

- **Mailboxes**

```
graph TD
    User[User: user@cs.wisc.edu] --> Mail[Mail program: 192.12.00.5:4]
    Mail <--> NS[Name server: 192.12.00.5:5]
    Mail --> TCP[TCP: 192.12.00.5:5]
    TCP --> IP[IP: 192.12.00.5]
```
- **Services**  
`nearby ps printer with short queue and 2MB`

CS 640

---

---

---

---

---

---

---

---

## Domain Naming System

- Hierarchical name space for Internet objects



- Names are read from right to left separated by periods
  - Each suffix in a domain name is a domain  
`wall.cs.wisc.edu`, `cs.wisc.edu`, `wisc.edu`, `edu`

CS 640

---

---

---

---

---

---

---

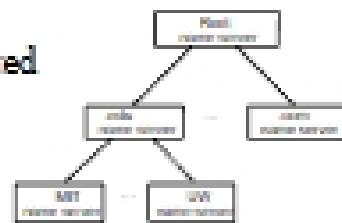
---

## Name Servers

- Partition hierarchy into *zones* (administrative authorities)



- Each zone implemented by two or more *authoritative name servers*



CS 640

---

---

---

---

---

---

---

---

## Resource Records

- Each name server maintains a collection of *resource records* (**Name, Value, Type, Class, TTL**)
  - Each record is a translation based on type
  - Name/Value: not necessarily host names to IP addresses
- Type (some examples)
  - A: Name = full domain name, Value = IP address
  - NS: Value gives domain name for host running name server that knows how to resolve names within specified domain.
  - CNAME: Value gives canonical name for particular host; used to define aliases.
  - MX: Value gives domain name for host running mail server that accepts messages for specified domain.
- Class: allow other entities (other than NIC) to define types
  - IN is what is used by the Internet
- TTL: how long the resource record is valid

CS 640

---

---

---

---

---

---

---

---