# CS 470
# Spring 2019

Mike Lam, Professor


# Naming

Content taken from the following:

*"Distributed Systems: Principles and Paradigms"* by Andrew S. Tanenbaum and Maarten Van Steen (Chapter 4)
Various online sources (including openclipart.org)

# Naming

- "What's in a name?"
  - *"That which we call a .com by any other TLD would load just as quickly."*

*“There are only two hard things in Computer Science: cache invalidation and naming things.”*

*- Phil Karlton (Netscape)*

# Addressing

- Concept of an entity and its name vs. its address
- Some names are true identifiers
  - Each identifier refers to at most one entity
  - Each entity is referred to by at most one identifier
  - Identifiers are never re-used at another time
- Name-to-address binding
  - Name space: domain of all possible names
  - Static vs. dynamic
  - Central vs. decentralized
    - Name server: central host responsible for maintaining bindings

# Naming schemes

```
eb40af8e
c6c1904c
0eceda3e
28dec8ba
4b6683e7
88c9618b
3566223f
38b22b10
```

Flat

```
/
  bin/
    bash
  etc/
    passwd
  usr/
    bin/
      nano
      vim
    lib/
```

Structured

```
(444, Molloy)
(446, Bernstein)
(456, Weikle)
(458, Heydari)
(470, Lam)
```

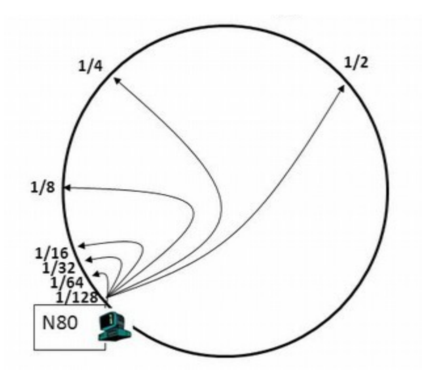
Attribute-based

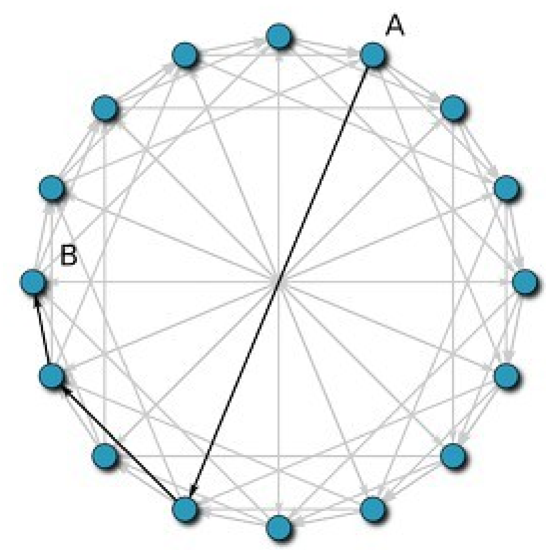
# Flat naming

- Identifiers contain no location information
- Various lookup approaches
  - Broadcast / multicast
  - Forwarding pointers
  - Proximity routing
- Examples: ARP, Chord

# Distributed hash tables

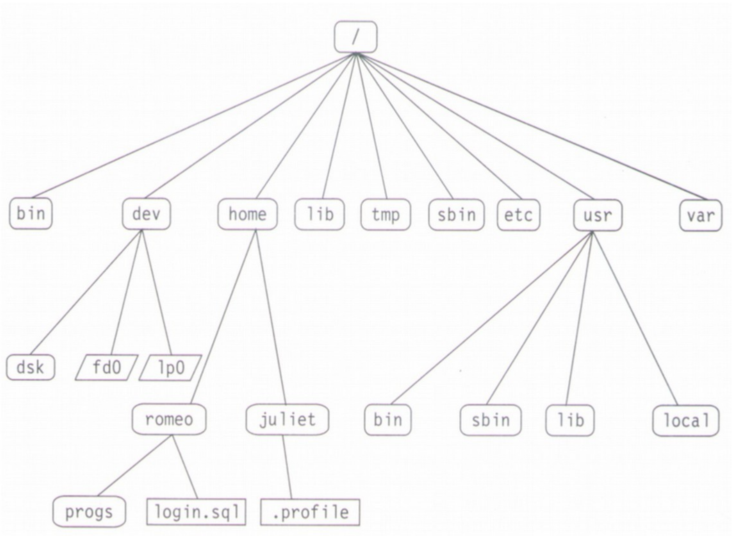
- Chord uses an m-bit identifier space and modulo arithmetic
  - Key k is stored at succ(k), the node with the smallest id ≥ k
- Each node maintains a finger table of forward shortcuts
- To look up k, repeatedly follow lookups in finger table
  - Goal: halve distance to destination every hop



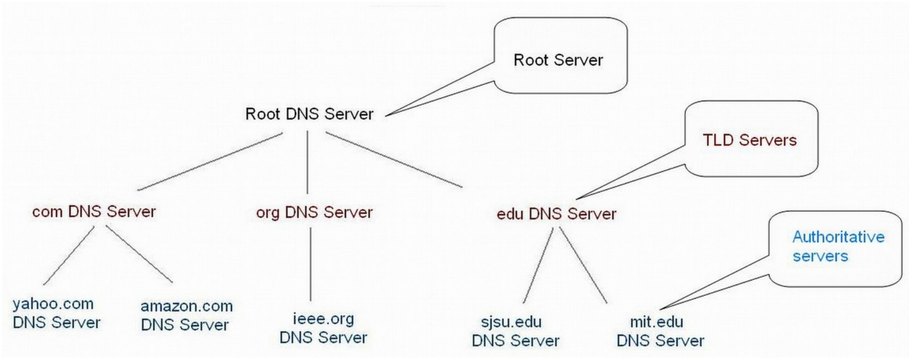



# Structured naming

- Root vs. leaf nodes
- Absolute vs. relative names
  - Global vs. local names
- Iterative vs. recursive resolution
- Linking and aliasing
  - Hard vs. soft (symbolic) links
- Mounting and mount points
- Examples: file systems, DNS, NFS

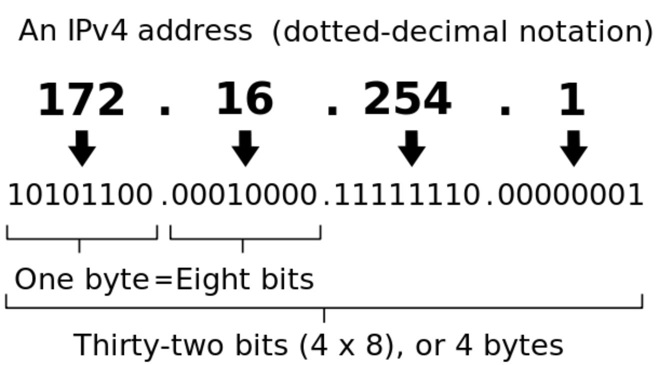



```
Filesystem                          Size  Used Avail Use% Mounted on
/dev/mapper/rhel_login01-root        50G   23G   28G  46% /
/dev/sda6                           497M  206M  292M  42% /boot
nfs.cluster.cs.jmu.edu:/nfs/home    100G  4.6G   96G   5% /nfs/home
nfs.cluster.cs.jmu.edu:/nfs/scratch 2.0T  862G  1.2T  43% /scratch
```



# IPv4

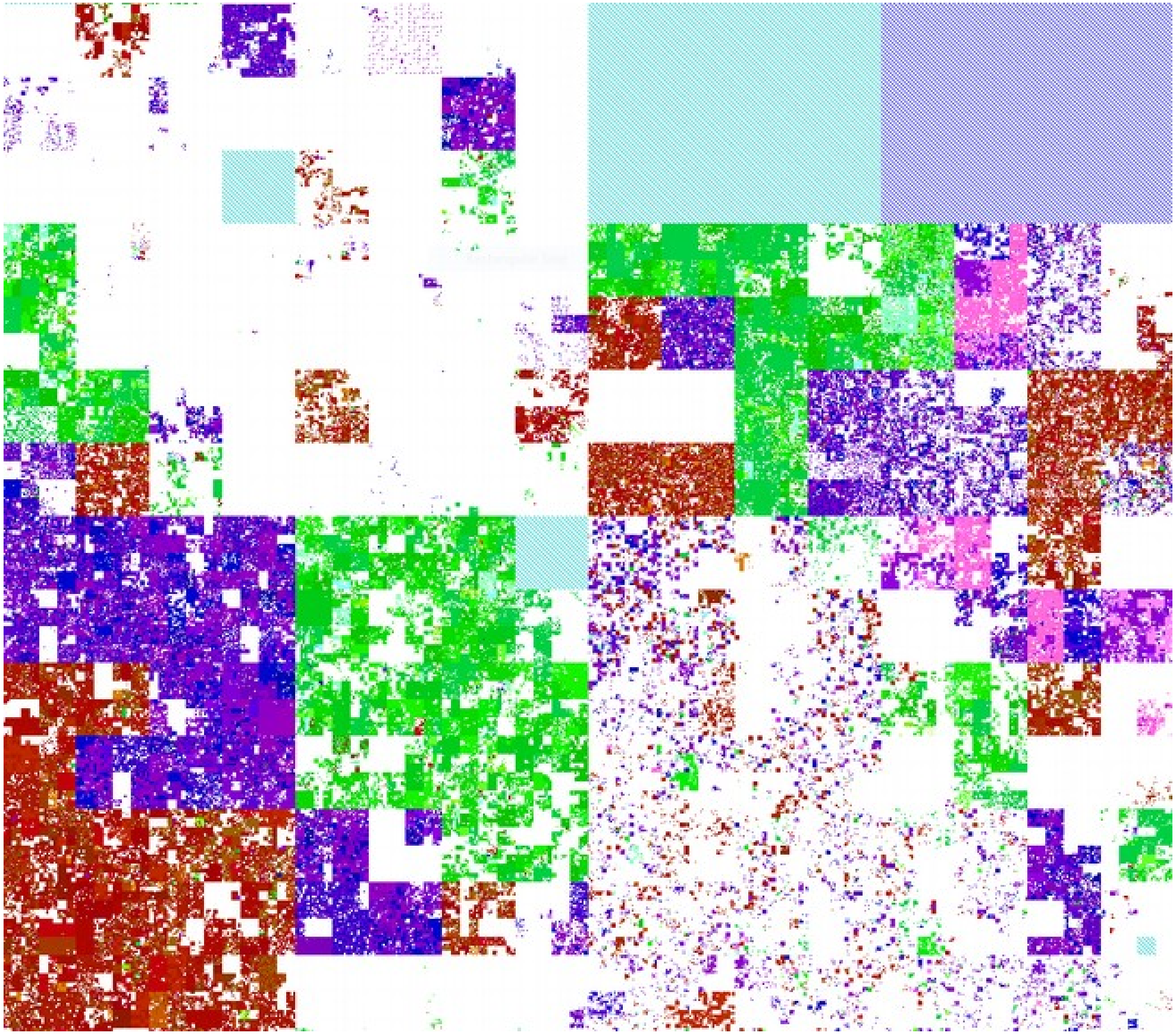
- IPv4: 32 bits - four octets w/ CIDR notation (/8, /16, etc.)
  - Classful addressing: Class A, Class B, Class C
  - IETF and IANA allocate addresses (32 bits - 4 billion total addresses)
  - Published in 1981; now nearly exhausted
- Notable networks
  - Private (10.0.0.0/8)
  - Loopback (127.0.0.0/8)
  - JMU (134.126.0.0/16)
  - Private (192.168.0.0/16)




from https://en.wikipedia.org/wiki/IPv4


https://xkcd.com/195/

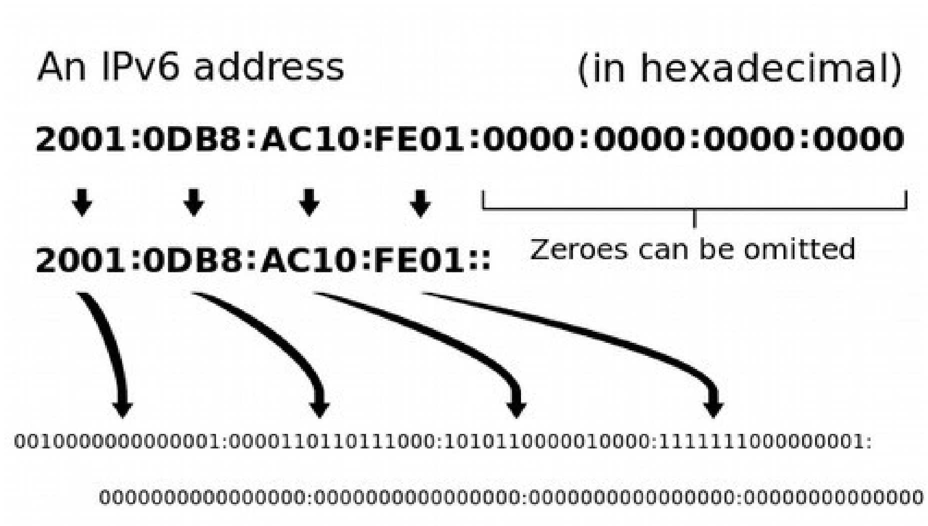
# IPv4 map



from `https://ant.isi.edu/address/browse/index.html`

# IPv6

- IPv6 published in 1998
  - 128 bits - $3.4 \times 10^{38}$ total addresses
  - Eight groups of 16 bits (4 hex chars)
  - 64-bit routing prefix, 64-bit host/interface identifier
  - Slow uptake due to migration complications




from `https://en.wikipedia.org/wiki/IPv6`

# IPv4 vs. IPv6

- The IPv6 name space is far larger than you think!
  - In fact, there is NO WAY to draw the two address spaces to scale. If IPv4 were a 1.6-inch square, IPv6 would be a square the size of the solar system!
  - $2^{128} \approx 10^{38} \gg$ the number of drops of water in all the world's oceans ($10^{25}$) or the number of stars in the observable universe ($10^{23}$)
  - "If we had been assigning IPv6 addresses at a rate of 1 billion per second since the earth was formed, we would have by now used up less than one trillionth of the address space."
  - "We could assign an IPv6 address to every atom on the surface of the earth – and have enough addresses left over for another hundred earths."

Sources:

- `http://waitbutwhy.com/2014/11/1000000-grahams-number.html`
- `http://www.tcpipguide.com/free/t_IPv6AddressSizeandAddressSpace-2.htm`
- `http://www.brucebnews.com/2010/10/ipv6-and-really-large-numbers/`

# Attribute-based naming

- Human-friendly resource identifiers
- Storage of (key, value) pairs
- Often implemented with distributed hash tables
  - Centralized vs. decentralized lookups
  - You will implement this in P4!
- Semantic overlay networks
  - Nodes maintain links to "semantically proximate" nodes
  - Most useful in distributed peer-to-peer networks
  - Exploit small-world effect