CSE 224: NETWORKING FUNDAMENTALS AND DNS

George Porter
Jan 6, 2022

UC San Diego

ATTRIBUTION

UC San Diego

AGENDA

Today:

- Chapter 2 (DNS), 3, and first few pages of 4 of “Network Programming with Go”

Next week:

- In class demo/exercises focused on Go’s “net” package

TODO

1. Project 1 due Tuesday Jan 11 at 5pm

UC San Diego

A REQUEST


Bernie
I am once again asking
to please stop emailing me about the wait list and class size. It is out of my control!
imgflip.com

Outline

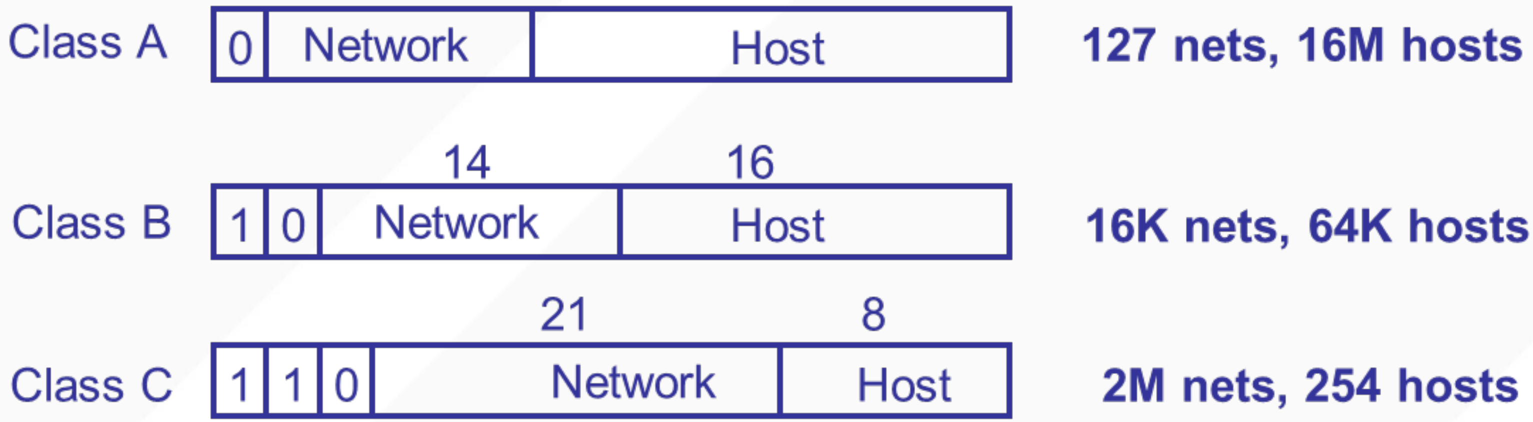
1. ~~Performance~~
2. ~~Layering~~
3. Addressing
4. DNS

IP VERSION 4 (IPV4)

11000000	.	10101000	.	00000001	.	00001010	(Binary)
192	.	168	.	1	.	10	(Decimal)

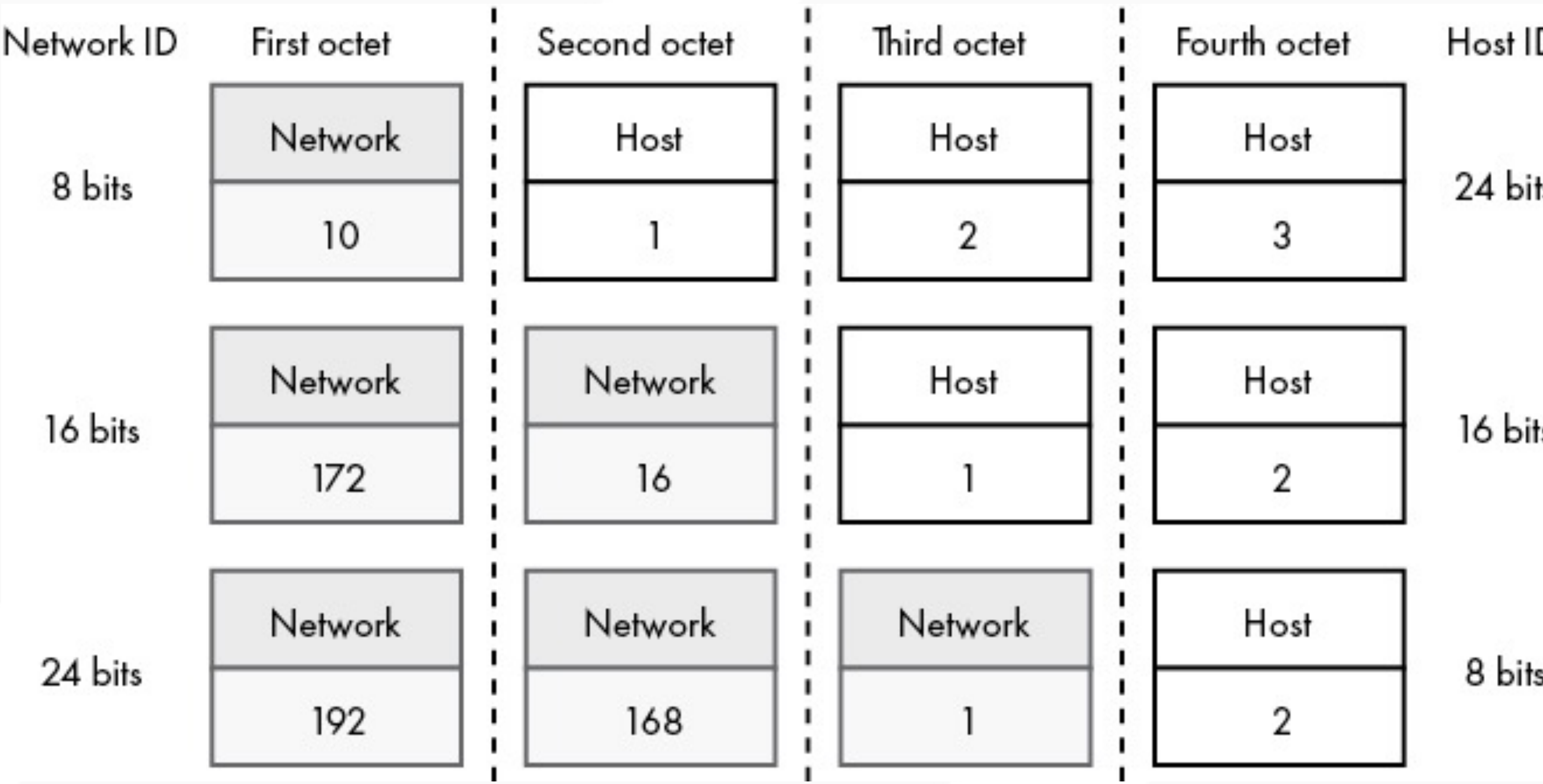
CLASS-BASED ADDRESSING (NOT REALLY USED ANYMORE)

- Most significant bits determines "class" of address

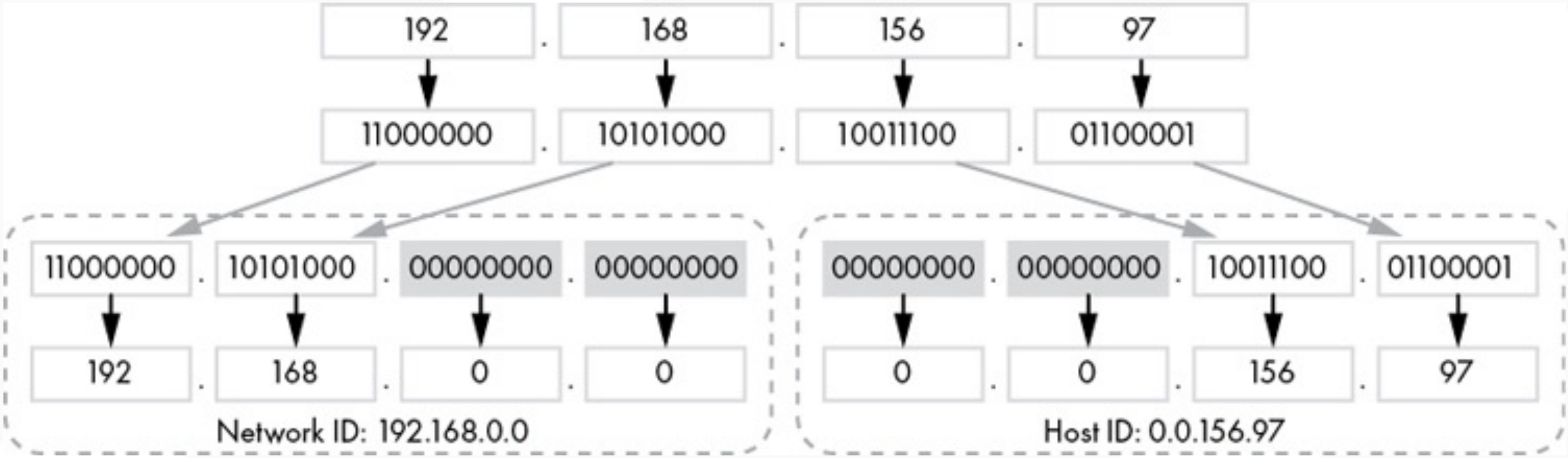


- Special addresses
  - Class D (1110) for multicast, Class E (1111) experimental
  - 127.0.0.1: local host (a.k.a. the loopback address)
  - Host bits all set to 0: network address
  - Host bits all set to 1: broadcast address

CLASS-BASED ADDRESSING EXAMPLES

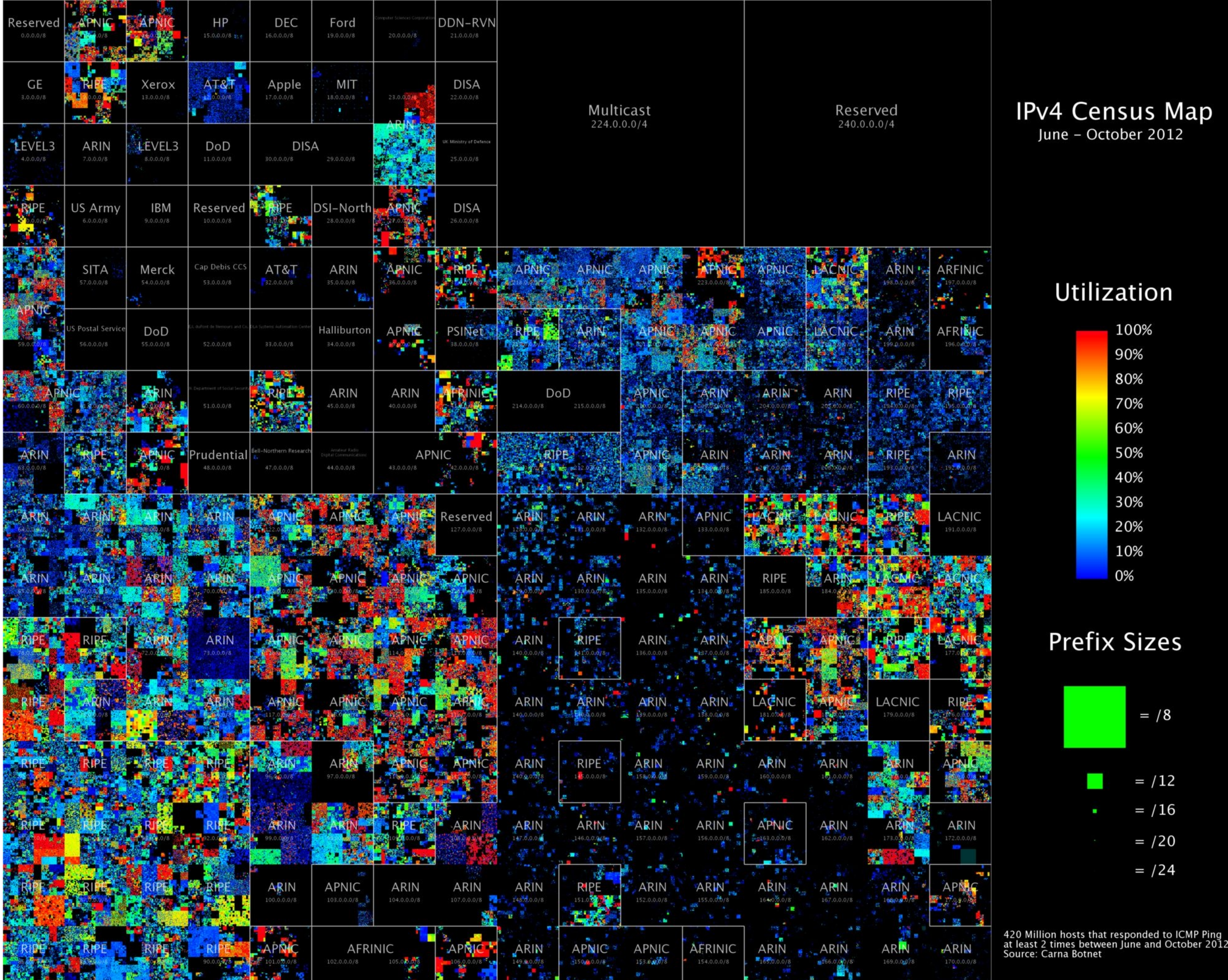


ADDRESSING EXAMPLE



IP ADDRESS PROBLEM (1991)

- Address space depletion
  - In danger of running out of classes A and B
- Why?
  - Class C too small for most organizations (only ~250 addresses)
  - Very few class A – very careful about giving them out (who has 16M hosts anyway?)
  - Class B – greatest problem



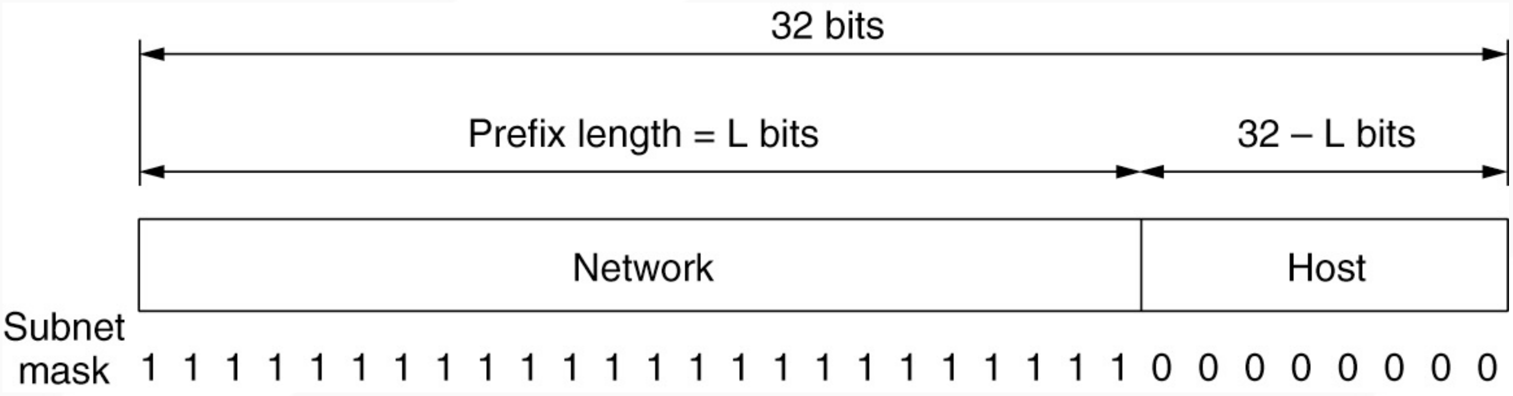
IPv4 Census Map
June – October 2012
Multicast
224.0.0.0/4
Reserved
240.0.0.0/4
Utilization
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Prefix Sizes
= /8
= /12
= /16
= /20
= /24
420 Million hosts that responded to ICMP Ping
at least 2 times between June and October 2012
Source: Carna Botnet

CIDR

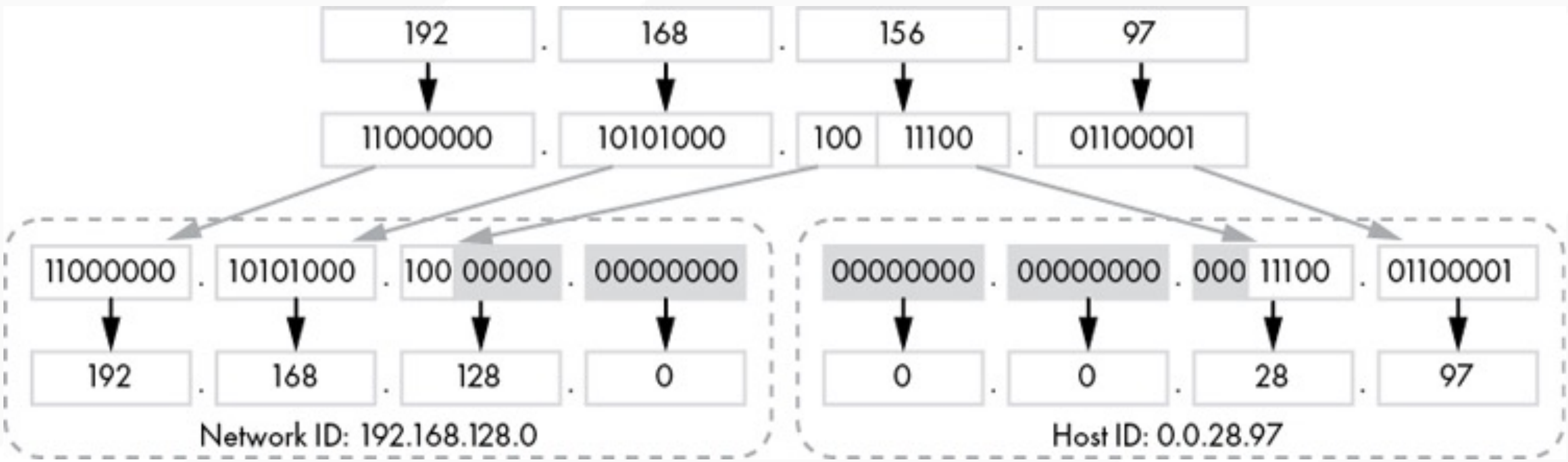
- Classless Inter-Domain Routing (1993)
  - Networks described by variable-length prefix and length
  - Allows arbitrary allocation between network and host address

  - e.g. 10.95.1.2 contained within 10.0.0.0/8:
    - 10.0.0.0 is network and remainder (95.1.2) is host
- Pro: Finer grained allocation; aggregation
- Con: More expensive lookup: longest prefix match

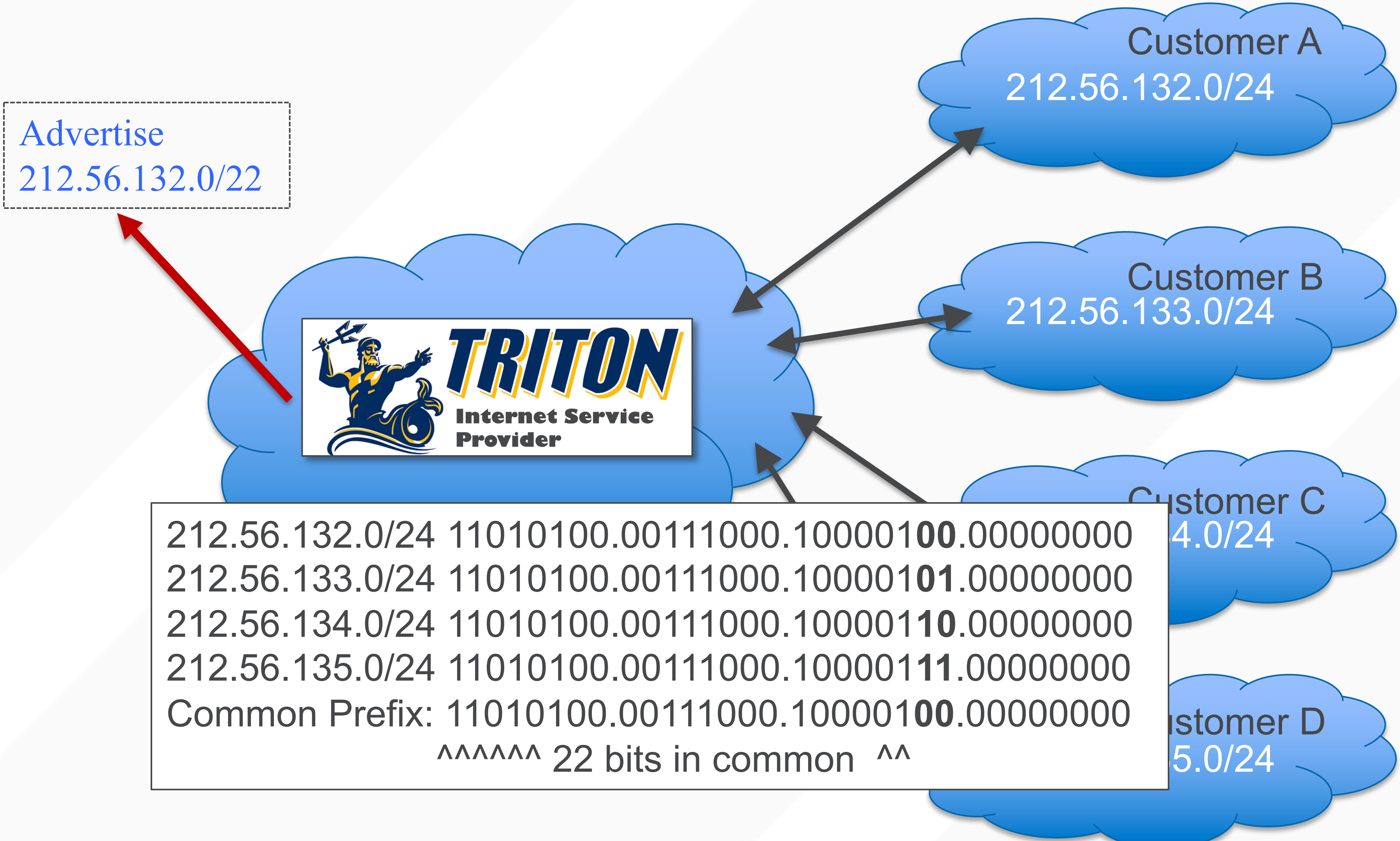
SUBNETS AND NETMASKS



192.168.156.97/19

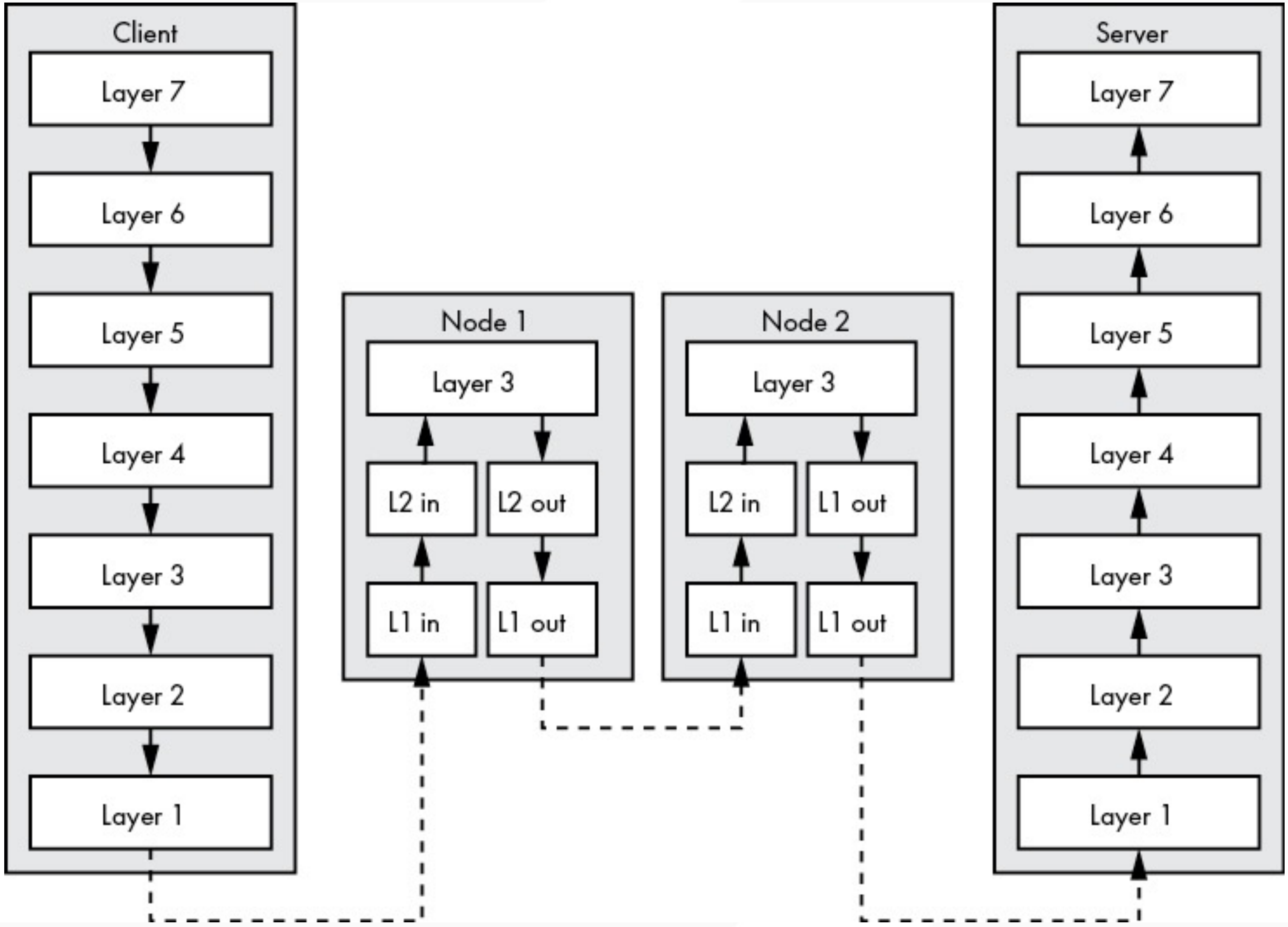


ADDRESS AGGREGATION EXAMPLE



212.56.132.0/24 11010100.00111000.100001**00**.00000000
212.56.133.0/24 11010100.00111000.100001**01**.00000000
212.56.134.0/24 11010100.00111000.100001**10**.00000000
212.56.135.0/24 11010100.00111000.100001**11**.00000000
Common Prefix: 11010100.00111000.100001**00**.00000000
^^^^^^ 22 bits in common ^^

ROUTING

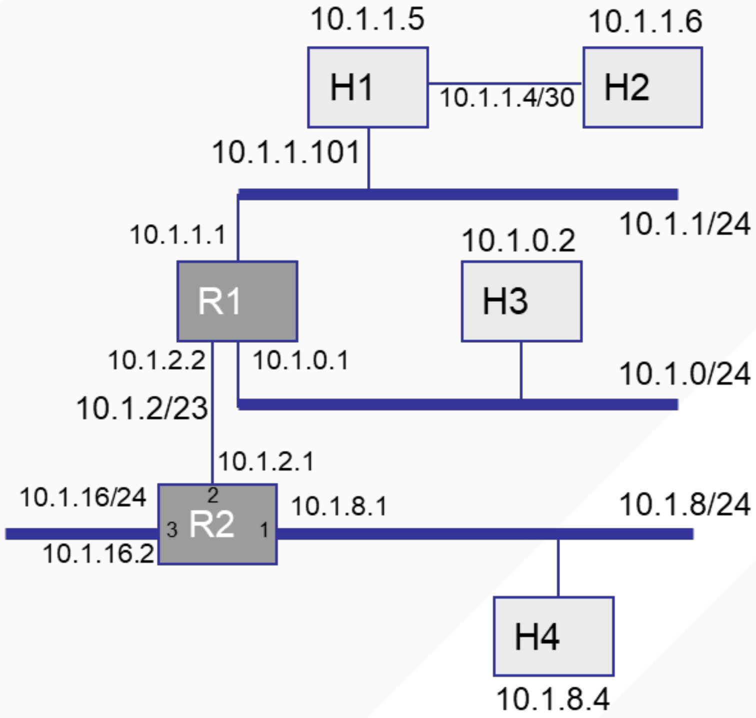


Client
Layer 7
Layer 6
Layer 5
Layer 4
Layer 3
Layer 2
Layer 1
Node 1
Layer 3
L2 in
L2 out
L1 in
L1 out
Node 2
Layer 3
L2 in
L1 out
L1 in
L1 out
Server
Layer 7
Layer 6
Layer 5
Layer 4
Layer 3
Layer 2
Layer 1

ROUTING TABLE EXAMPLE (R2)

- Packet to 10.1.1.6
- Matches 10.1.0.0/23

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.16.1
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.0.0/23	10.1.2.2
10.1.16.0/24	interface3

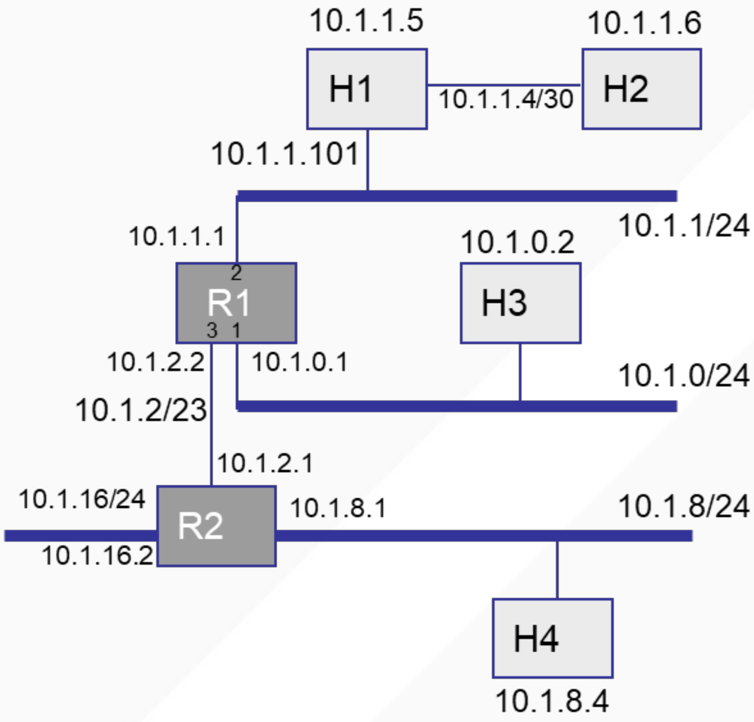


ROUTING TABLE EXAMPLE 2 (R1)

- Packet to 10.1.1.6
- Matches 10.1.1.4/30
  - Longest prefix match

Routing table at R1

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.2.1
10.1.0.0/24	interface1
10.1.1.0/24	interface2
10.1.2.0/23	interface3
10.1.1.4/30	10.1.1.101

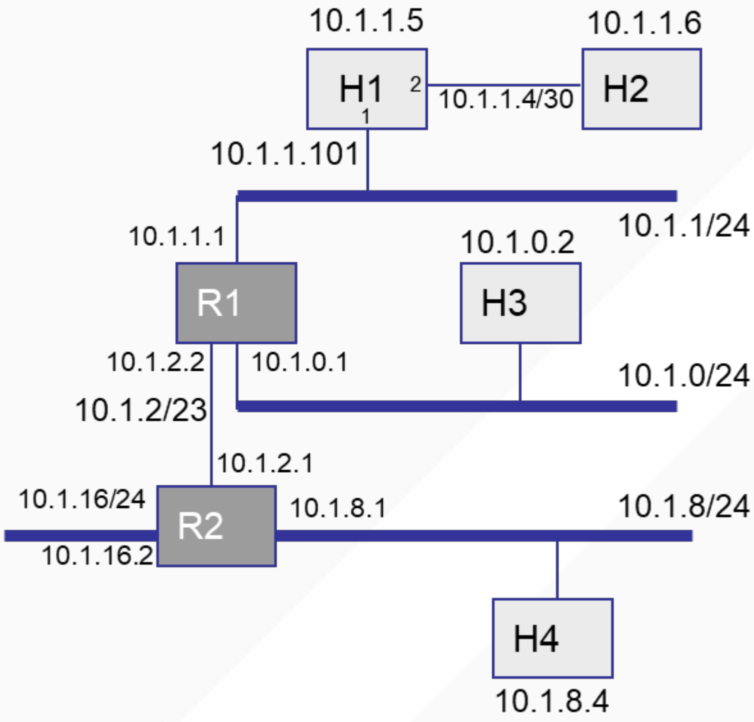


ROUTING TABLE EXAMPLE 3 (H1)

- Packet to 10.1.1.6
- Direct route
  - Longest prefix match

Routing table at H1

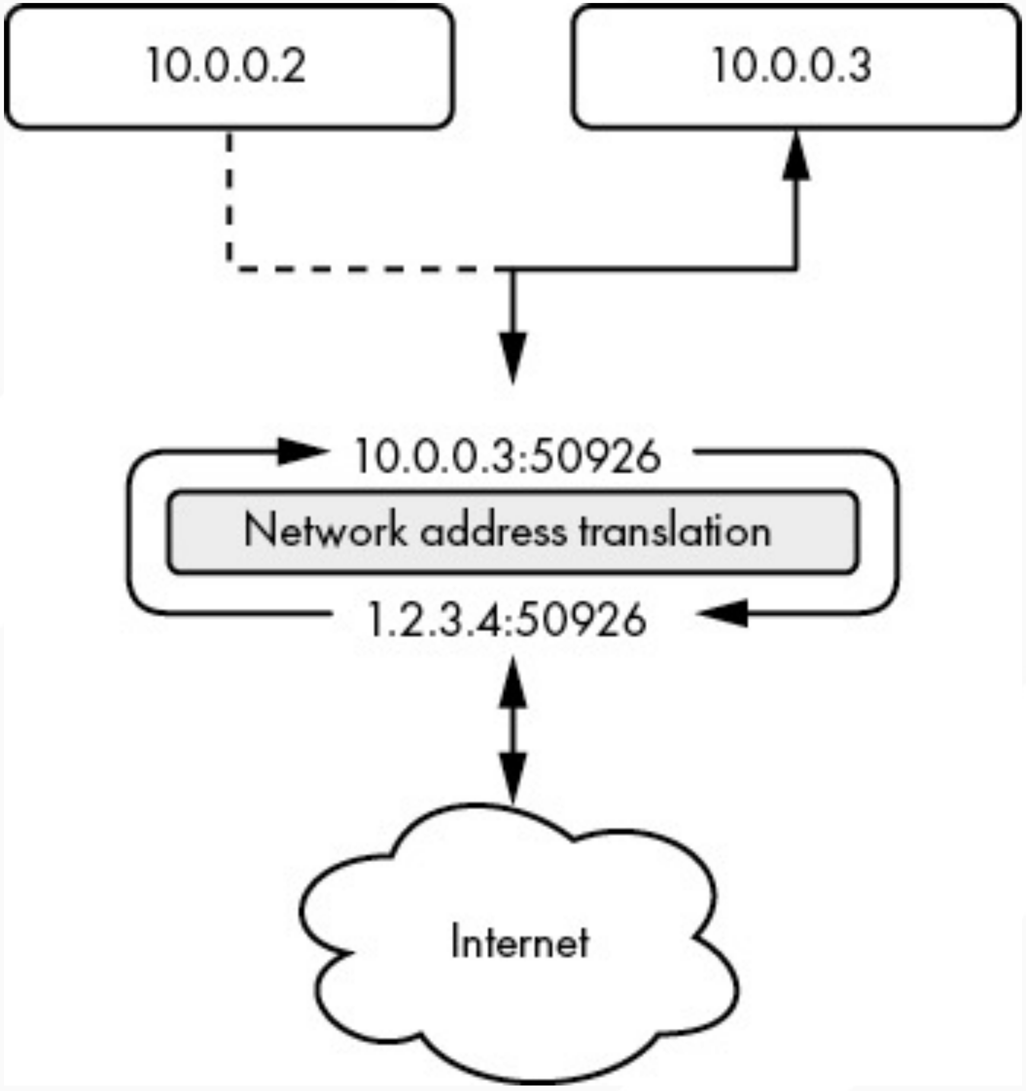
Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.1.1
10.1.1.0/24	interface1
10.1.1.4/30	interface2



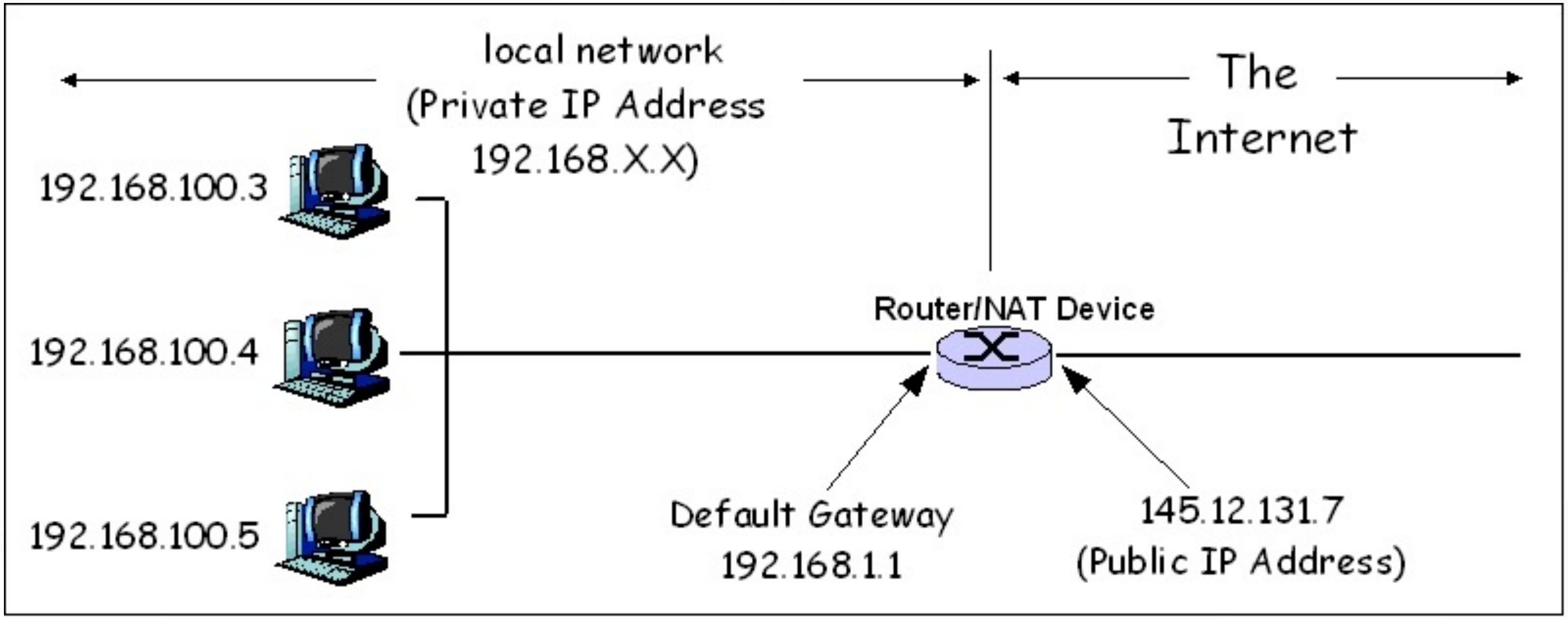
PORTS

- IP addresses identify a *machine*
  - Actually they identify a network interface on a machine
- How to identify different programs on the machine?
  - Process ID/PID? (no… why not?)
  - Instead we use a port (which is a 16-bit number)
  - 0-1024 reserved for the OS, you can use 1025-65535

NETWORK ADDRESS TRANSLATION

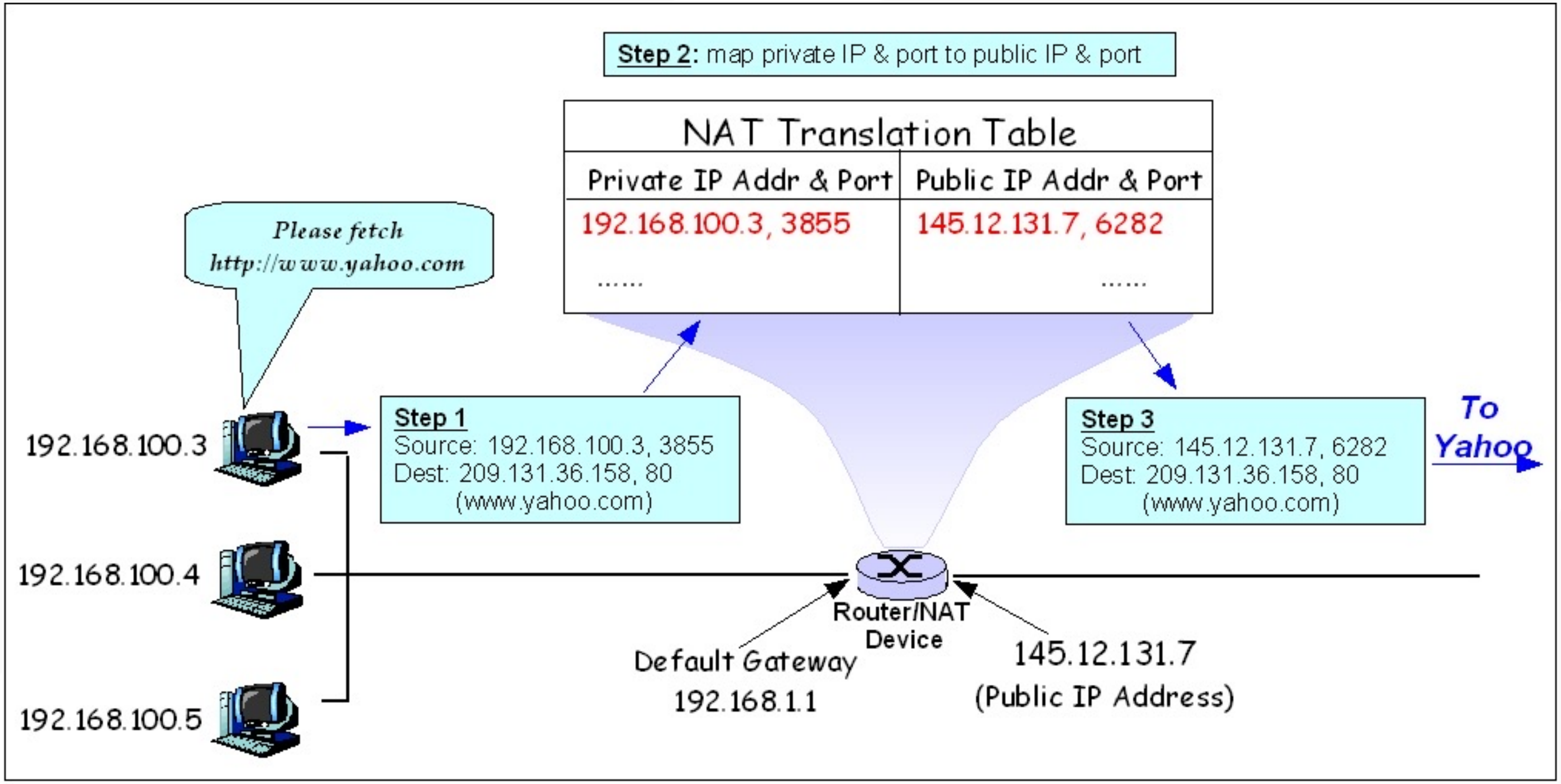


LOAD BALANCING VIA NETWORK ADDRESS TRANSLATION



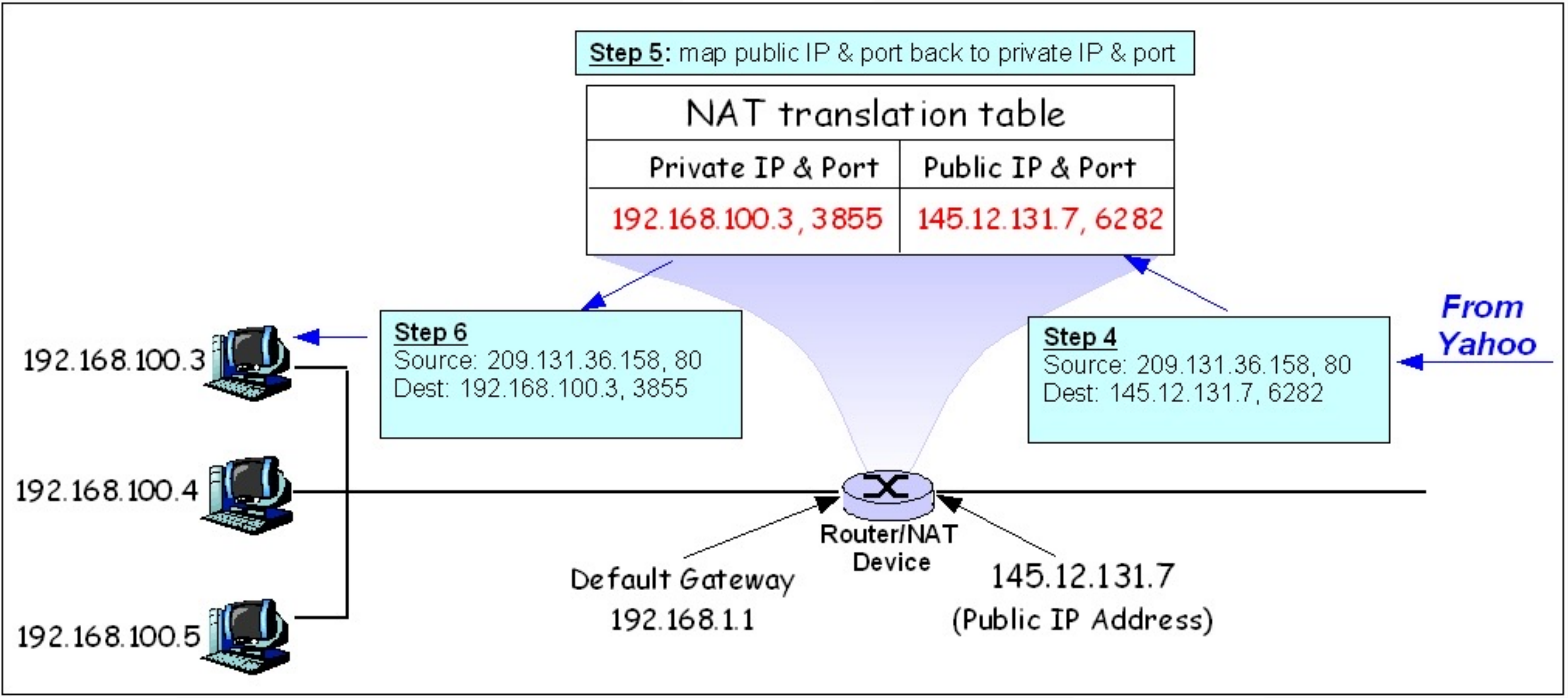
Courtesy of: https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols

LOAD BALANCING VIA NETWORK ADDRESS TRANSLATION



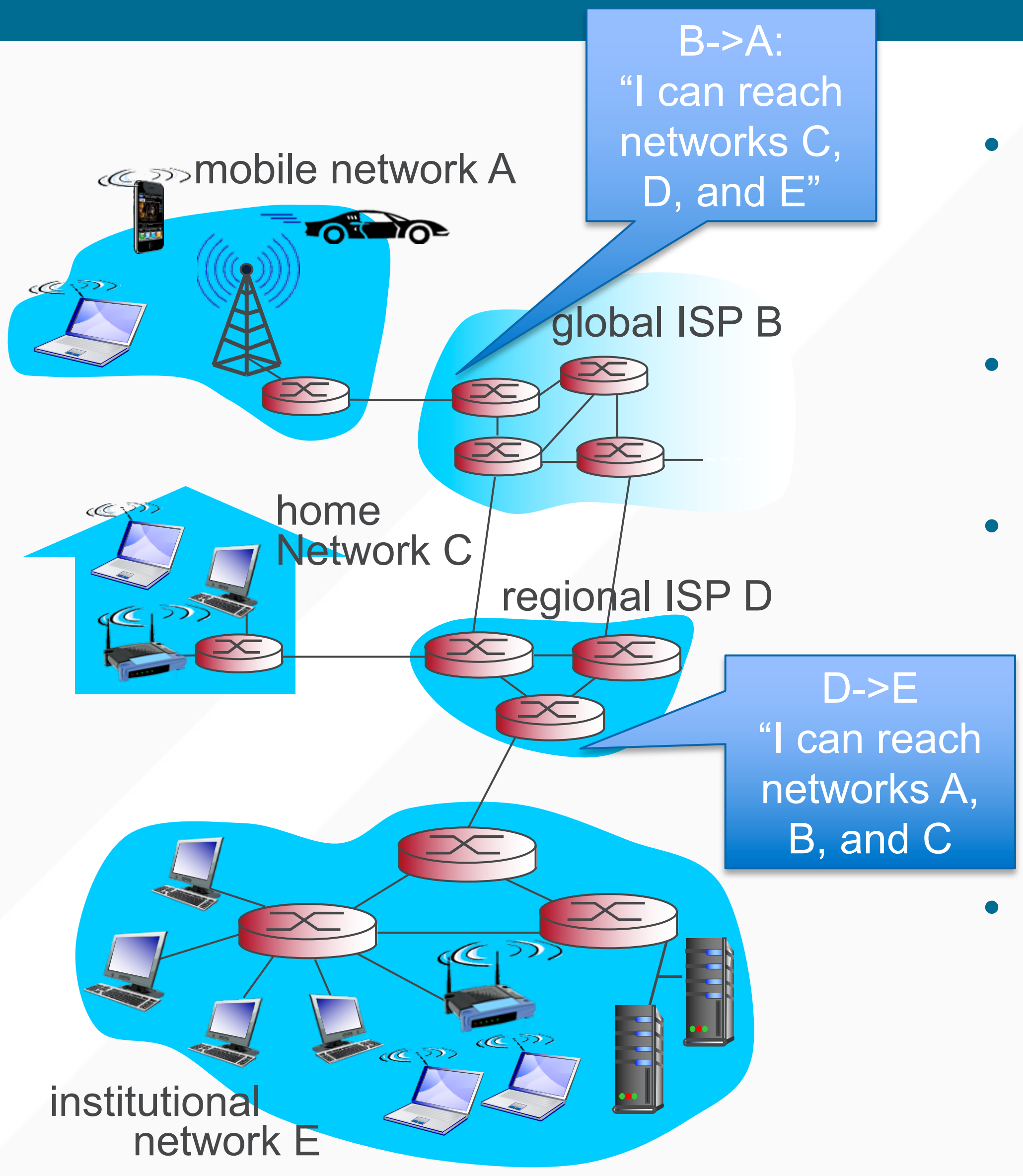
Courtesy of: https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols

LOAD BALANCING VIA NETWORK ADDRESS TRANSLATION



Courtesy of: https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols

ROUTING PACKETS BETWEEN NETWORKS



- Networks use *Border Gateway Protocol (BGP)* to announce reachability
- Each network talks just with its neighbors
- Goal is to get a packet to the destination network
  - It is up to that destination network to get individual packets to their ultimate destination
- Back-to-back packets from the same "connection" might take different paths!
  - Might arrive out of order too

Outline

1. ~~Performance~~
2. ~~Layering~~
3. ~~Addressing~~
4. DNS

DNS HOSTNAME VERSUS IP ADDRESS

- **DNS host name** (e.g. www.cs.ucsd.edu)
  - **Mnemonic** name appreciated by humans
  - **Variable length**, full alphabet of characters
  - Provides **little** (if any) information about **location**
- **IP address** (e.g. 128.112.136.35)
  - Numerical address appreciated by **routers**
  - **Fixed length**, decimal number
  - **Hierarchical** address space, related to host **location**

MANY USES OF DNS

- Hostname to IP address translation
  - IP address to hostname translation (reverse lookup)
- Host name aliasing: other DNS names for a host
  - Alias host names point to canonical hostname
- **Email**: Lookup domain’s mail server by domain name

ORIGINAL DESIGN OF DNS

- Per-host file named /etc/hosts (1982)
  - Flat namespace: each line = IP address & DNS name
  - SRI (Menlo Park, California) kept the master copy
  - Everyone else downloads regularly
- *But, a single server doesn't scale*
  - Traffic implosion (lookups and updates)
  - Single point of failure
- Need a distributed, hierarchical **collection** of servers

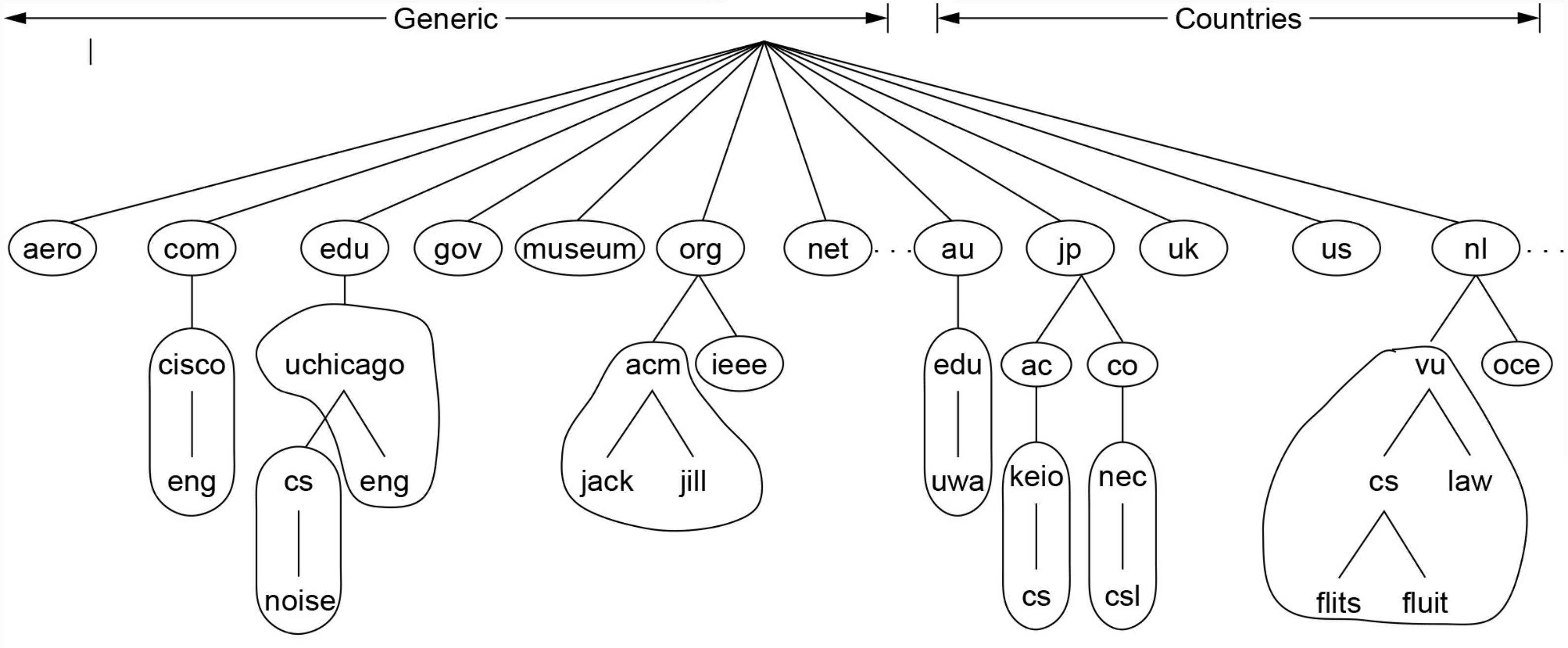
DNS: GOALS AND NON-GOALS

- A wide-area **distributed database**
- Goals:
  - **Scalability**; decentralized maintenance
  - **Robustness**
  - Global scope
    - Names mean the same thing everywhere
  - Distributed updates/queries
  - Good **performance**

DOMAIN NAME SYSTEM (DNS)

- Hierarchical name space divided into contiguous sections called zones
  - Zones are distributed over a collection of DNS servers
- Hierarchy of DNS servers:
  - Root servers (identity hardwired into other servers)
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Performing the translations:
  - Local DNS servers located near clients
  - Resolver software running on clients

DNS IS HIERARCHICAL



- Hierarchy of namespace matches hierarchy of servers
- Set of nameservers answers queries for names within zone
- Nameservers store names and links to other servers in tree

DNS ROOT NAMESERVERS

- 13 root servers

TLD AND AUTHORITATIVE SERVERS

- ftp://ftp.internic.net/domain/named.root
- Top-level domain (TLD) servers
  - Responsible for com, org, net, edu, etc, and all top-level country domains: uk, fr, ca, jp
  - Network Solutions maintains servers for com TLD
  - Educause non-profit for edu TLD
- Authoritative DNS servers
  - An organization's DNS servers, providing authoritative information for that organization
  - May be maintained by organization itself, or ISP

COMMON TLDS

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

LOCAL NAME SERVERS

- Do not strictly belong to hierarchy
- Each ISP (or company, or university) has one
  - Also called default or caching name server
- When host makes DNS query, query is sent to its local DNS server
  - Acts as proxy, forwards query into hierarchy
  - Does work for the client

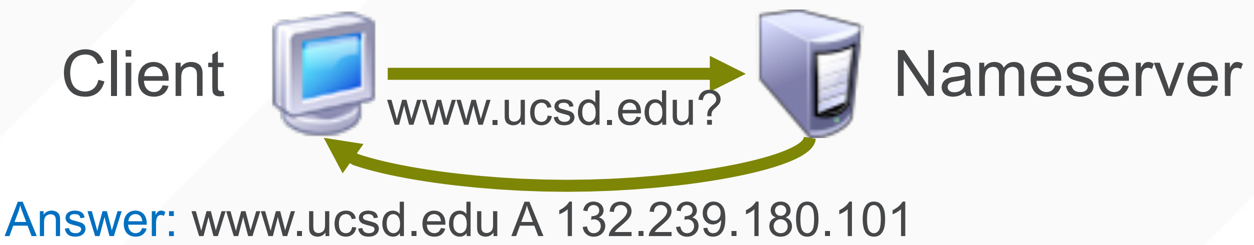
DNS RESOURCE RECORDS

- DNS is a distributed database storing **resource records**
- Resource record includes: (**name**, type, **value**, time-to-live)

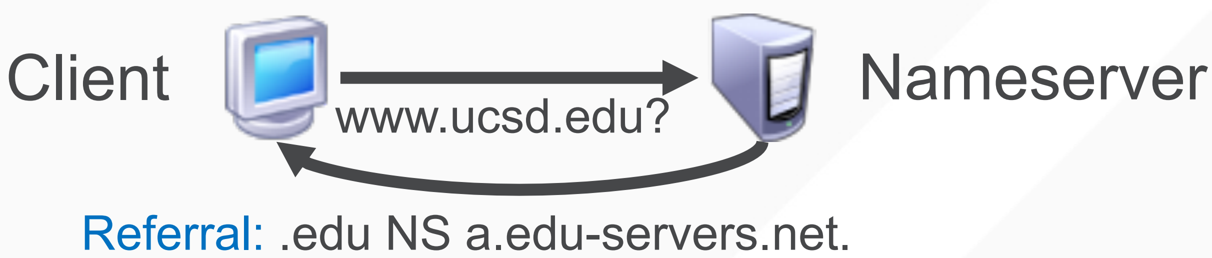
Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

DNS IN OPERATION

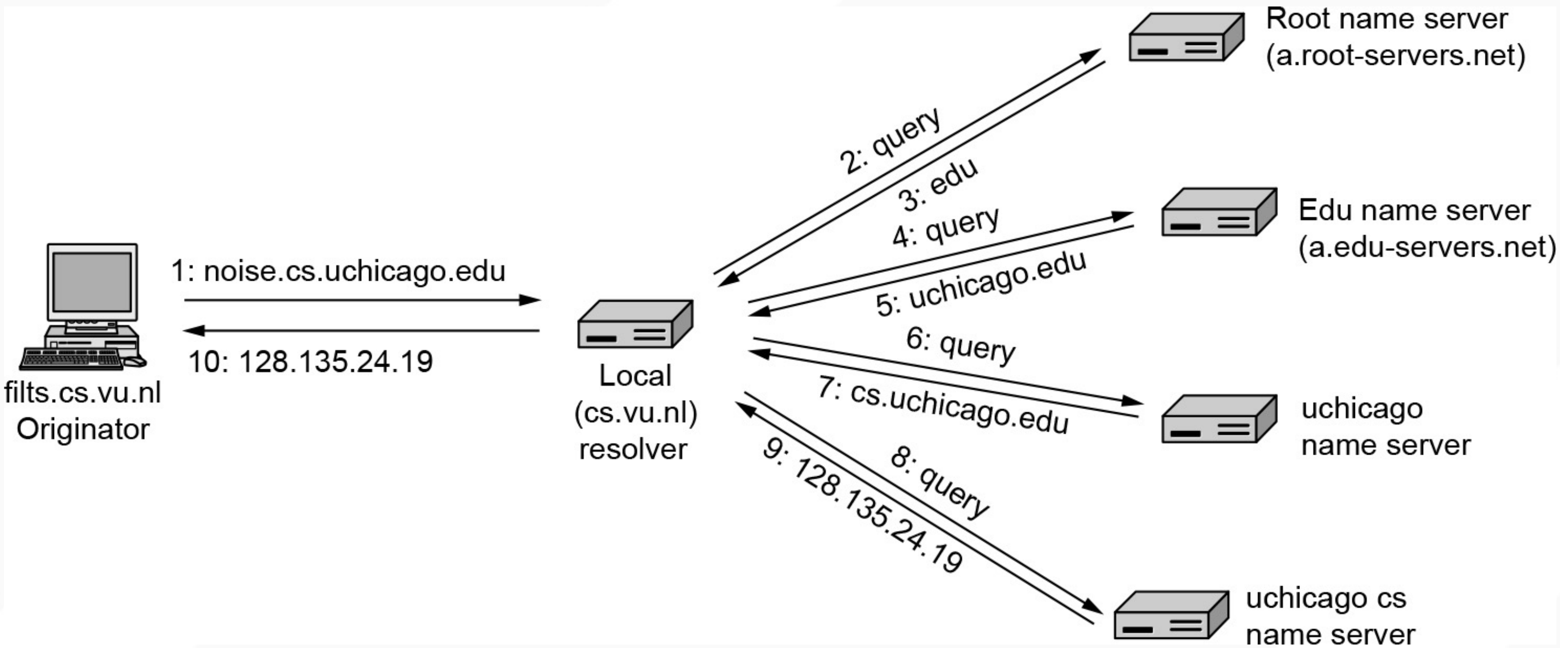
- Most queries and responses are UDP datagrams
  - Two types of queries:
- ***Recursive***: Nameserver responds with answer or error



- ***Iterative***: Nameserver may respond with a referral



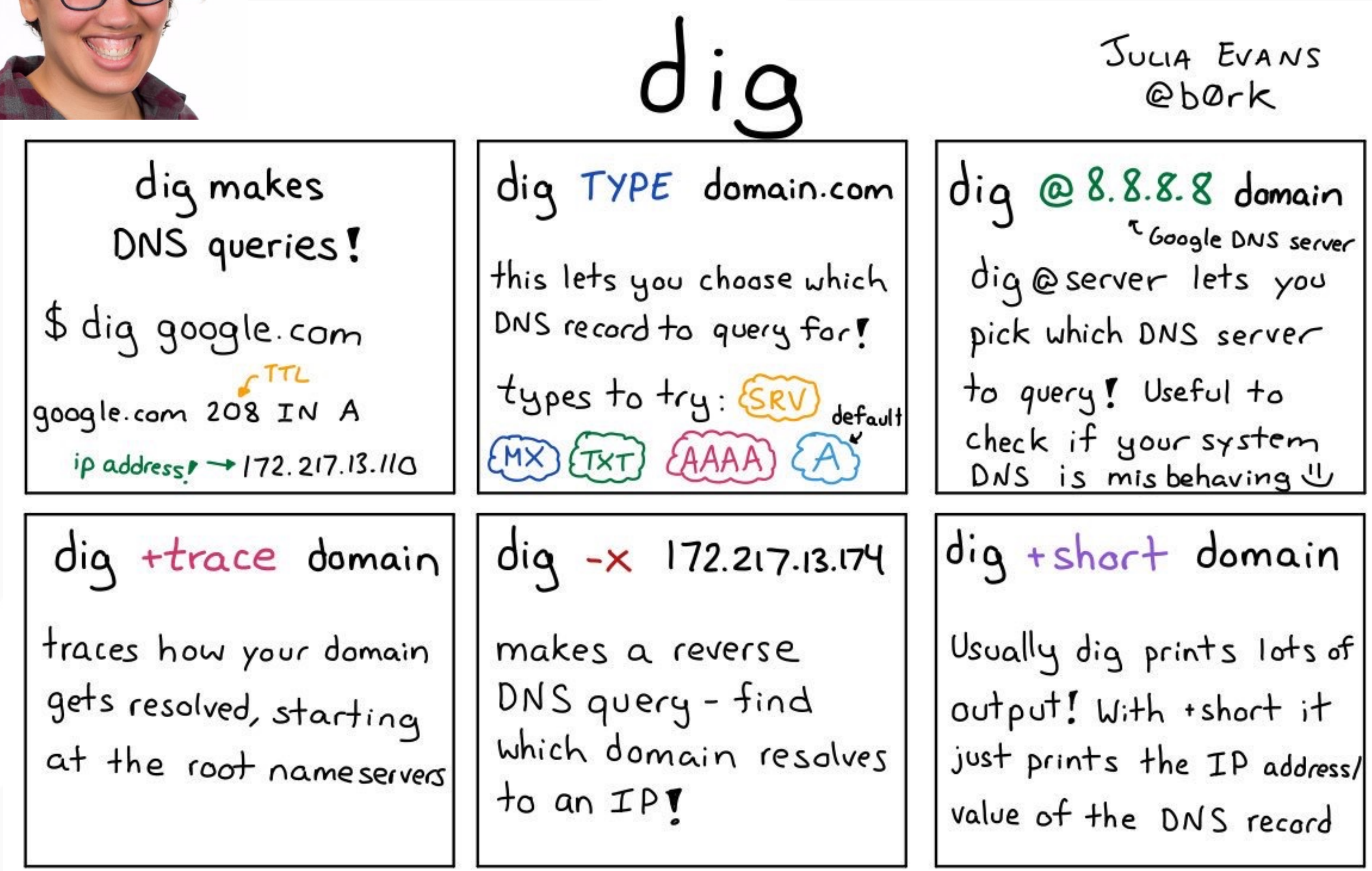
ITERATIVE LOOKUP



DNS CACHING

- Performing all these queries takes time
  - And all this before actual communication takes place
- Caching can greatly reduce overhead
  - The top-level servers very rarely change
    - Popular sites visited often
  - Local DNS server often has the information cached
- How DNS caching works
  - All DNS servers **cache responses to queries**
  - Responses include a time-to-live (TTL) field
    - Server deletes cached entry after TTL expires

JULIA EVAN'S GUIDE TO *DIG*



UC San Diego