CSE 224: NETWORKING FUNDAMENTALS

George Porter
Jan 10, 2023

ATTRIBUTION

UC San Diego

AGENDA

Today and Thusday:

- Basics of networking, addressing, performance metrics
- Basics of Go

Next week:

- Naming and addressing

TODO THIS WEEK

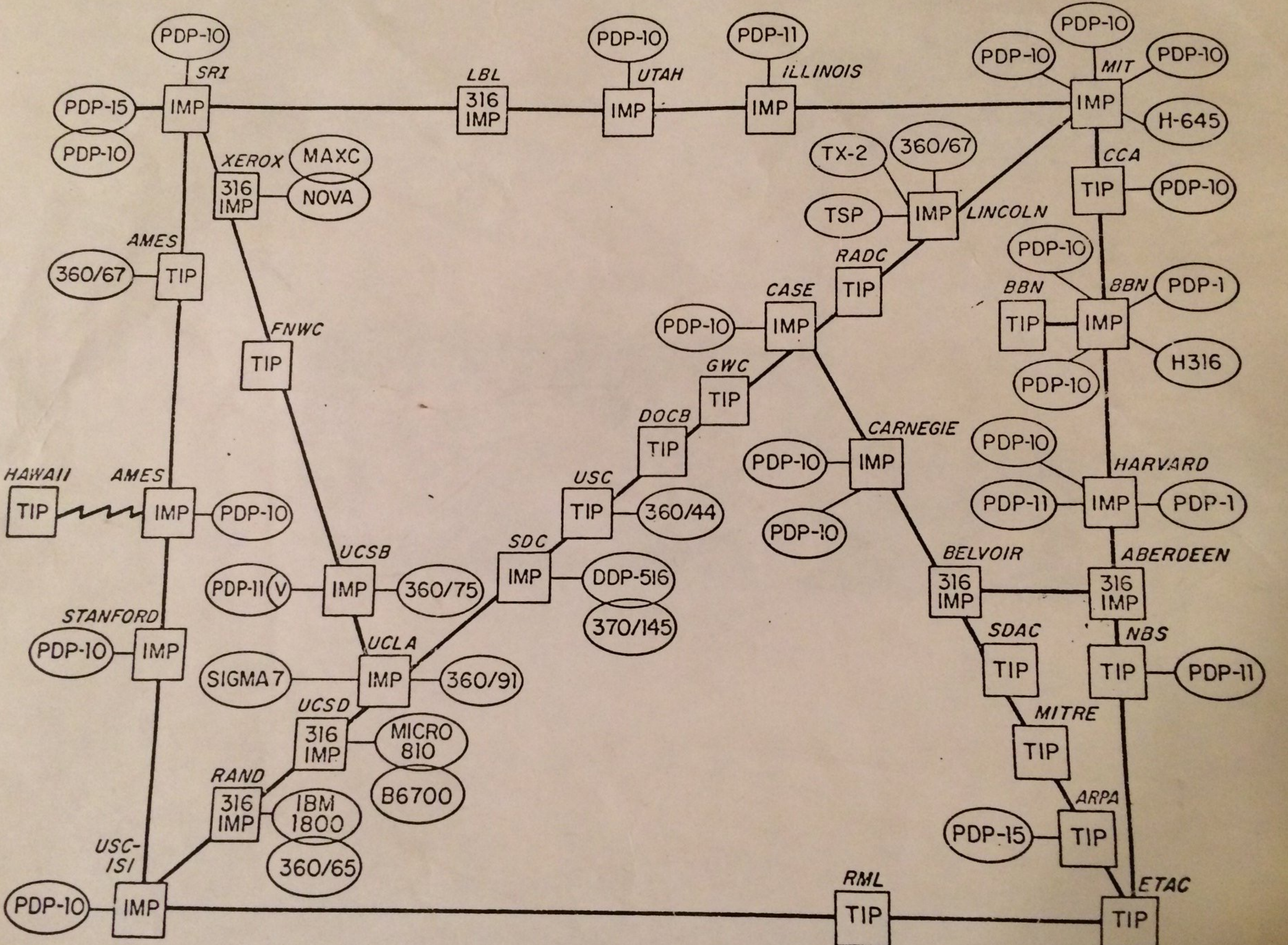
1. Read the first couple chapters of the "Go Language" book esp. Ch 1.
2. Read Network Programming with Go chapters 1, 2, and 3
3. Project 1 is now out (due Jan 17)

BRIEF HISTORY OF THE INTERNET

- 1968 - DARPA (Defense Advanced Research Projects Agency) contracts with BBN (Bolt, Beranek & Newman) to create ARPAnet
- 1970 - First five nodes:
  - UCLA
  - Stanford
  - UC Santa Barbara
  - U of Utah, and
  - BBN
- 1974 - TCP specification by Vint Cerf
- 1984 – On January 1, the Internet with its 1000 hosts converts en masse to using TCP/IP for its messaging

Data from the Internet Society

ARPA NETWORK, LOGICAL MAP, MAY 1973



Outline

1. Performance
2. Layering
3. Addressing

PERFORMANCE METRICS

- Bandwidth: number of bits transmitted per unit of time
- Latency = Propagation + Transmit + Queue
  - Propagation = Distance/SpeedOfLight(*)
  - Transmit = 1 bit/Bandwidth
  - Queue = Time waiting in switches/routers behind other traffic (traffic jam)
- Overhead
  - # secs for CPU to put message on wire
- Error rate
  - Probability P that message will not arrive intact

* In that particular medium

BANDWIDTH VS. LATENCY

1 Byte Object

	Latency: 1 ms	Latency: 100 ms
Bandwidth: 1 Mbps	1,008 µs	100,008 µs
Bandwidth: 100 Mbps	1,000 µs	100,000 µs

10 MB Object

	Latency: 1 ms	Latency: 100 ms
Bandwidth: 1 Mbps	80.001 s	80.1 s
Bandwidth: 100 Mbps	.801 s	.9 s

NETWORK PERFORMANCE MEASUREMENT UNITS

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

TERMINOLOGY STYLE

- Mega versus Mega, Kilo versus Kilo
  - Computer architecture: Mega → 2^{20}, Kilo → 2^{10}
  - Computer networks: Mega → 10^6, Kilo → 10^3
- Mbps versus MBps
  - Networks: typically megabits per second
  - Architecture: typically megabytes per second
- Bandwidth versus throughput
  - Bandwidth: available over link
  - Throughput: available to application
    - E.g. subtract protocol headers, etc.

PERFORMANCE TOOLS

- Ping
  - Test if other side is “alive”
  - Measures round-trip latency
- iperf3
  - Times how long it takes to send N bytes to the other endpoint
  - Used to calculate bandwidth

Outline

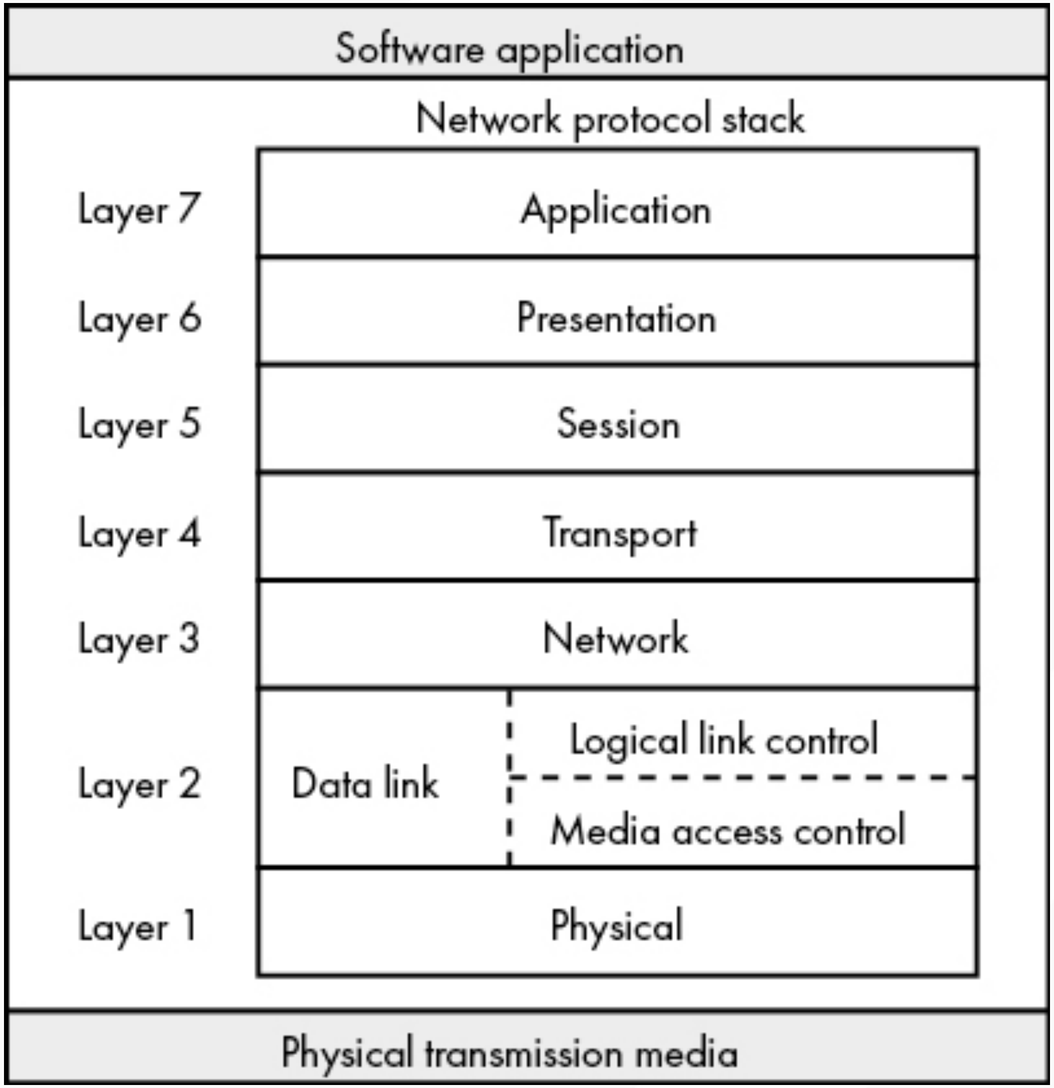
1. ~~Performance~~
2. Layering
3. Addressing

NETWORKING AND LAYERING

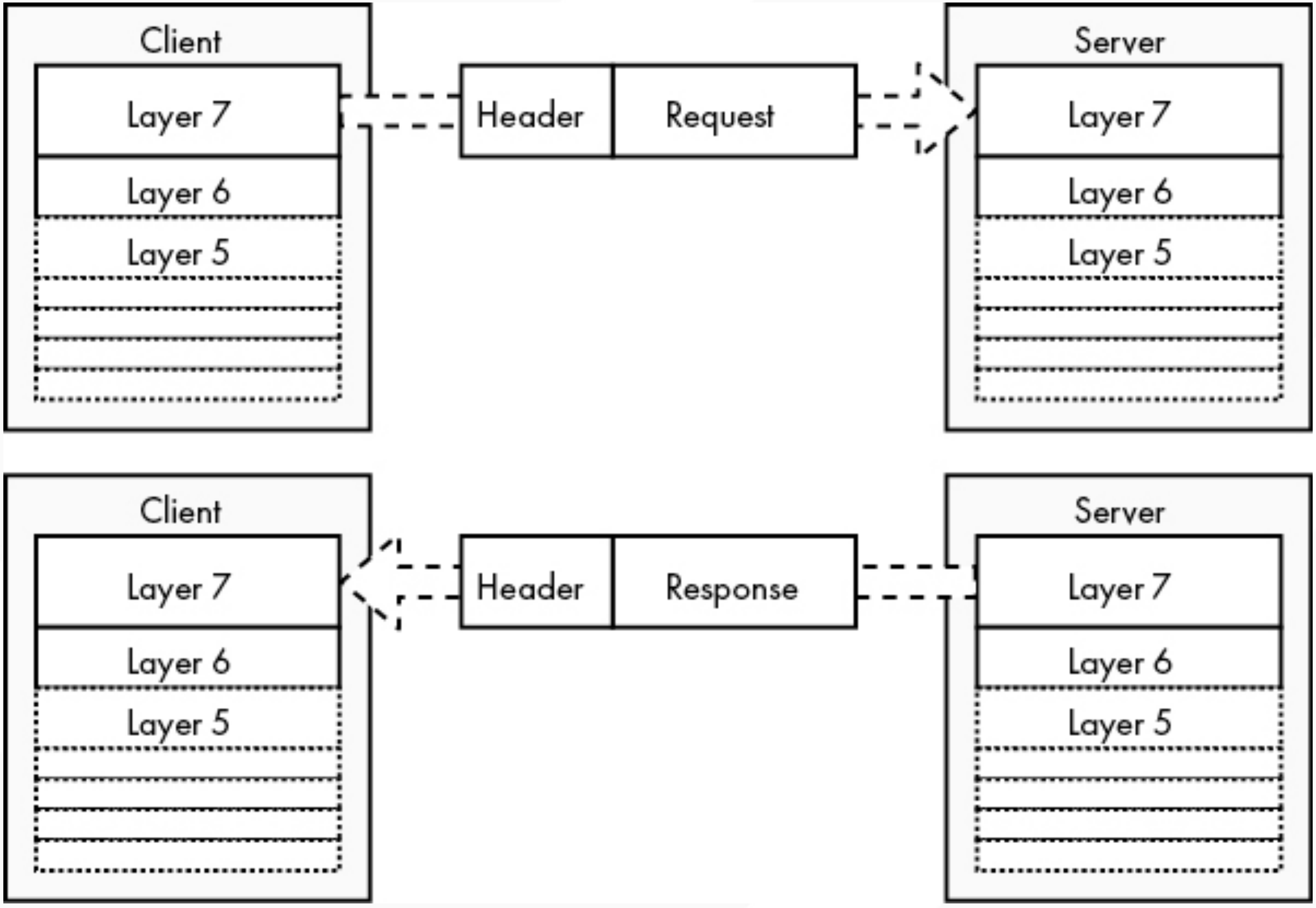
"All problems in computer science can be solved by another level of indirection."

David J. Wheeler

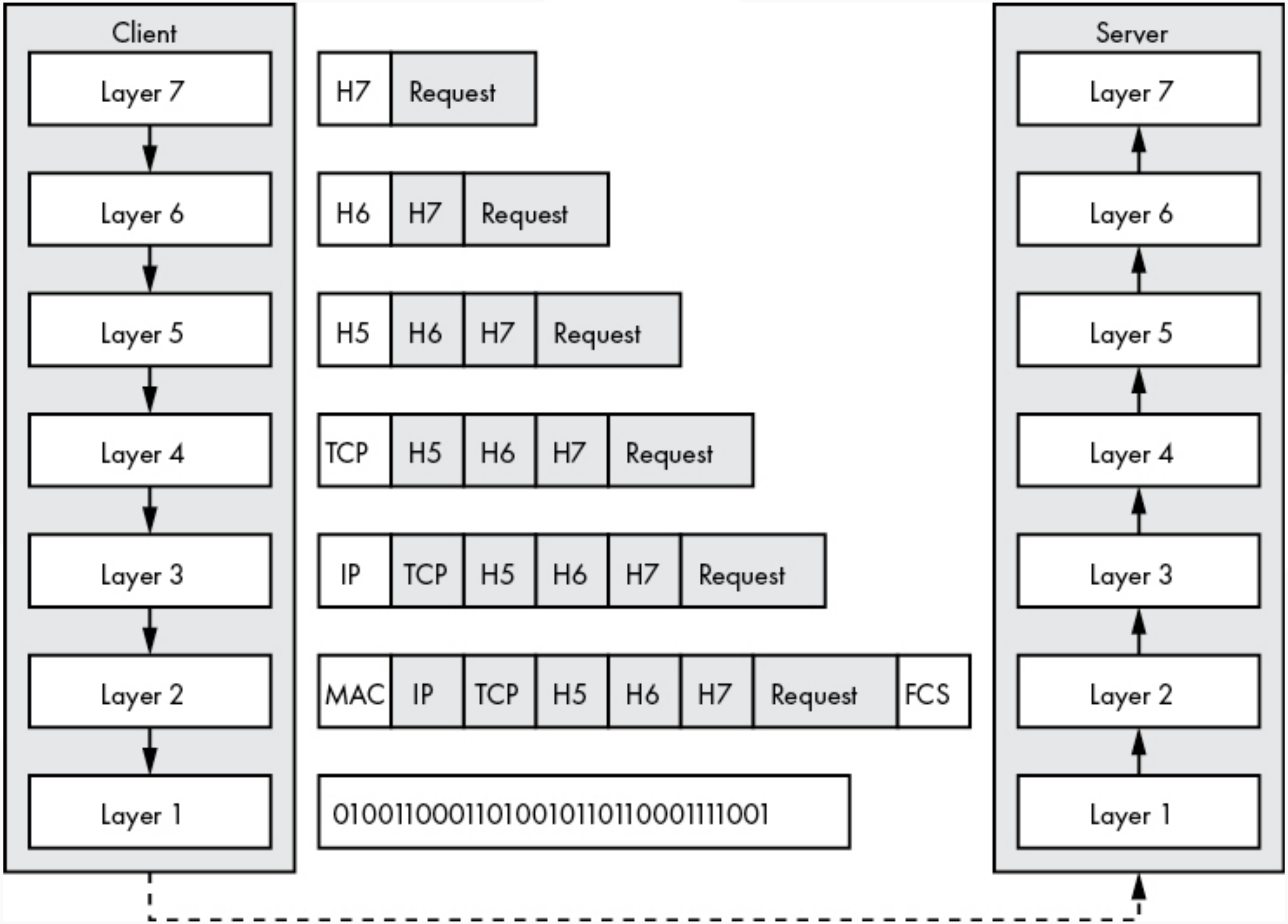
OSI NETWORK STACK



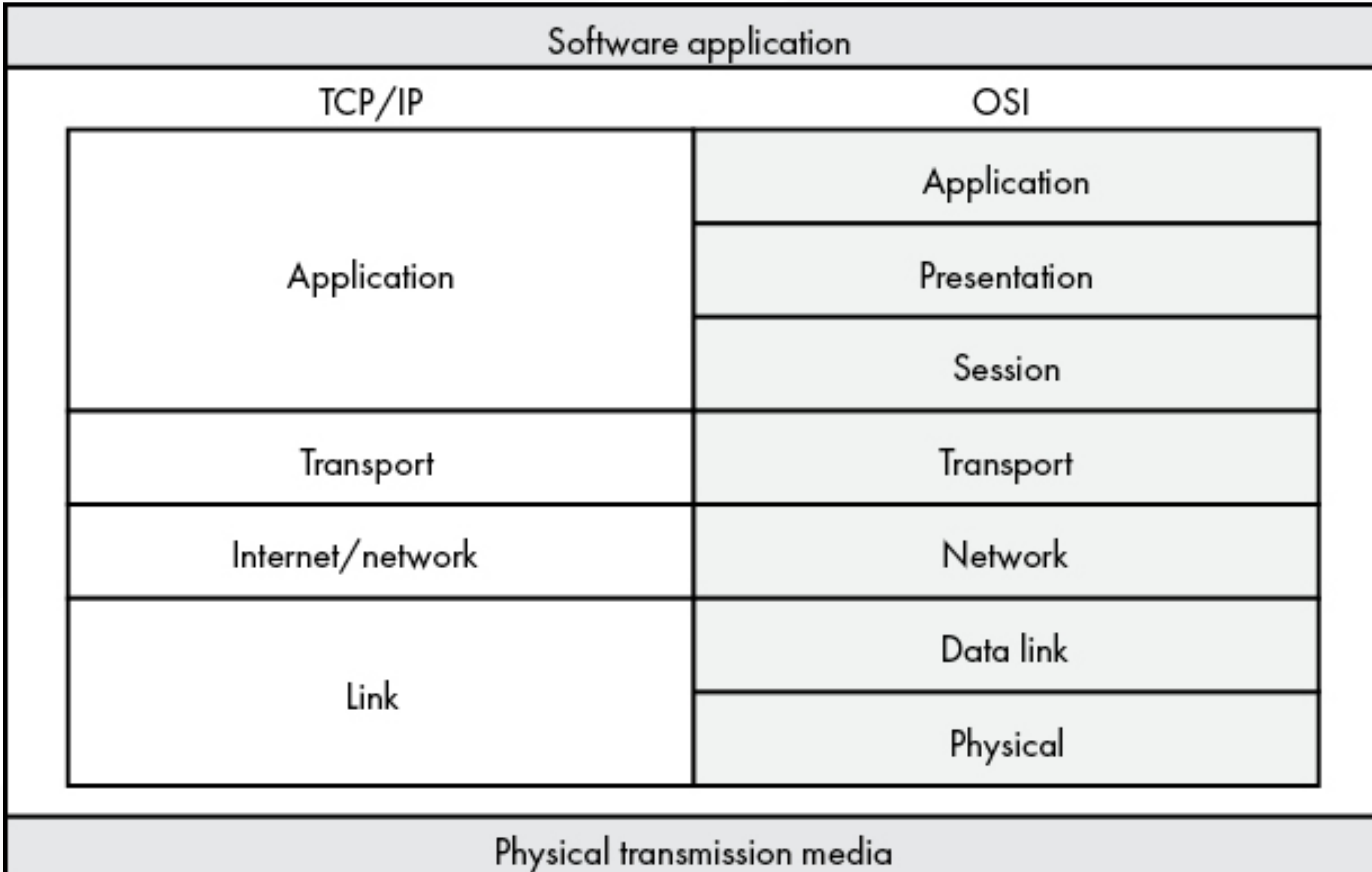
PROTOCOLS GOVERN MESSAGES EXCHANGED WITHIN A SINGLE LAYER



LAYERING AND ENCAPSULATION

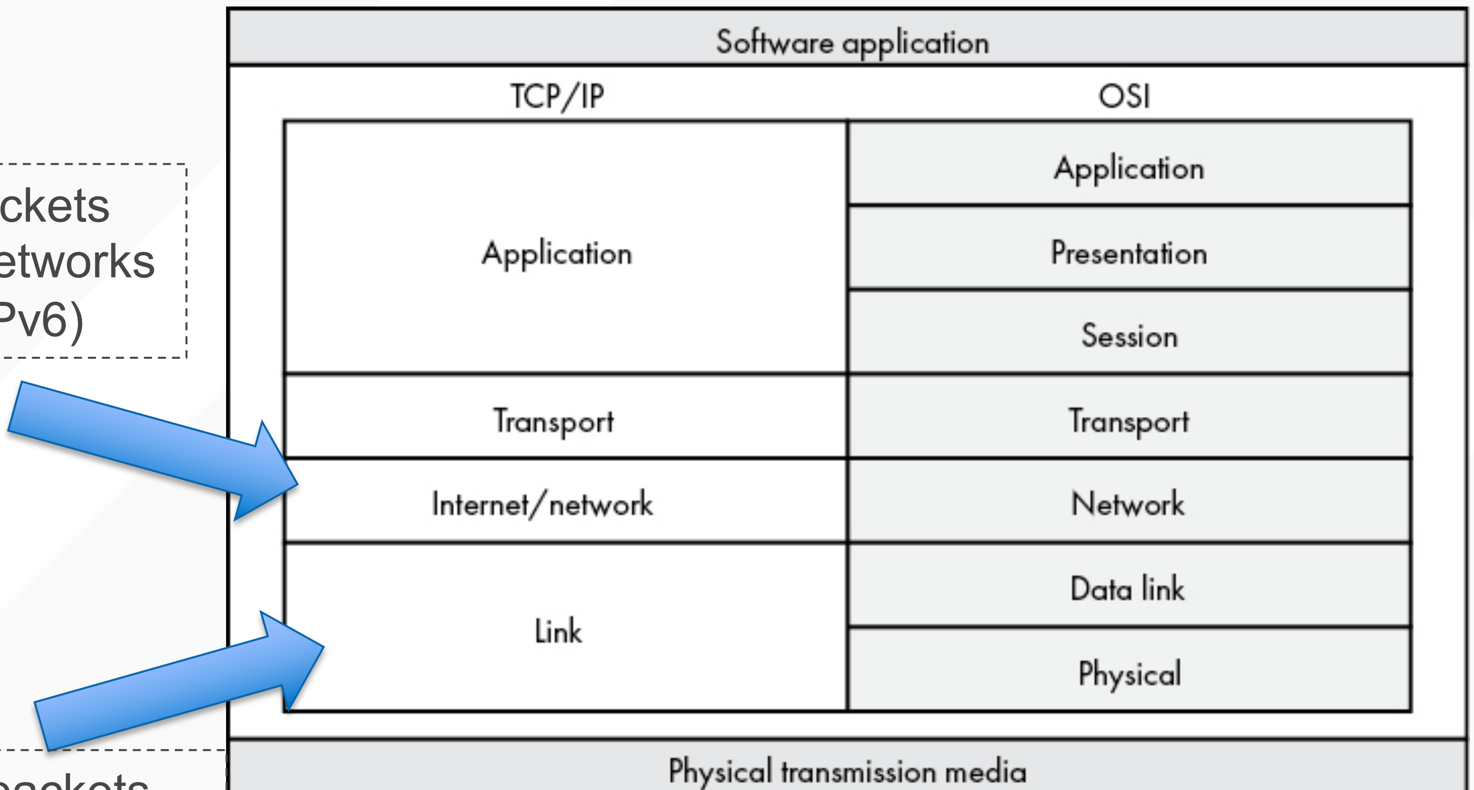


TCP/IP MODEL (VS OSI MODEL)



TCP/IP MODEL (VS OSI MODEL)

Forwarding packets between subnetworks (E.g. IPv4 or IPv6)



Forwarding packets (frames) within one subnetwork (E.g. Ethernet)

Outline

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

MOTIVATION FOR OUR DISCUSSION OF ADDRESSING

1. When you implement a *client* application, you will typically need to communicate with a remote system/service via its *IP Address*

2. When you implement a cloud-hosted *network service*, you will typically be assigned a *subnet* of IP addresses to use for your various server programs, and will need to use and manage those appropriately

IP PROTOCOL

- Scenario: An internet connected device wants to send a message to another internet connected device anywhere in the world
  - IP protocol handles this
  - Prepend an “IP header” to the message, set the destination IP address, set the source IP address, and Internet routers will forward it along the shortest path till it gets to the destination network
  - From there, the destination network forwards the packet to the intended host

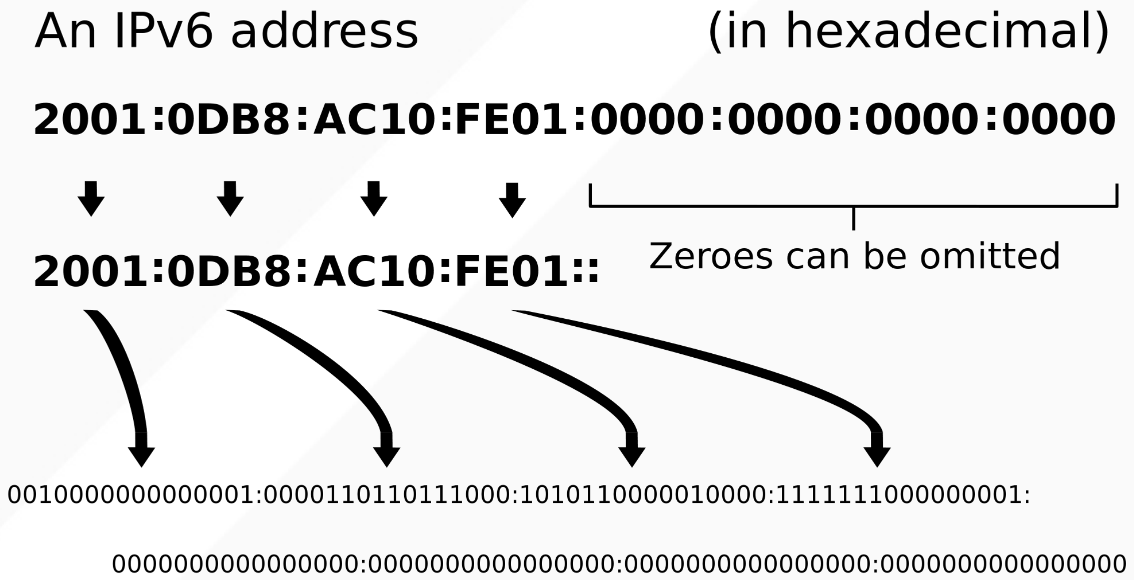
** Simplified model but sufficient for our purposes*

IP VERSION 4 (IPV4)

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

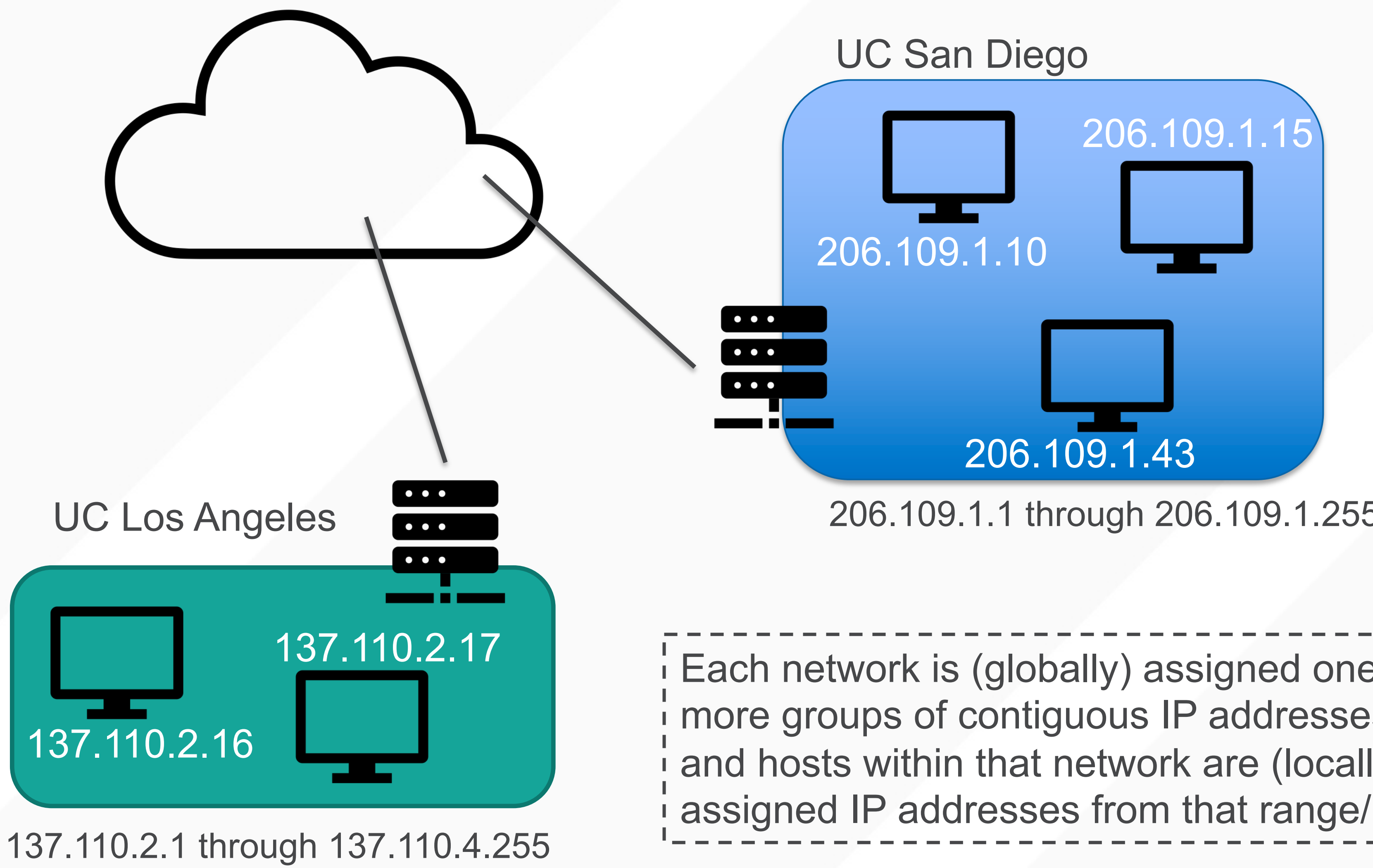
- 32-bits
- Usually represented as four 8-bit bytes (octets)
- E.g. 206.109.1.6, 127.0.0.1, 192.168.1.10

IP VERSION 6 (IPV6)



- 128-bits
- Represented as 8 16-bit blocks, in hex, separated by colons
- E.g. 2001:4860:4860::8888, ::1, 2345:0425:2CA1:0000:0000:0567:5673:23b5

THE INTERNET: A NETWORK OF SUB-NETWORKS



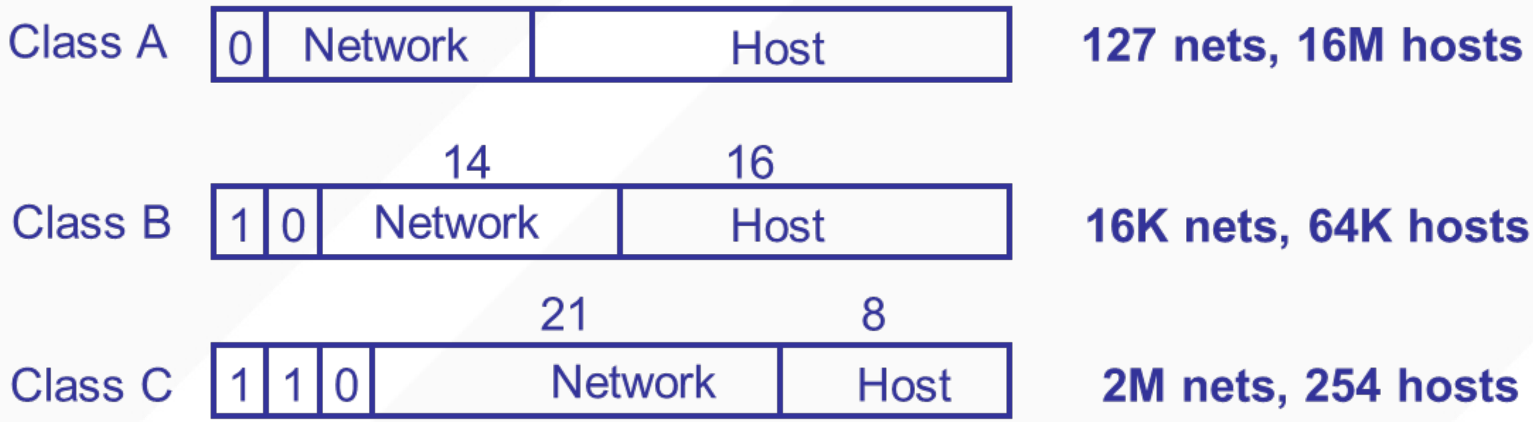
Each network is (globally) assigned one or more groups of contiguous IP addresses, and hosts within that network are (locally) assigned IP addresses from that range/pool

ROUTE AGGREGATION

- 2^32 possible (IPv4) addresses spread across ~100,000 independent networks
- Each router keeps the "next hop" on a per-network basis, not per-host basis
- But networks can be different sizes (e.g. UCSD is bigger than a small startup)
- Each router has to keep a list of networks (and their next hops) + how "big" each network is

CLASS-BASED ADDRESSING (NOT REALLY USED ANYMORE)

- Most significant bits determines “class” of address

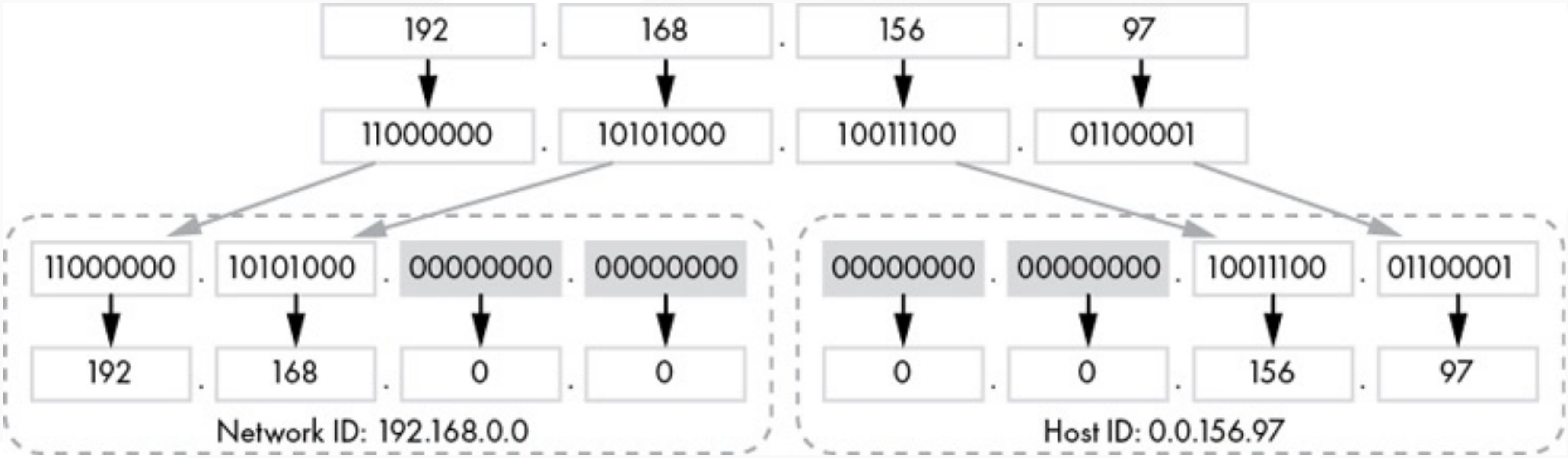


- Special addresses
  - Class D (1110) for multicast, Class E (1111) experimental
  - 127.0.0.1: local host (a.k.a. the loopback address)
  - Host bits all set to 0: network address
  - Host bits all set to 1: broadcast address

CLASS-BASED ADDRESSING EXAMPLES

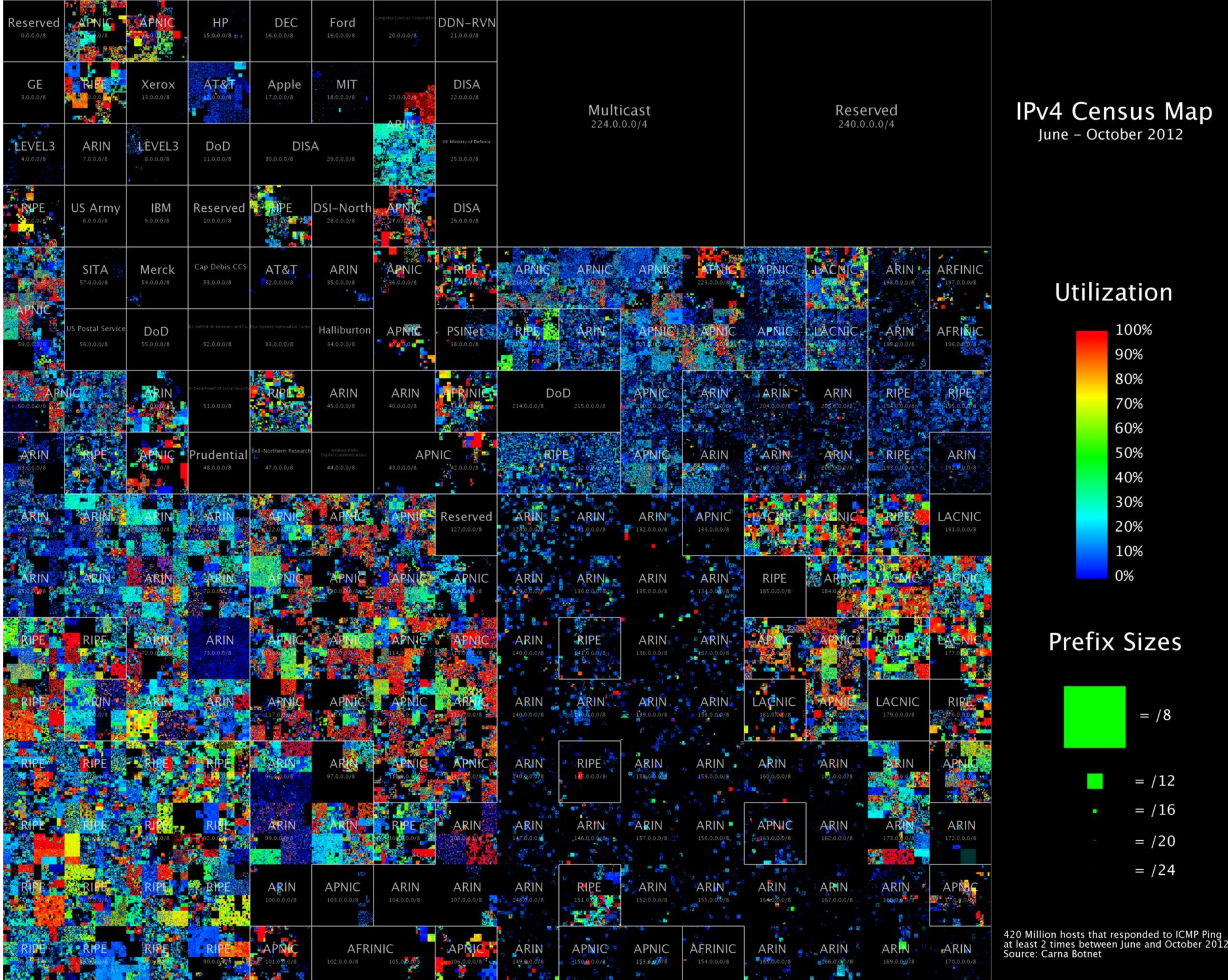
Network ID	First octet	Second octet	Third octet	Fourth octet	Host ID
8 bits	Network 10	Host 1	Host 2	Host 3	24 bits
16 bits	Network 172	Network 16	Host 1	Host 2	16 bits
24 bits	Network 192	Network 168	Network 1	Host 2	8 bits

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



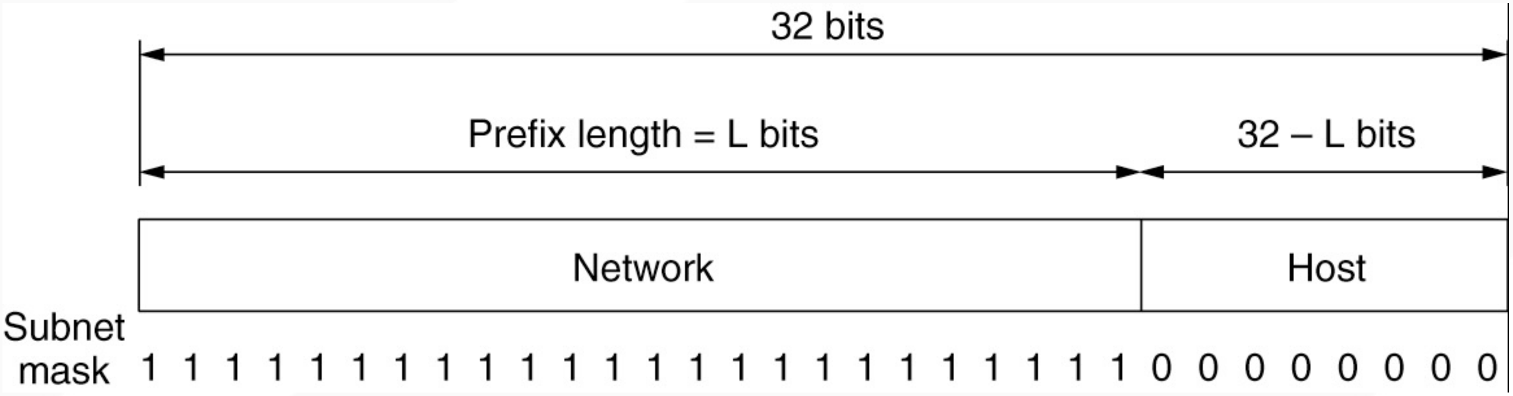
IPv4 Census Map
June – October 2012
Multicast
224.0.0.0/4
Reserved
240.0.0.0/4
Utilization
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Prefix Sizes
= /8
= /12
= /16
= /20
= /24
420 Million hosts that responded to ICMP Ping at least 2 times between June and October 2012
Source: Carna Botnet

CIDR

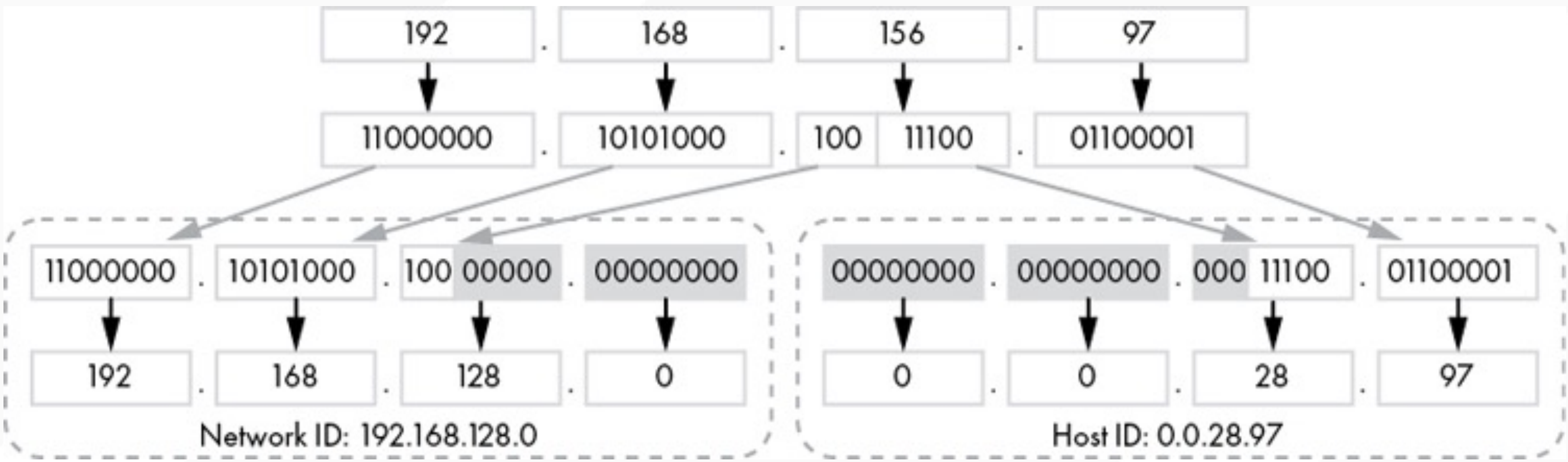
- Classless Inter-Domain Routing (1993)
  - Networks described by variable-length prefix and length
  - Allows arbitrary allocation between network and host address

  - 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

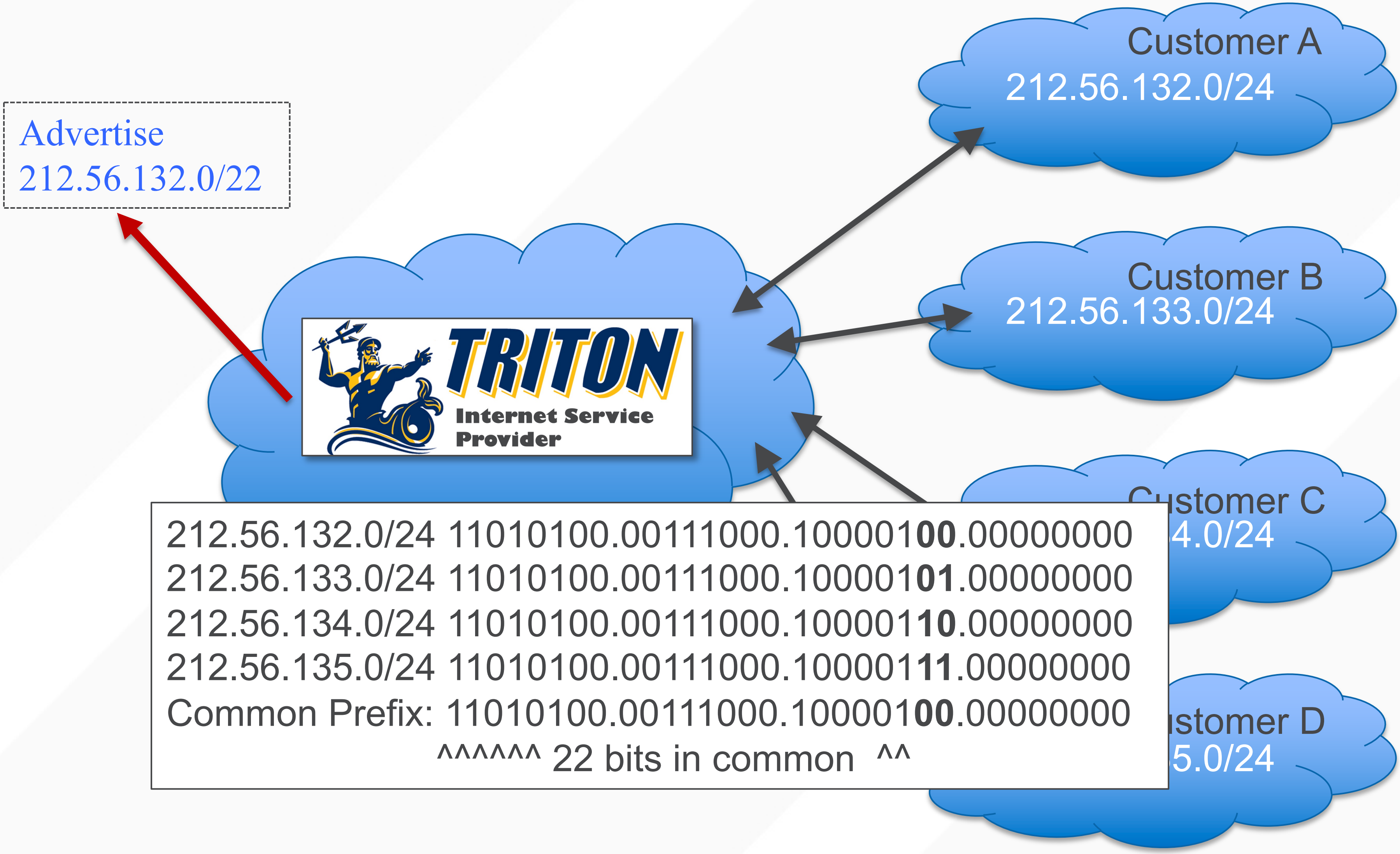
SUBNETS AND NETMASKS



192.168.156.97/19



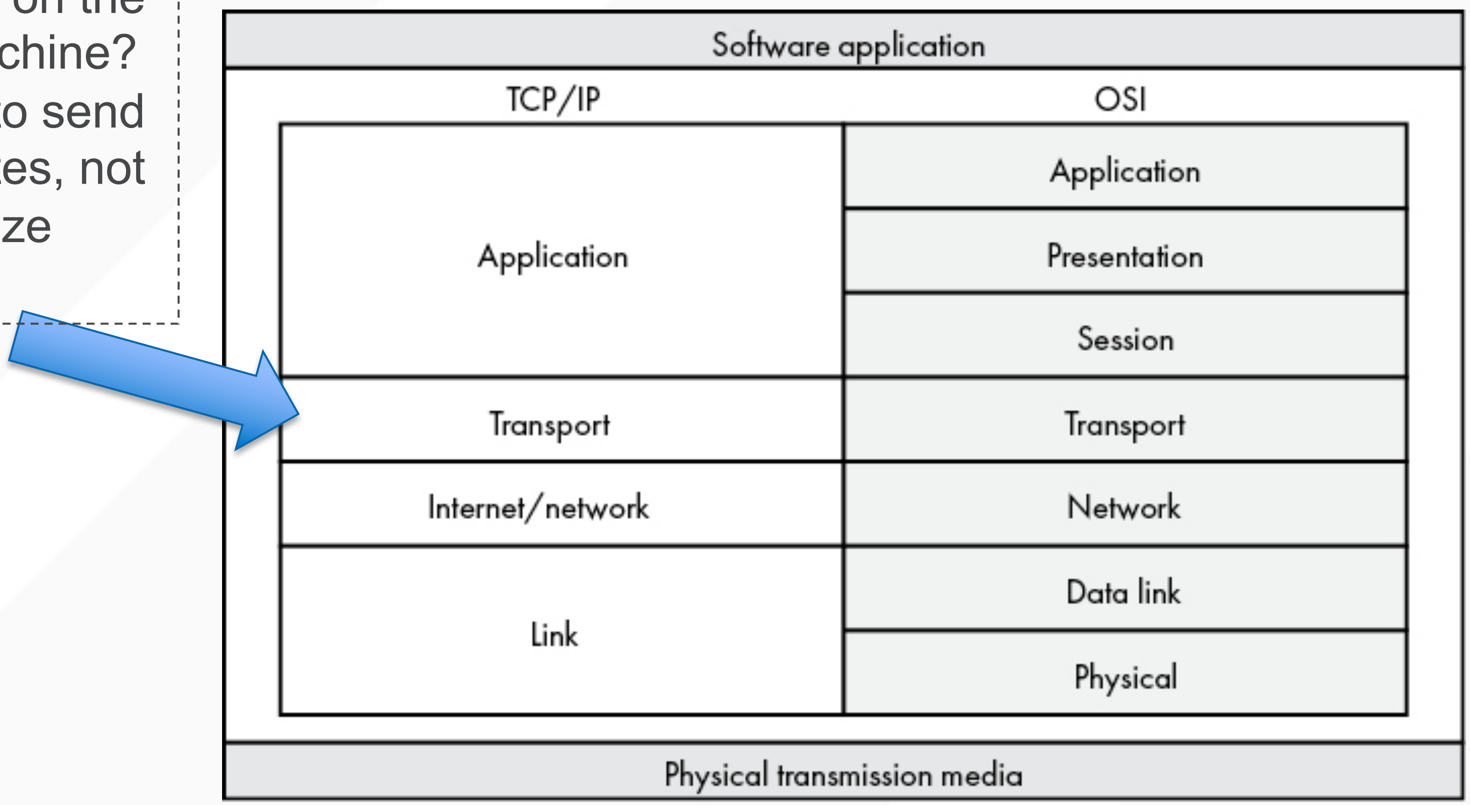
ADDRESS AGGREGATION EXAMPLE



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 ^^

TCP/IP MODEL (VS OSI MODEL)

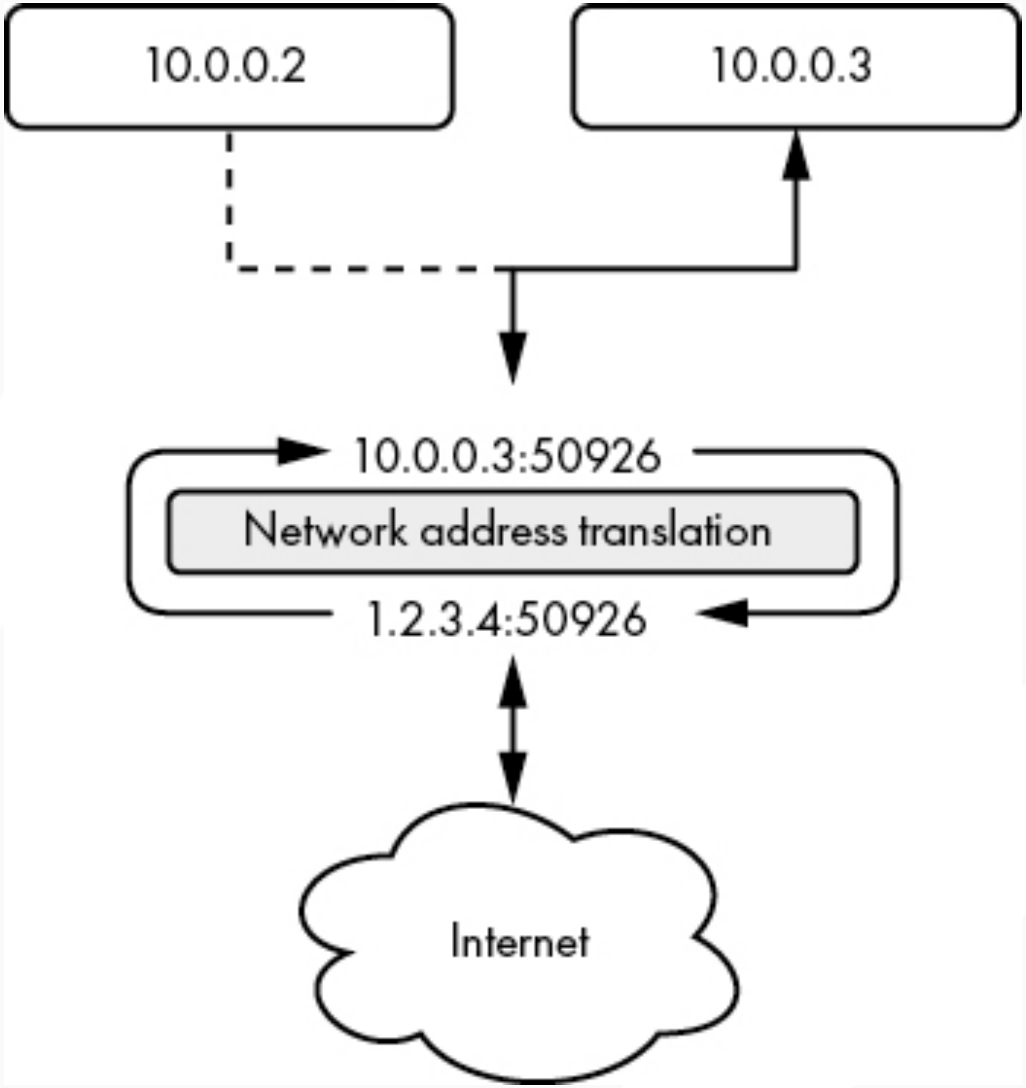
Which *process* on the destination machine? What if I want to send a stream of bytes, not just a limited-size message?



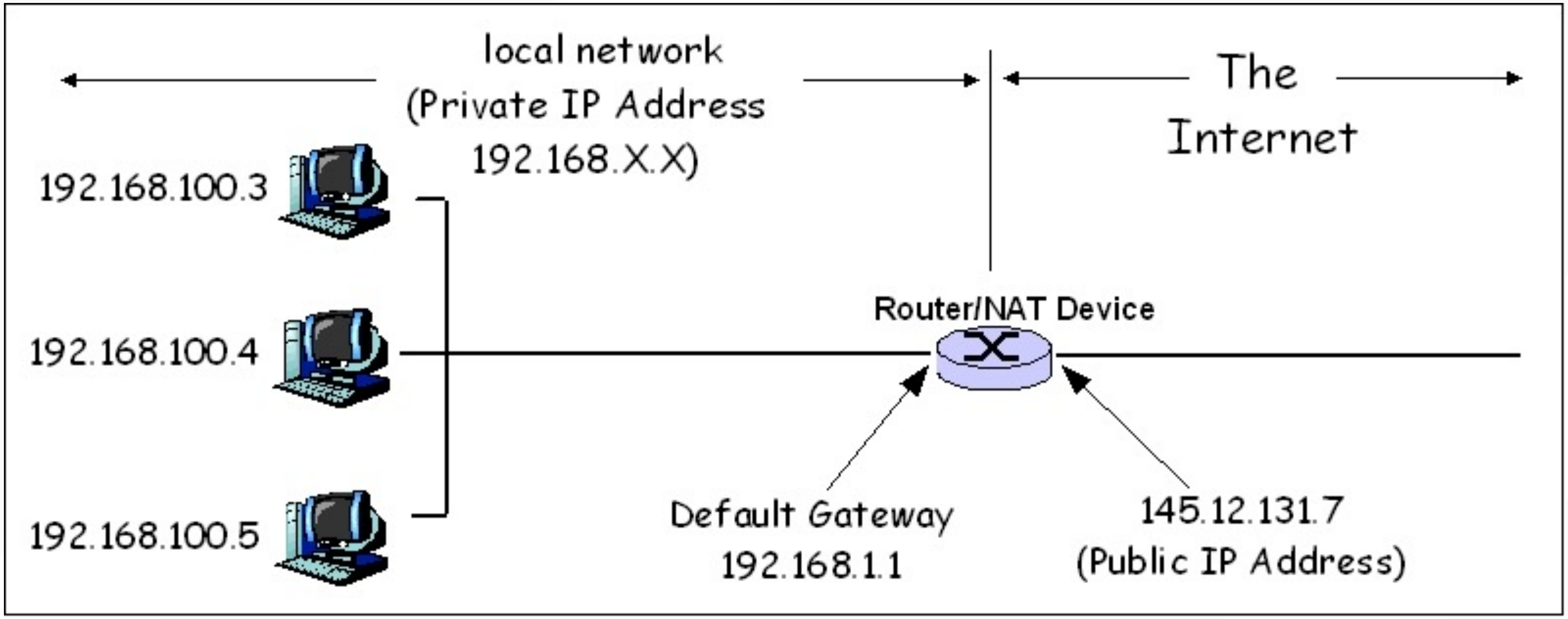
PORTS

- 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

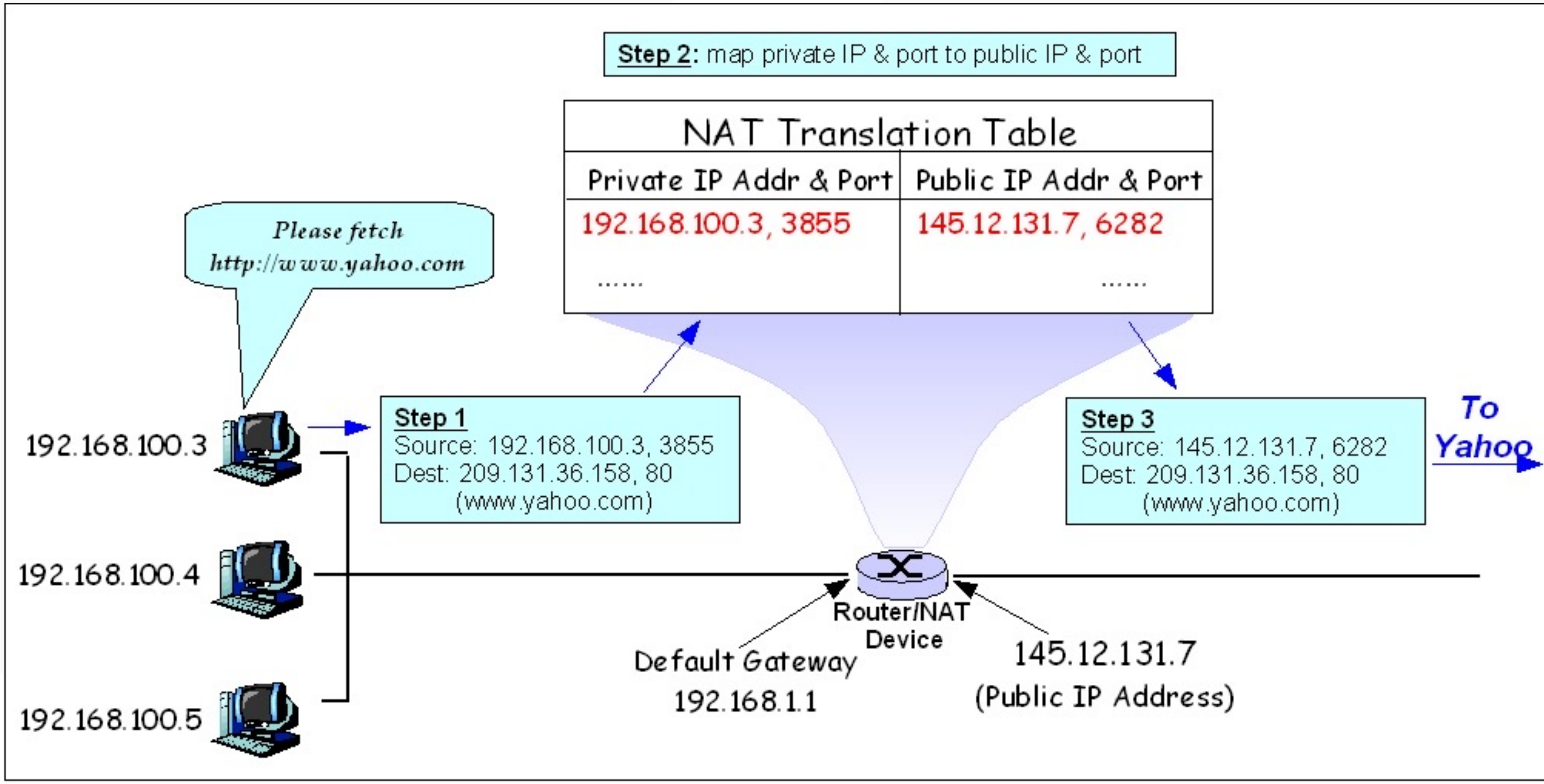


LOAD BALANCING VIA NETWORK ADDRESS TRANSLATION



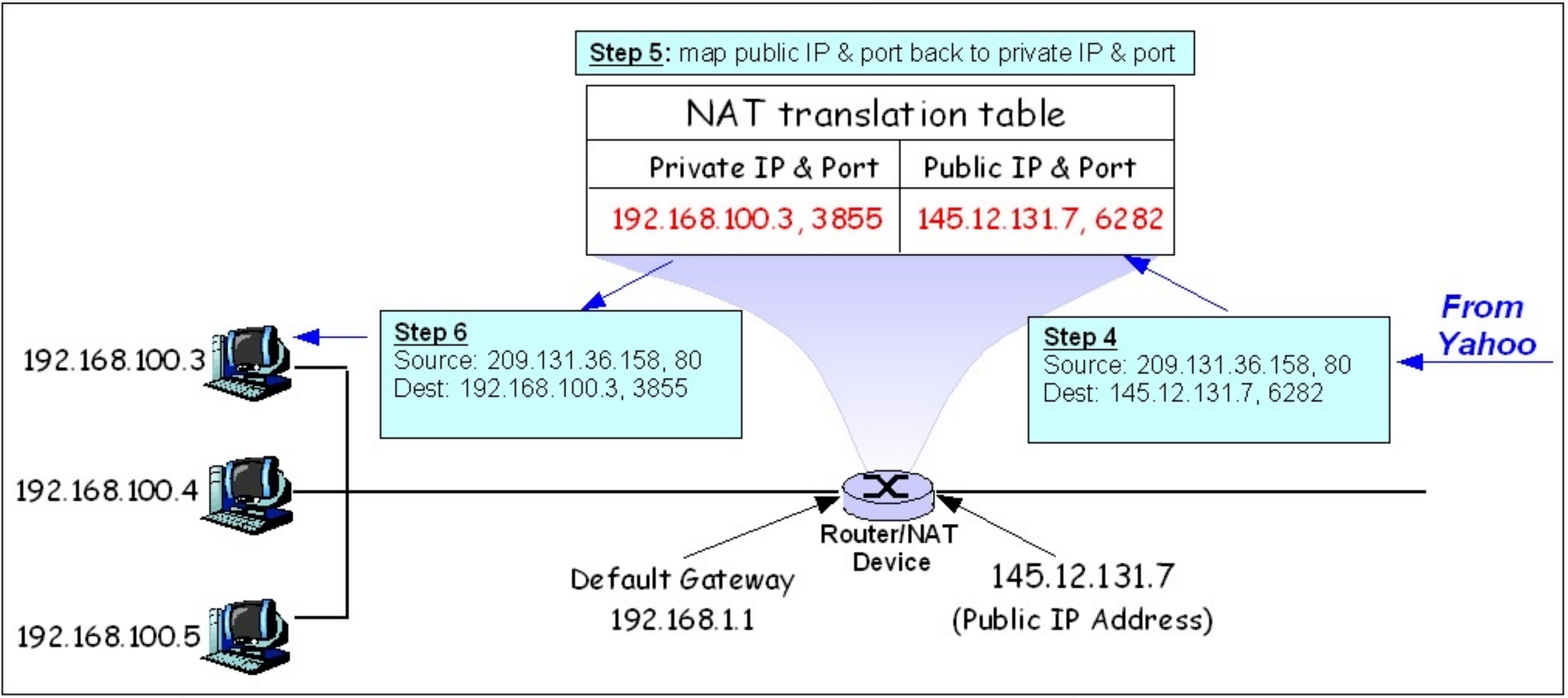
Courtesy of: https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols

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

UC San Diego