Naming and Addressing

Jeff Chase
Duke University

Name Spaces

OS-level

- Address spaces
- File tree and file pathname
- File descriptors

Networks

- Ethernet/LAN "MAC" addresses (48-bit)
- IP addresses (32-bit or 64-bit in IPv6)
- Port numbers
- DNS names, e.g., *vmm01.cod.cs.duke.edu*
- URL = DNS name + port + file pathname

Services

Naming issues

- Aliasing
  - Unix “links”
- Referential integrity
  - Reference counts and garbage collection
  - Dangling references
- Name changes
- Name resolution
  - Replication
  - Mobility
  - Security
- Location, location, location

Context and Hierarchy

- Names are assigned/resolved relative to a context
  - Uniformity vs. autonomy
  - Autonomy ➔local control and customizability
  - Autonomy ➔ fragmentation
  - Local vs. global name spaces
  - Collisions
- Nested contexts
- Contexts may be controlled by different administrative authorities

Naming Resolution Structures

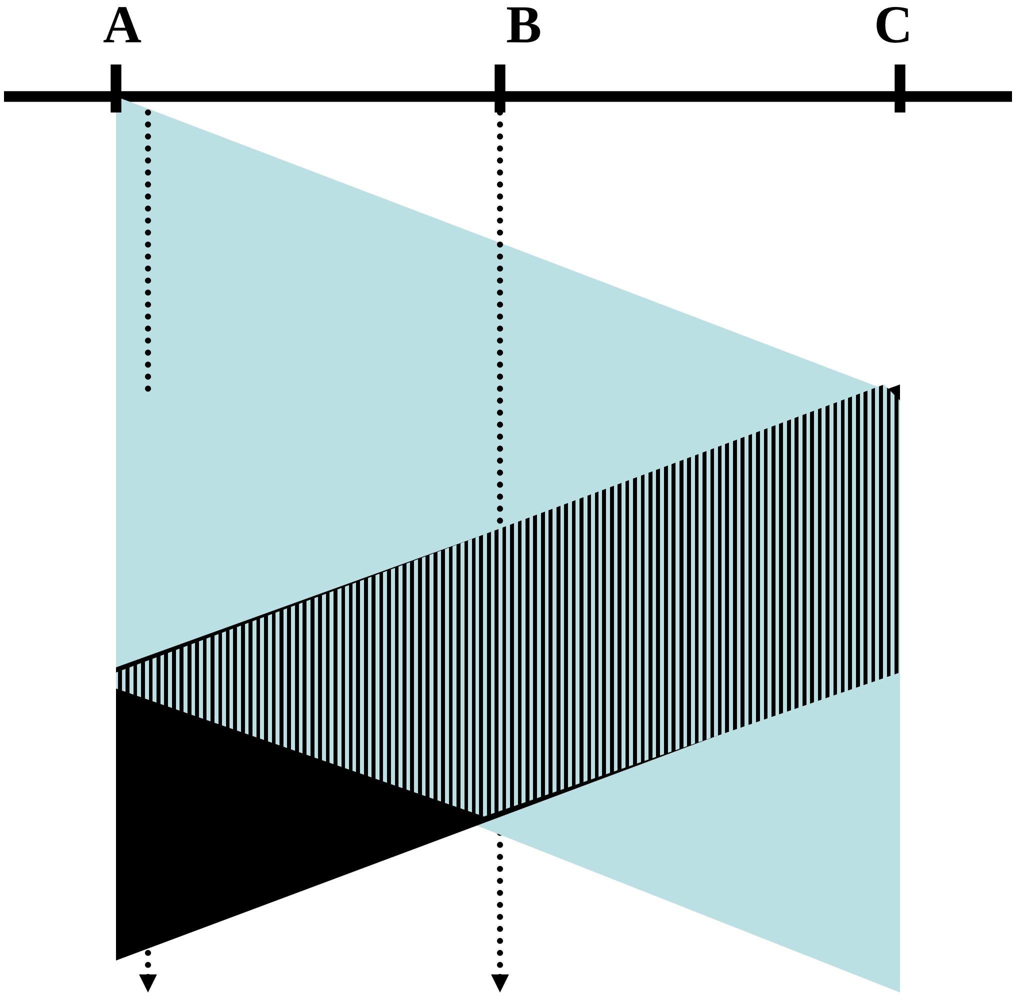
- Telephone book
  - Everyone has one
  - All lookups are local
  - Highly resilient
- Central server: directory assistance
  - Authoritative
  - Useful even when you don't have your phone book
- Scalable directory service?
- What about name changes?

Ethernet Addressing

- 48-bit address space
- Hardwired into network interface
- Sequence of six bytes/octets: 12:34:56:78:9A:BC
- First three bytes identifies the vendor of the NIC
- Name space is “flat”
  - No structure in the names with respect to location
- How to route network traffic to a node by MAC address?

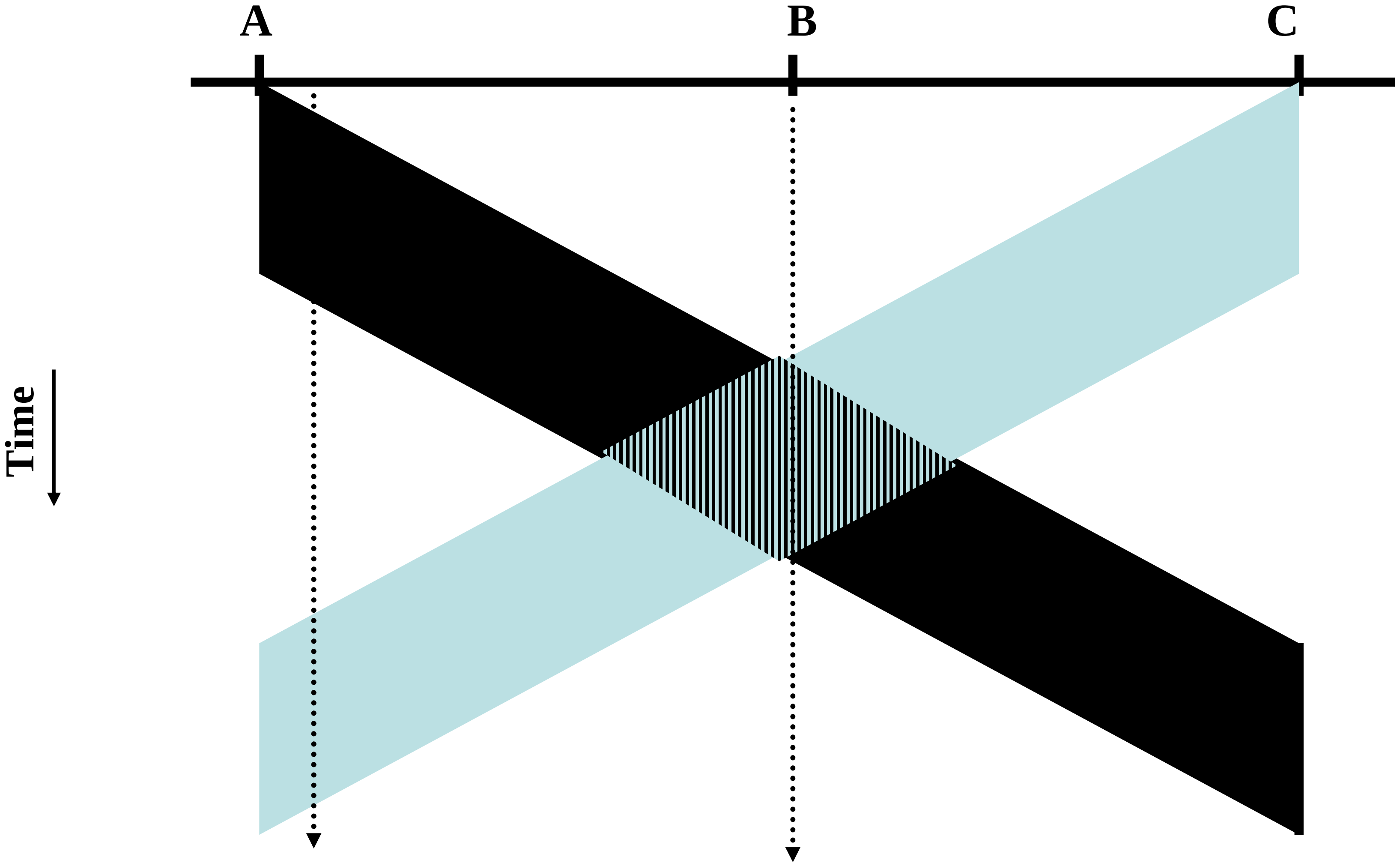
Ethernet 101

- May be a broadcast medium.
  - E.g., hubbed network
  - E.g., wireless (802.11)
- Collisions may occur.
  - "CSMA/CD"
  - Exponential backoff
- All nodes must be able to detect the collision.
  - Any node can be sender
- => Must either have short wires, long packets, or both.
- Modern wired Ethernet:
  - full-duplex
  - switched point-to-point



[Srini/Anderson]

Collision Detection: Failure



[Srini/Anderson]

ARP

LAN (Ethernet) Address Resolution Protocol

- Each node/if is configured to reside on a given IP subnet assigned to its LAN.
- To send to an IP address within the subnet, must know MAC address.
- ARP resolves IP address to MAC address.
- ARP broadcasts "who on this LAN is named Fred"?
- Exactly the node with IP address "Fred" responds.
- Each node maintains an ARP cache of IP->MAC.
- Caches IP->MAC mapping for incoming ARPs.

ARP/MAC Questions

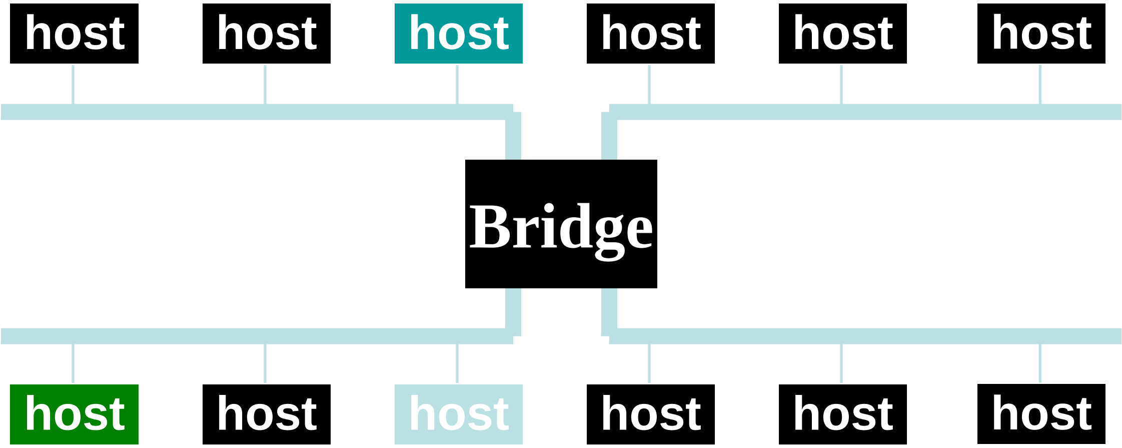
- What if nonexistent destination “Fred”?
- Stale name bindings in the cache?
  - Nodes can change IP address (DHCP)
- Manageability? Plug-and-Play
- Secure?
- Are MACs unforgeable?
  - Software licensing by MAC
  - What is the MAC address for a virtual machine?
- Scalable?

ARP Poisoning/Spoofing

- Doctor up ARP packets
- Send bogus or aliased MAC addresses
- Send false mappings
- Confuse bridges/switches
- Confuse hosts
- Redirect traffic to your MAC or into a black hole
- ….or to your Registration Page
- Example:
  - Hi Alice, my name is Bob, I am at MAC X.
  - Hi Bob, my name is Alice, I am at MAC X.
  - Traffic Alice <-> Bob passes through X.

Scaling Ethernet

- Self-learning bridges/switches.
- Connect them together in “arbitrary” topologies



Scaling Ethernet

- Bridges learn where MACs are connected.
  - Direct-connected to local port.
  - Connected to a neighboring switch/bridge
  - And so on…
- Cache source (MAC, port) when frames go by.
- Broadcast if you don't know where a destination is.
- Topology issues?
- Manageable? Scalable?

DNS 101

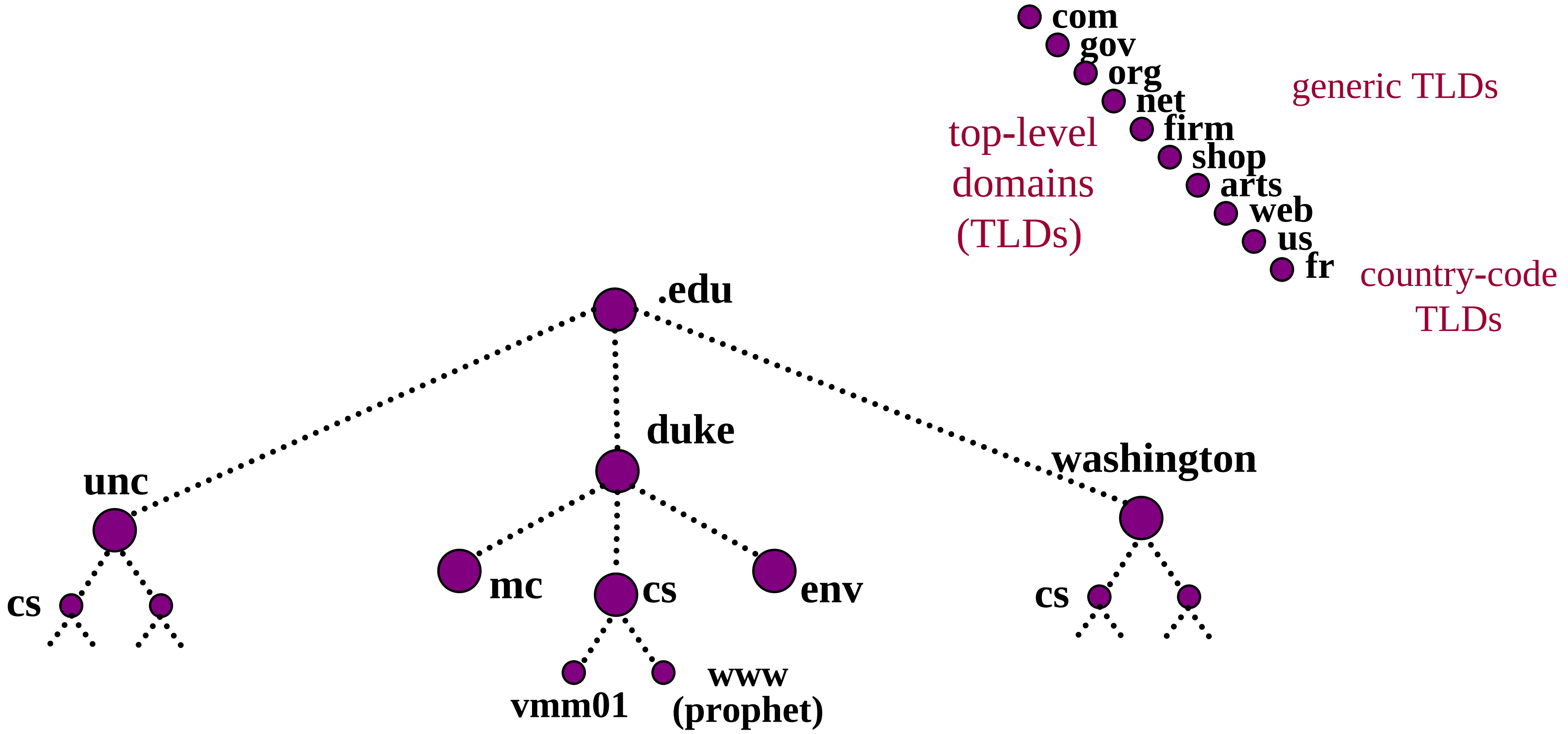
Domain names are the basis for the Web's global URL space.

- Symbolic veneer over the IP address space
  - Human readable
- autonomous naming domains, e.g., *cs.duke.edu*
  - specific nodes, e.g., *vmm01.cs.duke.edu*
  - service aliases (e.g., www, mail servers)
- Almost every Internet application uses domain names when it establishes a connection to another host.
- "Phone book for the Internet"

DNS Service

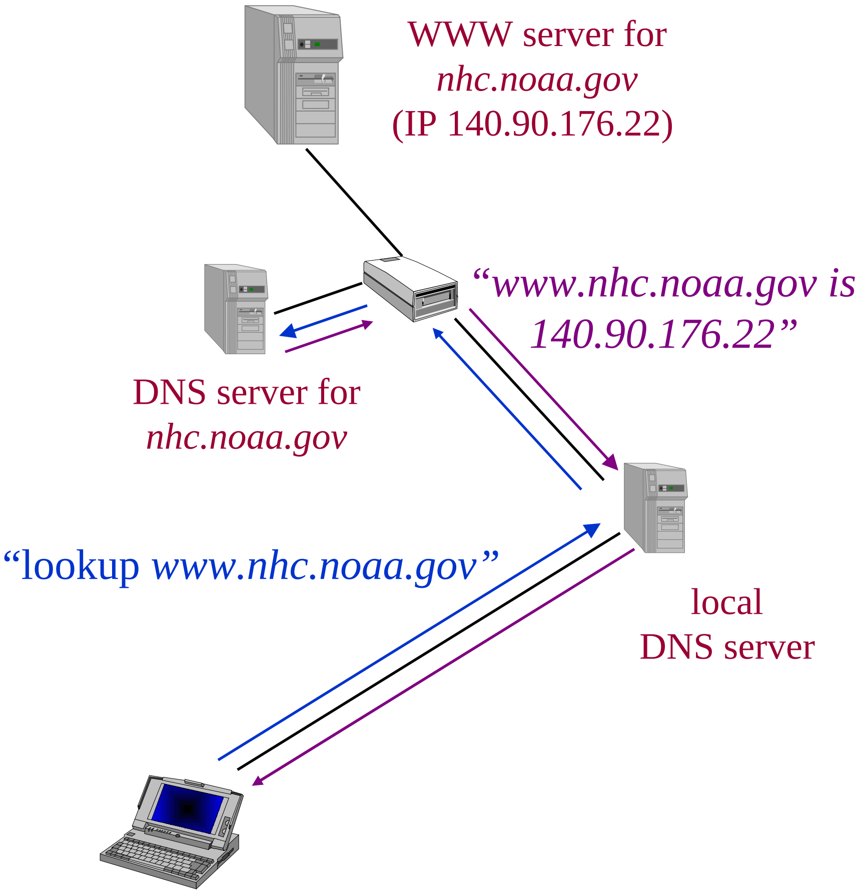
- The *Domain Name System* (DNS) is a planetary name service that translates Internet domain names.
    - maps *<DNS name>* to *<IP address>*
    - (mostly) independent of location, routing etc.
- *Hierarchical* name space and service structure:
    - Fully qualified names are "little endian"
    - Scalability
    - Decentralized administration
    - Domains are naming *contexts*
- Replaced primordial flat *hosts.txt* namespace

Domain Name Hierarchy



How is this different from hierarchical directories in distributed file systems? Do we already know how to implement this?

DNS Service 101

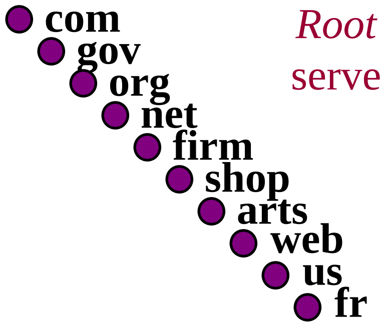


- client-side *resolvers*
  - typically in a library
  - *gethostbyname, gethostbyaddr*
- cooperating servers
  - query-answer-referral model
  - forward queries among servers
  - server-to-server may use TCP ("zone transfers")

DNS Name Server Hierarchy

DNS servers are organized into a hierarchy that mirrors the name space.

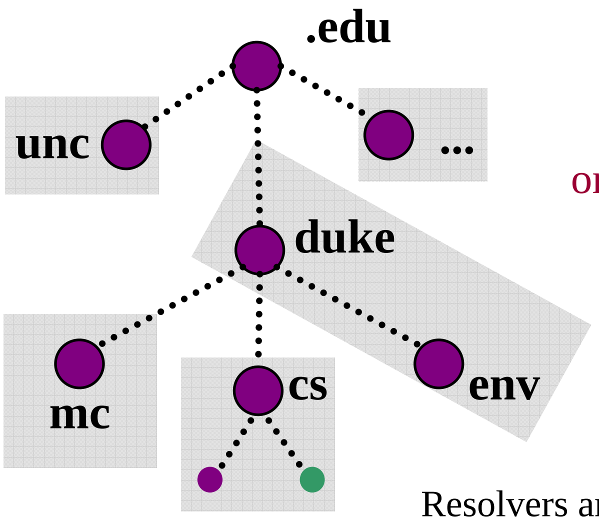
Specific servers are designated as *authoritative* for portions of the name space.



Root servers list servers for every TLD.

Servers may delegate management of *subdomains* to child name servers.

Parents refer subdomain queries to their children.



Subdomains correspond to organizational (*admininstrative*) boundaries, which are not necessarily geographical.

Servers are bootstrapped with pointers to selected peer and parent servers.

Resolvers are bootstrapped with pointers to one or more local servers; they issue *recursive* queries.

Ethernet MACs: Summary

- Global naming context
  - Vendors responsible for unique assignment
- Fixed-width machine-readable name space
- Name resolution by local broadcast
  - Many shortcuts in bridged/switched LANs
  - No "location hint" in an ethernet MAC →mobility
- No controlling authority
  - Easy manageability: plug and play
  - Weak security
- Global name resolution?
  - Need "internetworking".

DNS: Summary

- Human readable names (“symbolic”).
- Hierarchical structure of nested naming contexts
  - Like file system directories
  - Name resolution by pathname traversal
- Each naming context has a controlling authority that resolves names in the context.
- Each parent context has a “secure binding” to the authoritative server for each child context.
- Everyone has a secure link to the “root”.
- Dynamic, distributed, global name service
- Hierarchical structure is simple but problematic.

DNS: The Big Issues

- Who can obtain a new domain name, and by whose authority?
- What about trust? How can we know if a server is authoritative, or just an impostor?
    - What happens if a server lies or behaves erratically? What denial-of-service attacks are possible? What about privacy?
- What if an "upstream" server fails?

DNS: Cost of Hierarchy

- Root servers: the gang of 13
  - Not much diversity there: BIND on Unix
  - “A” root
- Special case of ‘centralized’
  - Think bottleneck
  - Single point of attack and failure
- October 21, 2002
  - DDOS attack against roots

DNS Politics
or,The Cost of Uniformity

- DNS is a global name space.
- That makes it a global political issue.
- History:
  - TLD registry run by Network Solutions, Inc.
  - US government (NSF) granted monopoly, regulated but not answerable to any US or international authority.
  - In 9/98, control transitioned to a more open management structure.
  - Still under US control, with many accusations of power grabs by US industry.

ICANN

Internet Corporation for Assigned Names and Numbers

- Sets prices for domain names
- Accredits domain name registrars
- Accepts/rejects proposed TLDs
- Controls the root servers
- Chartered by US Department of Commerce
- Oversight/control process unclear and controversial
- http://www.internetgovernance.org
- http://www.icann.org

DNS Governance: US Position

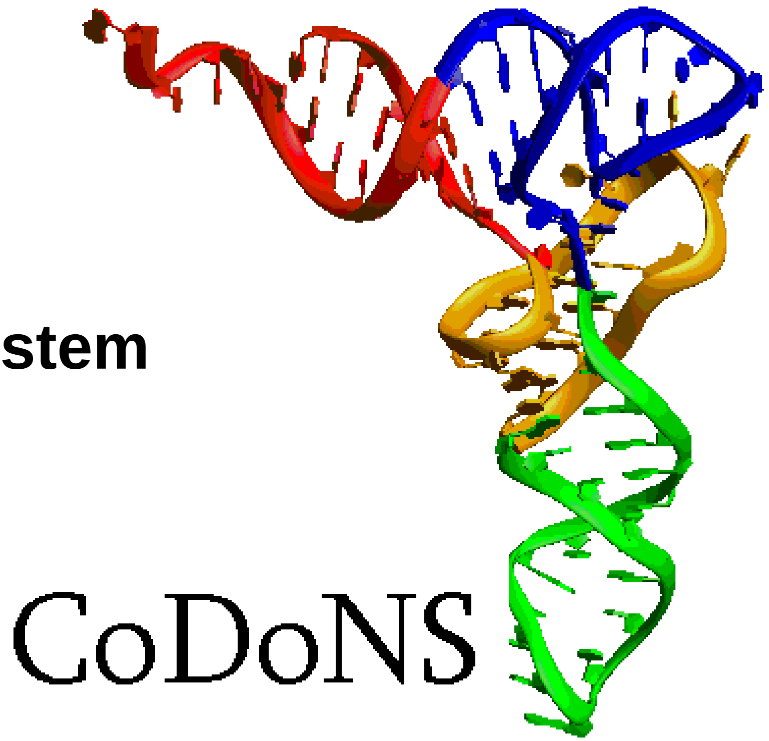
- June 30, 2005 “Statement of Principles”
  - Privatize
  - Internationalize
- Don’t relinquish US government control
  - Unilateral oversight
  - Seen by some as a US strategic asset
- ???
- UN Working Group on Internet Governance (WGIG)

Four Bugs of DNS Governance

Mike O'Dell, from Dave Farber's IP mailing list, on the four bugs in DNS governance.

- The first bug was creating a structure that *needs* governance.
- The second bug was creating a monopoly to own the structure.
- The third bug was creating yet another monopoly to provide "governance".
- The fourth bug is not adopting distributed system technology to render the other three bugs irrelevant.

Cooperative Domain Name System

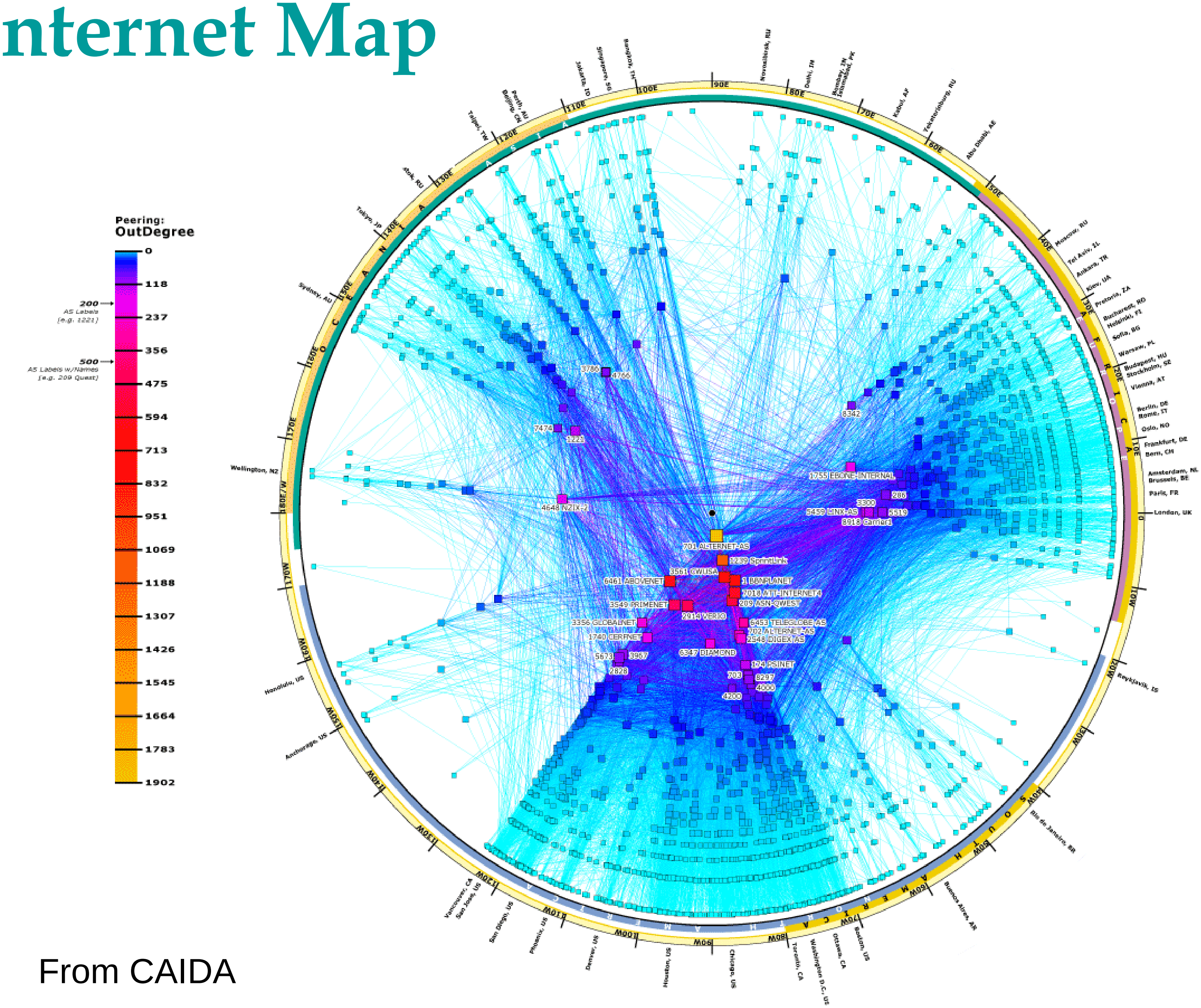


In a recent survey of the legacy DNS which spanned 535000 domains and 164000 name servers, we found that 79% of domainnames rely on two or fewer servers. Worse, 33% of domains have a single bottleneck link whose failure would result in disappearance of that domain. These bottlenecks in DNS make it trivial to launch DoS attacks against selected targets. Further, 20% of DNS servers contain security vulnerabilities that enable attackers to spoof records or block their distribution entirely. The static nameserver hierarchy that forms the foundation of the the legacy DNS system is fragile and vulnerable.

IP Addresses

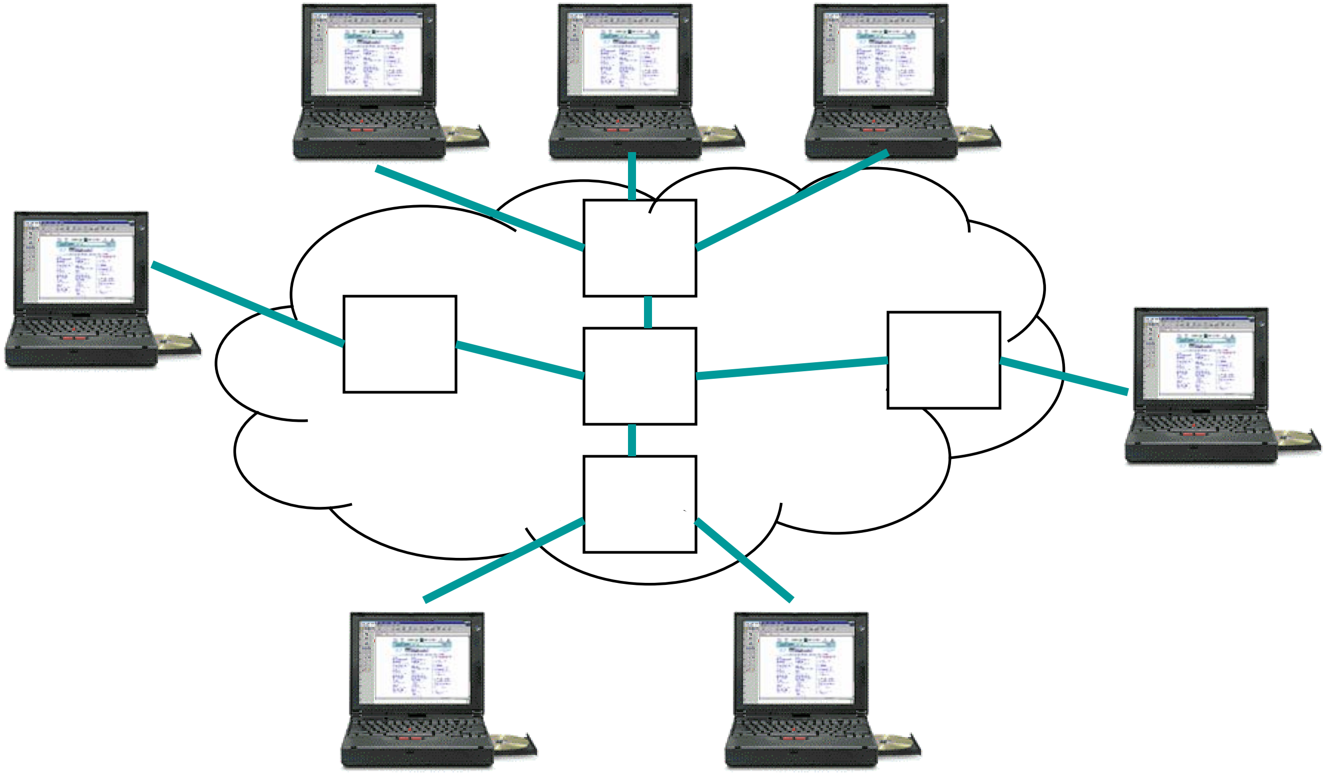
- IP addressing has characteristics of both MACs and DNS….with some new wrinkles.
- Like MACs
  - fixed-width addresses
  - But smaller! 32 bits in IPv4, e.g., 152.3.140.61
  - hierarchical partitioning
- Like DNS
  - Can't get anywhere without them
  - Partitioning matches administrative hierarchy
  - Name incorporates "location" hint

Internet Map



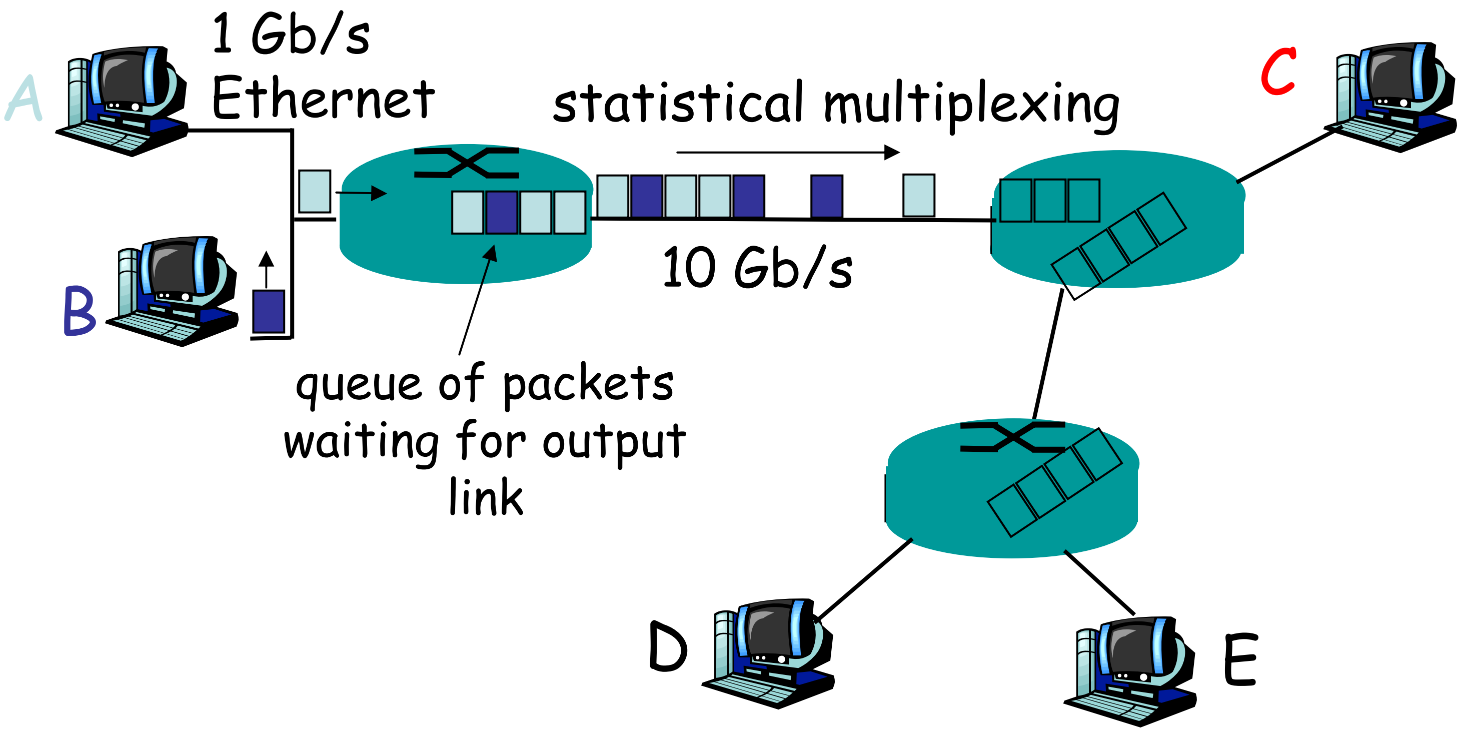
From CAIDA

A Switched Network



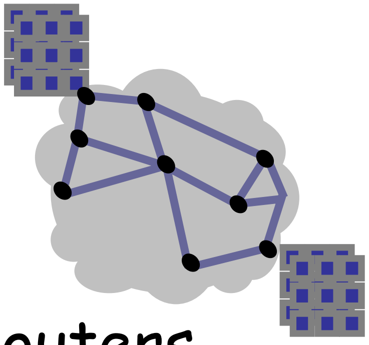
{razor,vahdat}@cs.duke.edu

Network Core: Packet Switching



{razor,vahdat}@cs.duke.edu

IP Routing 101

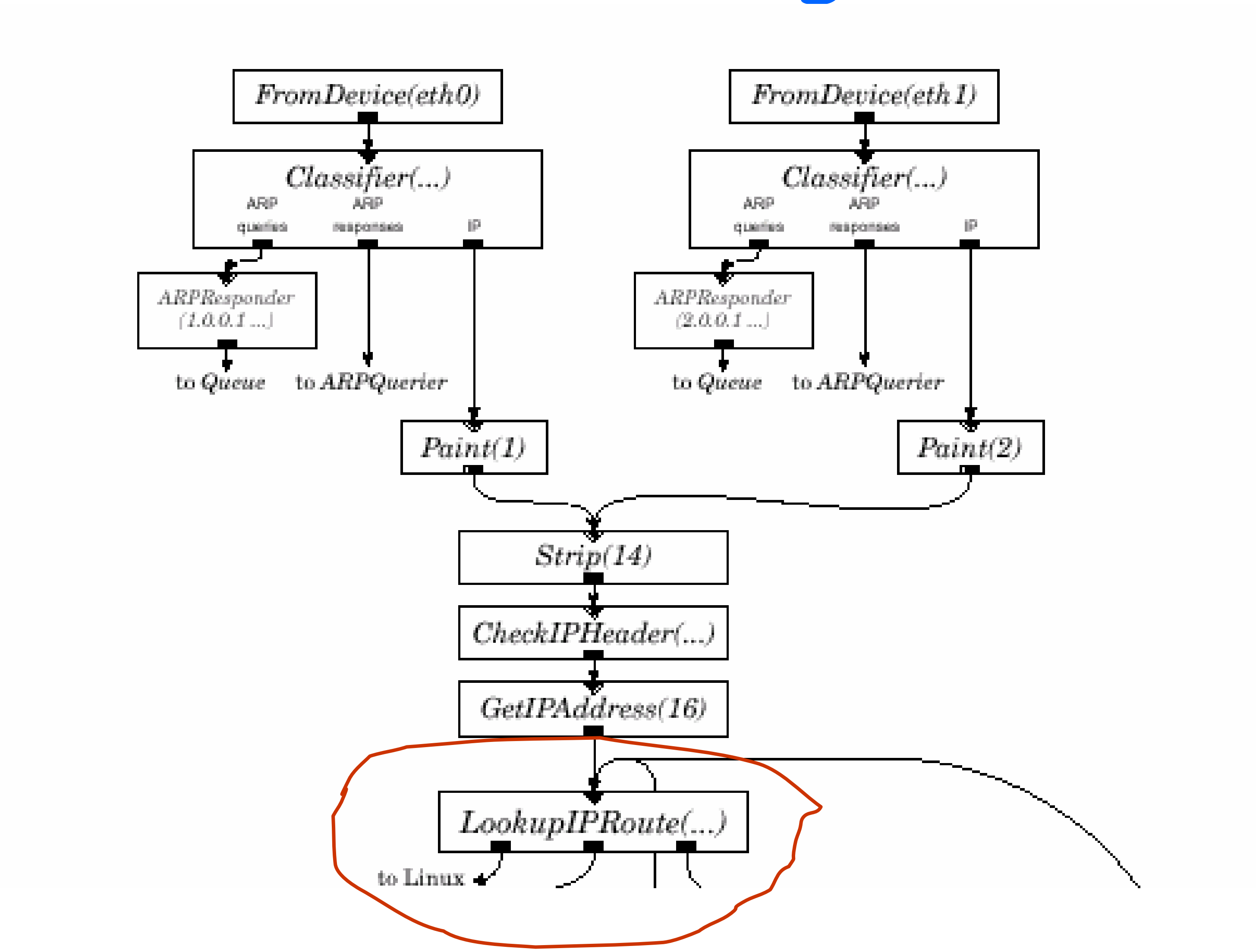


- The Internet is a graph of computers and routers.
  - Hosts on the "edge"
  - Routers in the "core"
- Routers forward packets to next-hop neighbor
  - Receive from interface A into memory.
  - Index local *forwarding table* by dest IP address
  - Retrieve next-hop neighbor
  - Enqueue packet for transmit on interface B.
- Some key issues:
  - How to represent forwarding info compactly?
  - How to build the forwarding tables?

Traceroute cs.unc.edu

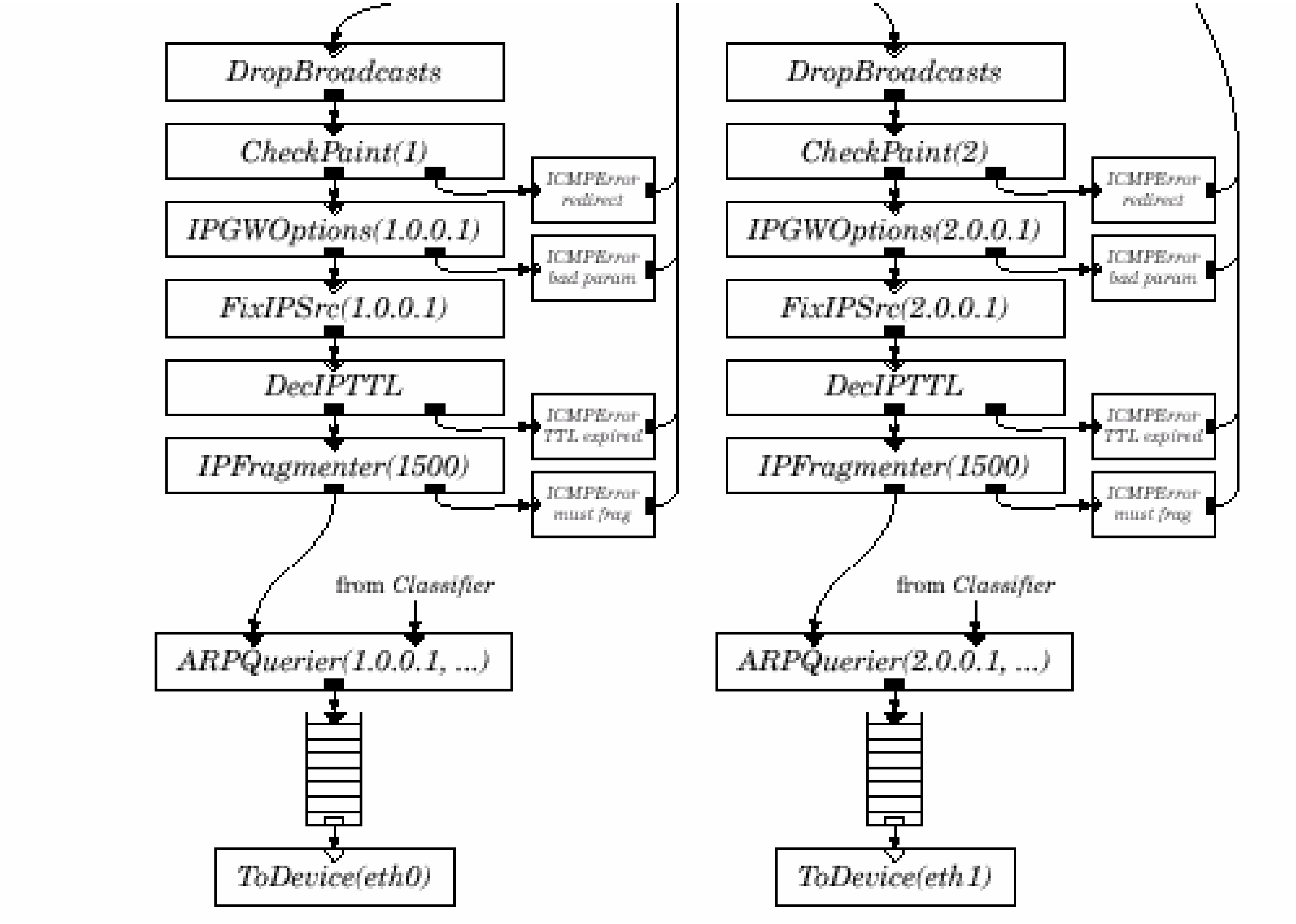
```
traceroute to cs.unc.edu (152.2.131.244), 30 hops max, 38 byte packets
 1  bud (152.3.140.61)  0.193 ms  0.172 ms  0.165 ms

 2  152.3.219.69 (152.3.219.69)  0.227 ms  0.202 ms  0.188 ms

 3  duke7600-roti-vlan201.netcom.duke.edu (152.3.218.209)  0.308 ms  0.326
ms  0.316 ms

 4  rlgh7600-gw-to-duke7600-gw.ncren.net (128.109.70.17)  1.417 ms  1.282
ms  1.123 ms

 5  unc7600-gw-to-rlgh7600-gw.ncren.net (128.109.70.30)  139.069 ms
199.394 ms  3.210 ms

 6  ciscokid.internet.unc.edu (128.109.36.253)  2.621 ms  2.574 ms  2.579 ms

 7  dove.cs.unc.edu (152.2.131.244)  2.737 ms  2.681 ms  2.726 ms
```

IP Routing



From Click

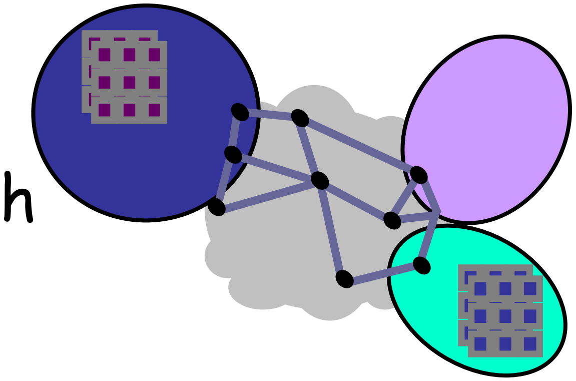
IP Routing



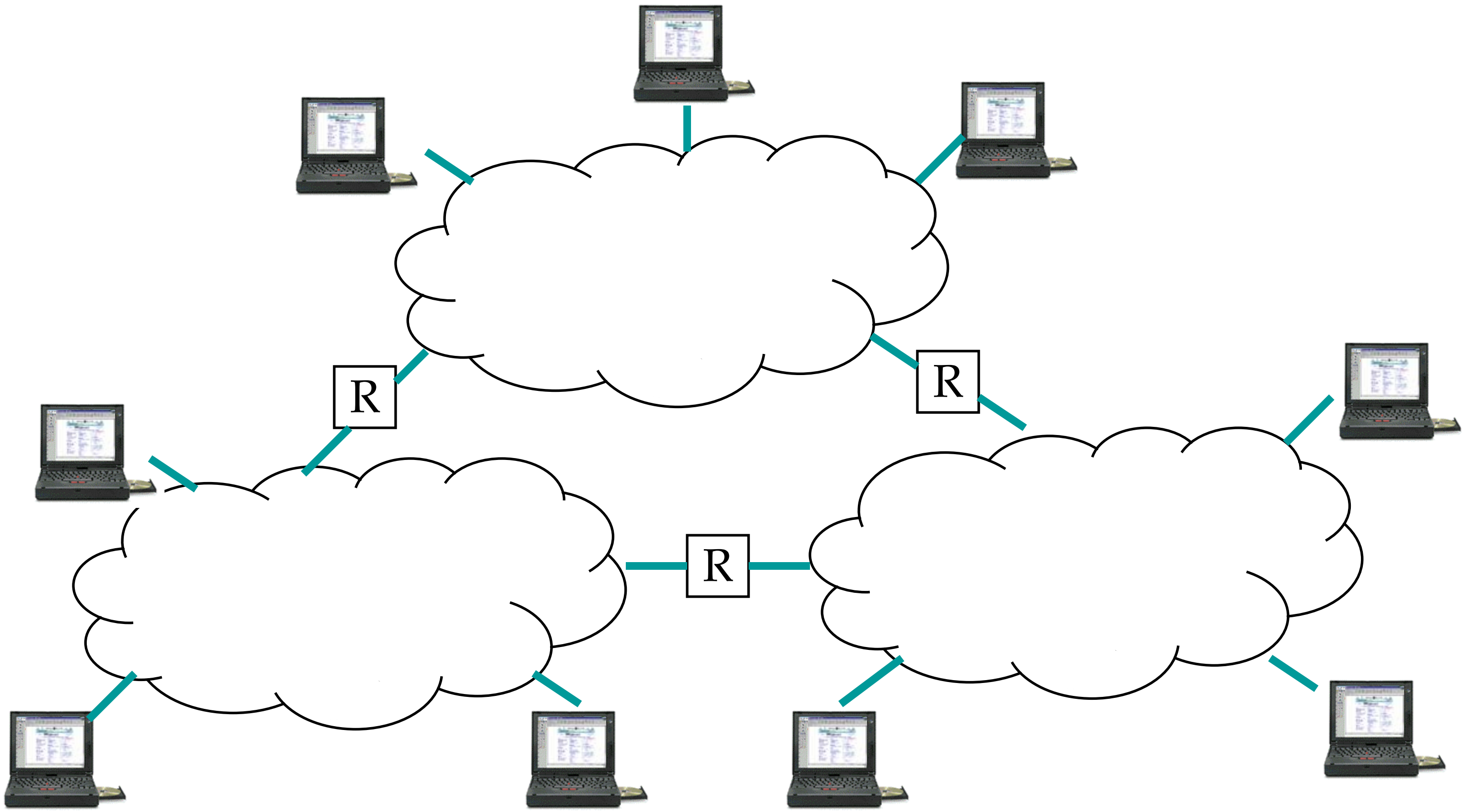
From Click

More IP Routing

- The Internet is really a graph of networks.
  - A set of nodes under common admin authority
  - Stub nets on the edge, transit nets in the core
  - “IP over everything, and everything over IP.”
- Each network is a routing “domain”.
  - Routers are owned by “autonomous systems” (AS).
    - E.g., an ISP
  - Step 1: route to the right domain
  - Step 2: route to the right node
  - interdomain routing is still research
    - Safe, yet adaptive
    - Decentralized, yet coordinated



Interconnected Networks



IP Address Allocation

- IP address is a (network number, host number) pair
- Originally ("classful" addrs), 4 address classes
  - "A": 0 | 7 bit network | 24 bit host (1M each)
  - "B": 10 | 14 bit network | 16 bit host (64K)
  - "C": 110 | 21 bit network | 8 bit host (255)
  - "D": 1110 | 28 bit multicast group #
- Assign net # globally, host # locally
  - IBM, MIT have class A addresses
  - Duke has class B address
- What is a network "prefix"?

IPv4 Address Issues

- We can run out
  - 4B IP addresses; 4B microprocessors in 1997
- We'll run out faster if sparsely allocated
  - Rigid structure causes internal fragmenting
  - E.g., assign a class C address to site with 2 computers
    - Waste 99% of assigned address space
- Need address aggregation to keep tables small
  - 2 million class C networks
  - Entry per network in IP forwarding tables
    - Scalability?

IP Subnets

Subdivide network number into *subnets*

- Variable-length IP address prefix identifies subnet
  - Size of prefix given by a *subnet mask*
  - 32 bits, bit is 1 if it is part of the subnet prefix
- Extends naming hierarchy
  - Separate authorities for name allocation
  - Hides internal net structure from external routers
  - Compact routing tables
- Is destination on the same subnet?
  - Yes →ARP and send to MAC
  - No → route to the subnet by longest-prefix match

Subnet Example

- Subnet mask: 255.255.255.128, IP: 128.96.34.15
  - This says top 25-bits identify the network
  - Class B: 16-bits for network #, 9-bits for subnet
  - Logical AND Host and mask for Subnet #
- 128.96.34.15 AND 255.255.255.128 ➔ 128.96.34.0

Public vs. Private IPs

- In the beginning, the Internet Architecture was based on two simple addressing principles:

  IP address uniquely identifies a host/endpoint.

  Any host can talk to any other host.

- Time passed.
- Things got a little ugly.
- Firewalls, private IP addresses, NATs
  - Contexts!
  - Customization!
  - Collisions!
- Today's Internet resembles a warren of gated communities connected by highways.