M120: DISTRIBUTED SYSTEMS

Naming

*Slides are variant of slides provided by Ken Birman, Jim Kurose, and Keith Ross

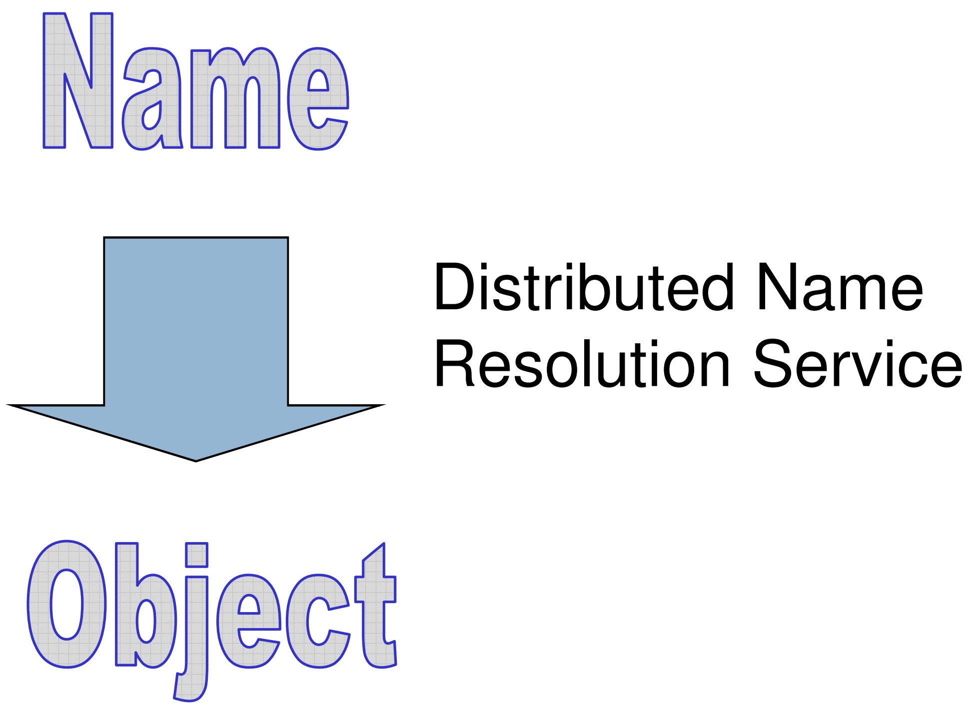
- **“Any problem in computer science can be solved with another layer of indirection”**

  David Wheeler

Naming is a layer of indirection

- What problems does it solve?
  - Makes objects human readable
  - Hides complexity and dynamics
    - Multiple lower-layer objects can have one name
    - Changes in lower-layer objects hidden
  - Allows an object to be found in different ways
    - One object can have multiple names
- A key functionality needed in distributed systems

Names map to objects through a resolution service

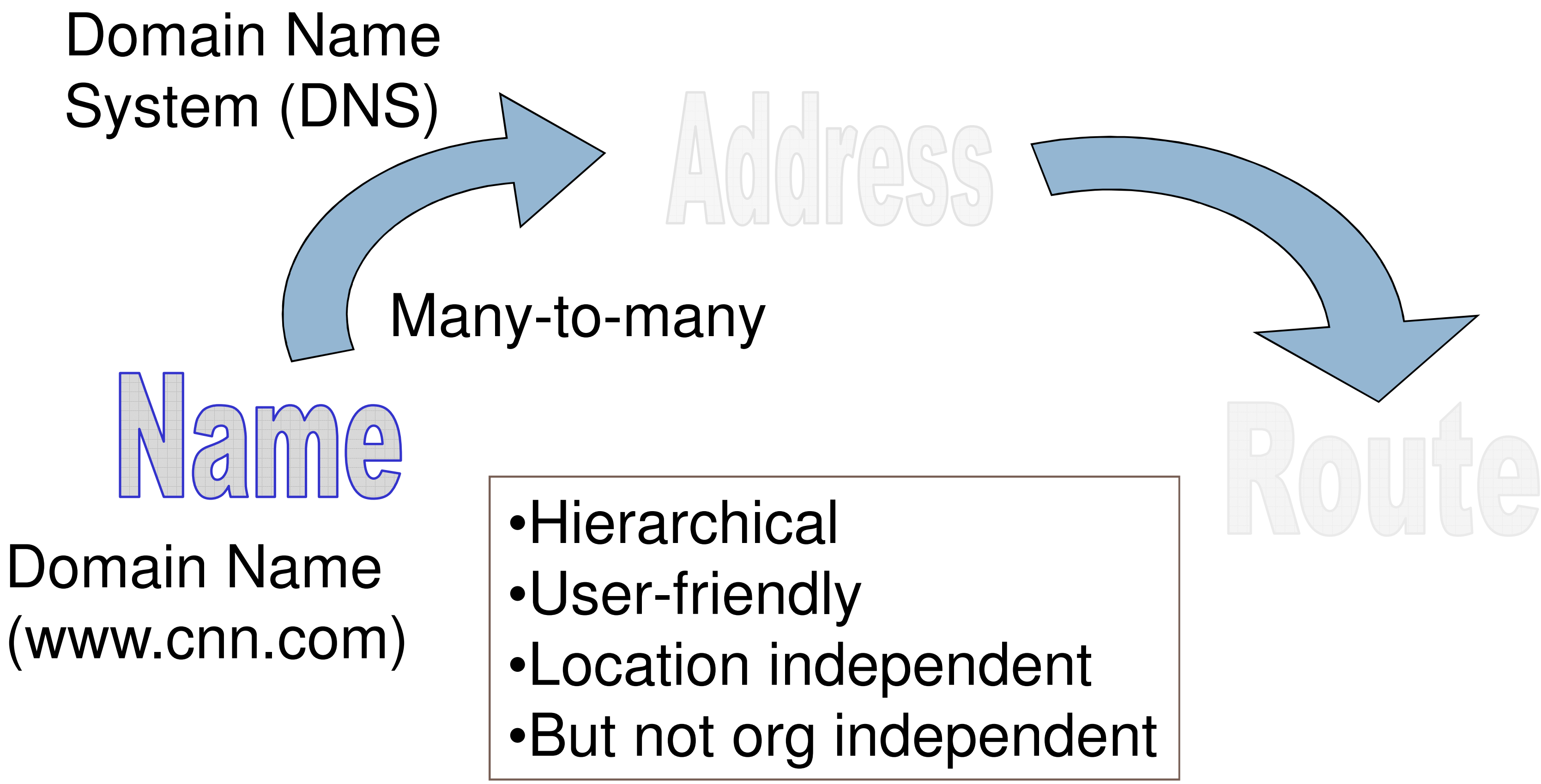


Identifiers vs Locators

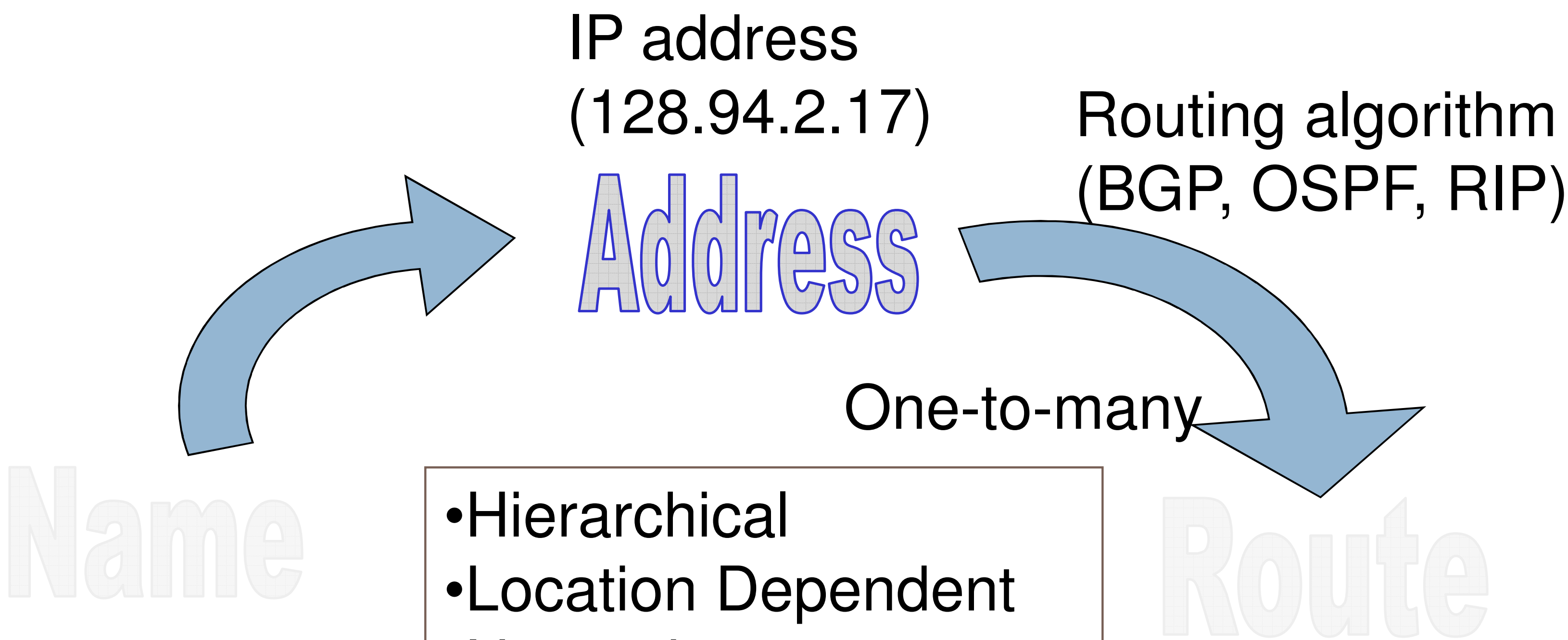
- A name is always an *identifier* to a greater or lesser extent
  - Can be persistent or non-persistent
  - Can be globally unique, locally unique, or even non-unique
- If a name has structure that helps the resolution service, then the name is also a *locator*

Naming in networks

DNS names map into addresses

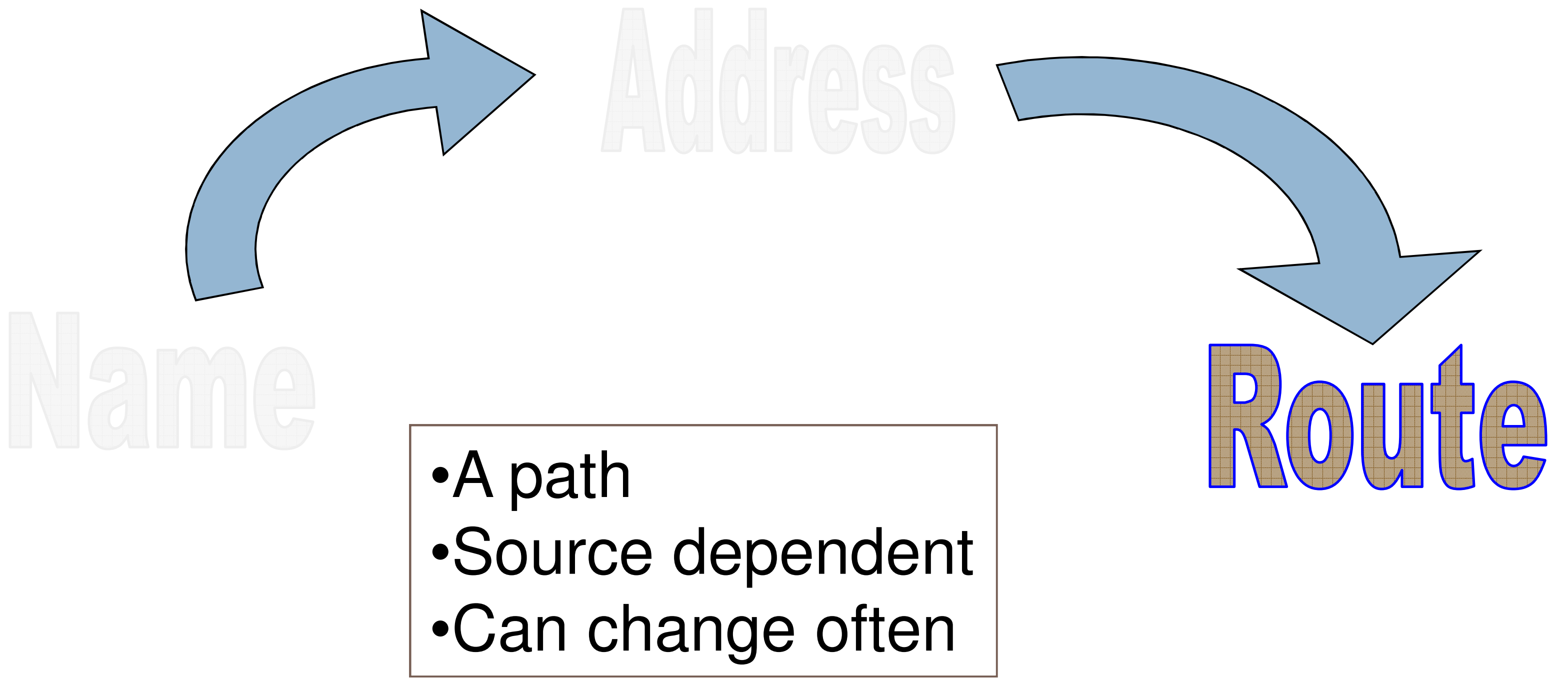


Addresses map into routes



- Hierarchical
- Location Dependent
- Non-unique
- Can change often
- Refers to an interface, not a host

Routes get packets to interfaces



DNS names and IP addresses are identifiers and locators

- Both are typically non-persistent
- Private IP addresses identify only in the context of an IP realm
- Domain names are good identifiers
  - bowser.eecs.harvard.edu identifies a host
  - www.cnn.com identifies a service

Domain Name System (DNS)

- Distributed directory service
- Hierarchical name space
- Each level separated by '.'
  - Analogous to '/' separator in file systems
- One global root
  - Replicated across 13 *logical* root servers
    - Many implemented with cluster of machines
  - For full list: http://root-servers.org/
  - There have been Denial of Service (DoS) attacks on these root servers, none really successful
  - Because of caching, queries to root servers are rare
- DNS maybe only global directory service???

DNS is simple but powerful

- Only one type of query
  - Query(domain name, RR type)
    - Resource Record (RR) type is like an attribute type
  - Answer(values, additional RRs)
- Limited number of RR types
- Hard to make new RR types
  - Not for technical reasons…
  - Rather because each requires global agreement
  - ICANN (Internet Corporation for Assigned Names and Numbers)

DNS is the core of the Internet

- Global directory service
  - Can resolve a name to nearly every computer on the planet
- Global name space
  - Can be the core of a naming or identifying scheme

DNS

People: many identifiers:

- Name, AMKA, ID card#, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Domain Name System:

- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol* endhosts, name servers communicate to *resolve* names (name/address translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge" (hello E2E!)

DNS

DNS services

- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

Why not centralize DNS?

DNS

DNS services

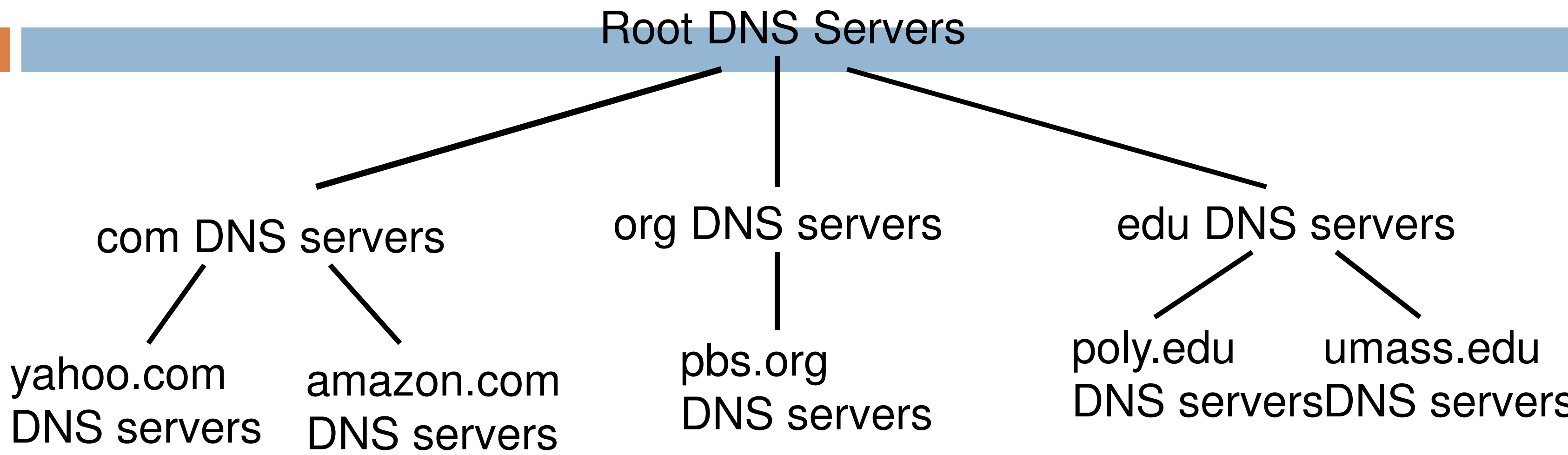
- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

Why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

doesn't *scale!*

Distributed, Hierarchical Database

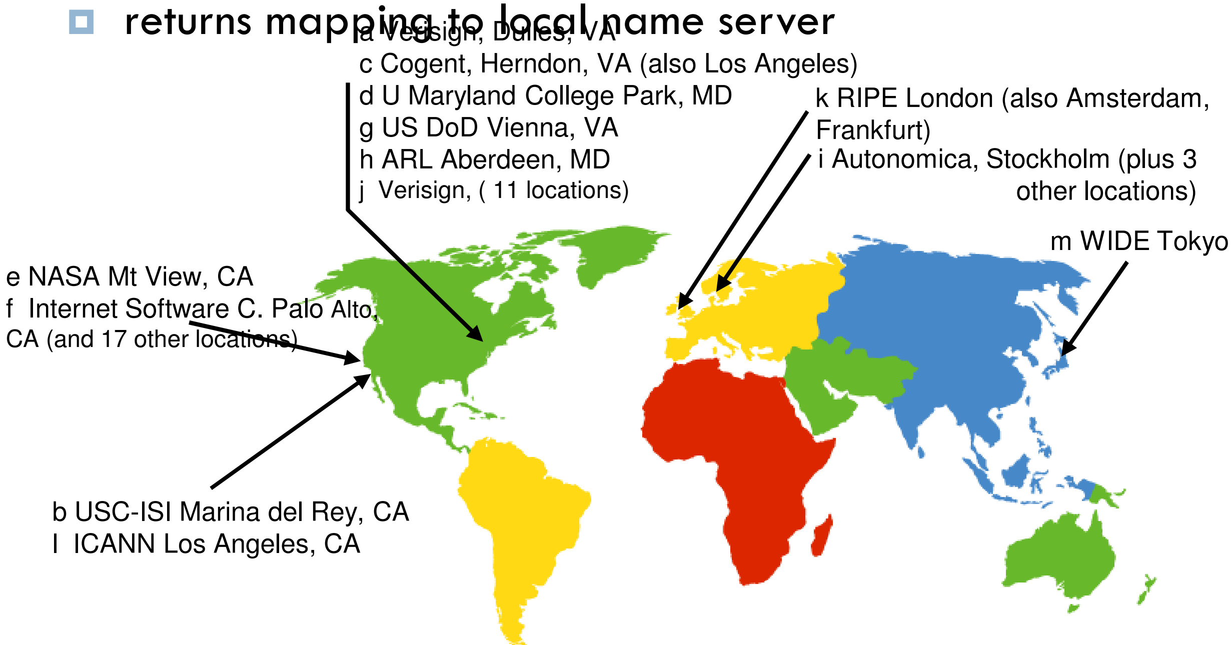


Client wants IP for www.amazon.com; 1st approx:

- Client queries root server to find com DNS server
- Client queries com DNS server to get amazon.com DNS server
- Client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: Root name servers

- contacted by local name server that cannot resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server



13 (logical) root name servers worldwide

TLD and Authoritative Servers

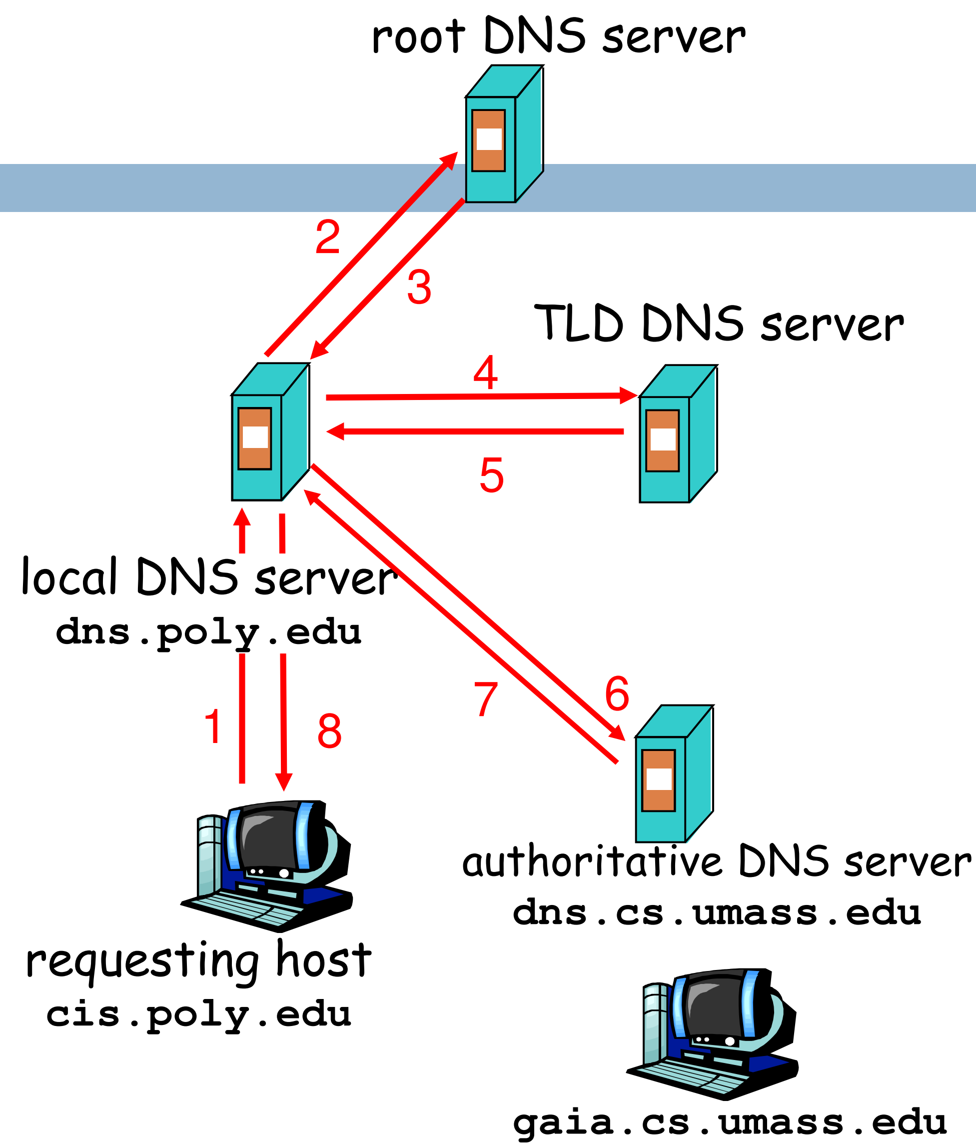
- Top-level domain (TLD) servers: responsible for com, org, net, edu, etc, and all top-level country domains gr, uk, fr, ca, jp.
  - Network solutions maintains servers for com TLD
  - Educause for edu TLD
- Authoritative DNS servers: organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail).
  - Can be maintained by organization or service provider

Local Name Server

- Does not strictly belong to hierarchy
- Each ISP (residential ISP, company, university) has one.
  - Also called “default name server”
- When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a proxy, forwards query into hierarchy

Example

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu
- This is an iterative query.

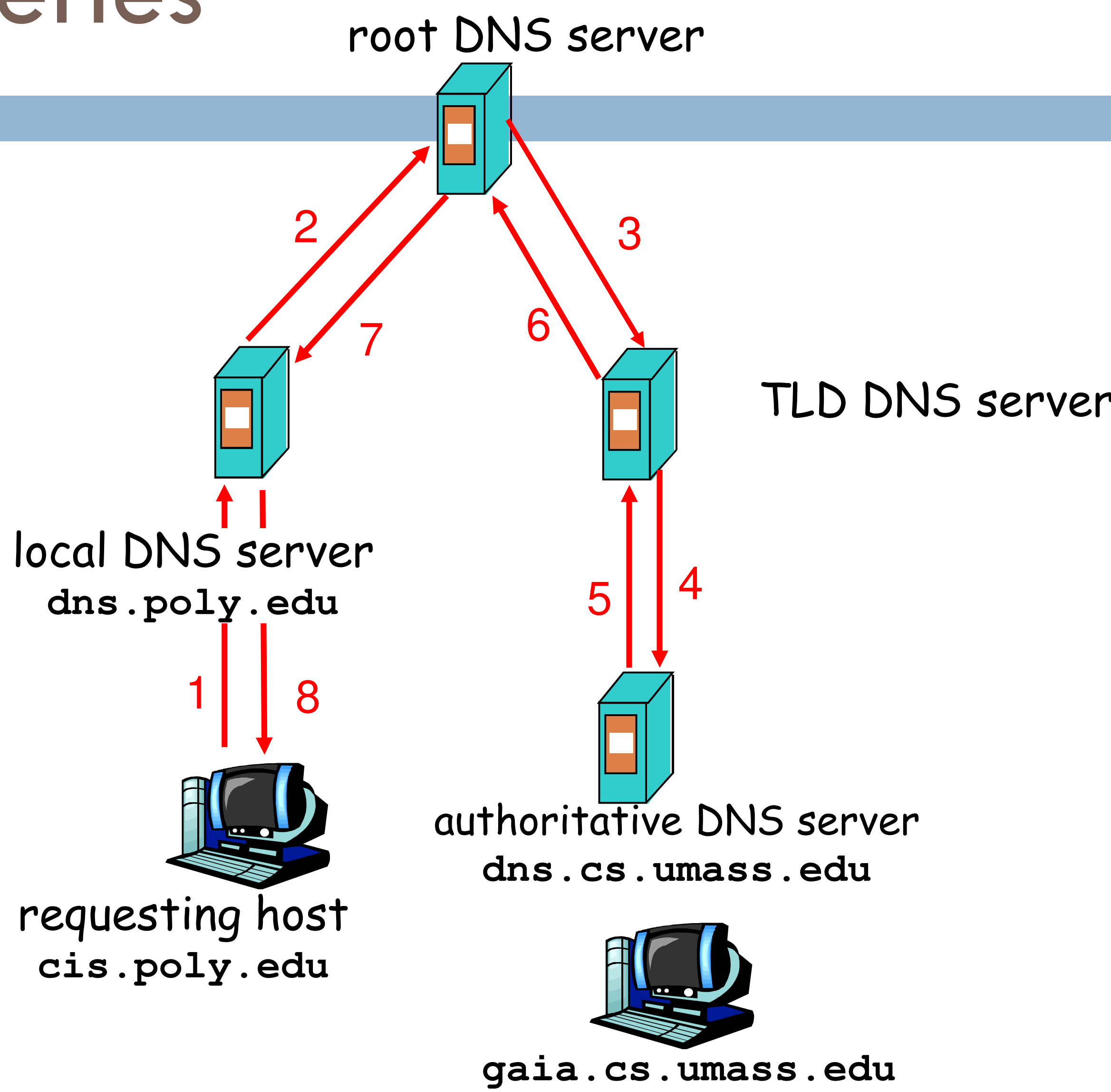


iterated query:

contacted server replies with name of server to contact

"I don't know this name, but ask this server"

Recursive queries



recursive query:

- puts burden of name resolution on contacted name server
- heavy load?

DNS: caching and updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited
- update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

DNS: a distributed database storing resource records (RR)

RR format: `(name, value, type, ttl)`

- Type=A
  - **`name`** is hostname
  - **`value`** is IP address

- Type=NS
  - **`name`** is domain (e.g. foo.com)
  - **`value`** is hostname of authoritative name server for this domain

- Type=CNAME
  - **`name`** is alias name for some "canonical" (the real) name `www.ibm.com` is really `servereast.backup2.ibm.com`
  - **`value`** is canonical name

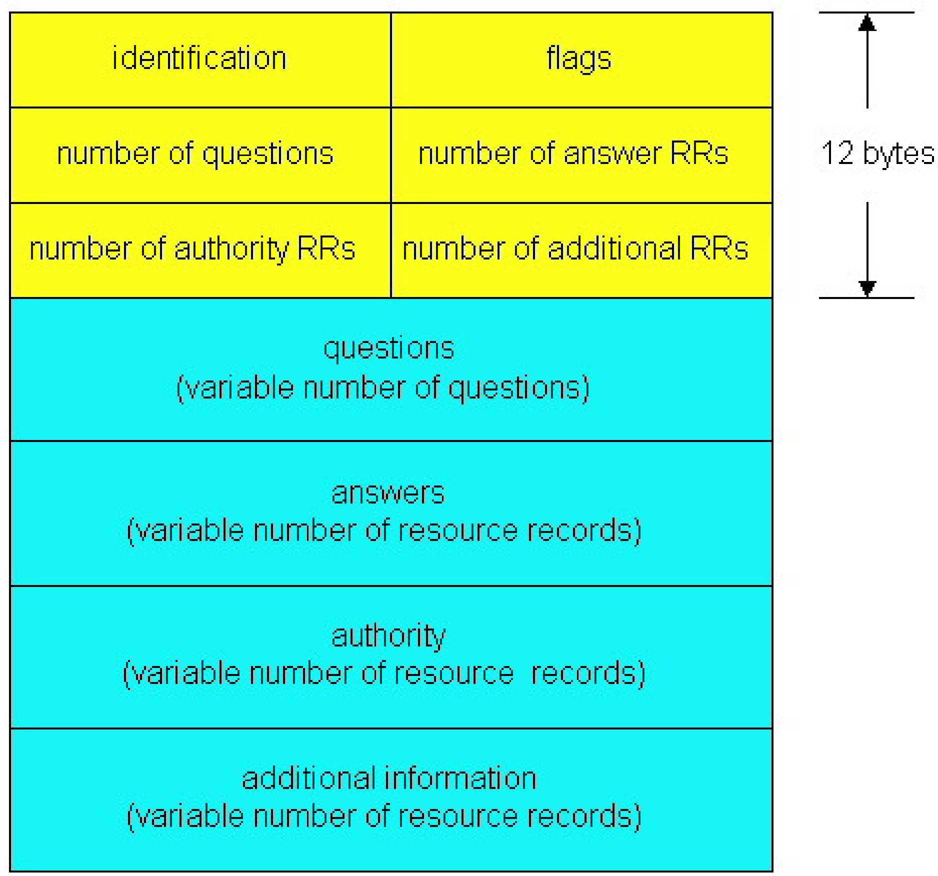
- Type=MX
  - **`value`** is name of mailserver associated with **`name`**

DNS protocol, messages

<u>DNS protocol :</u> *query* and *reply* messages, both with same *message format*

msg header

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



DNS protocol, messages

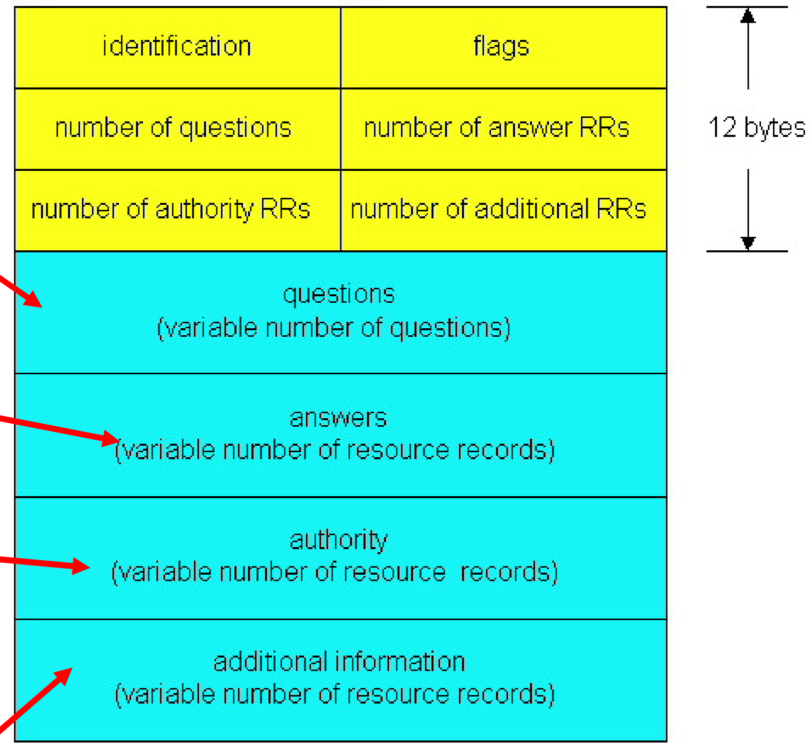
Name, type fields for a query

e.g., H, MX (looking for Mail server of hostname H)

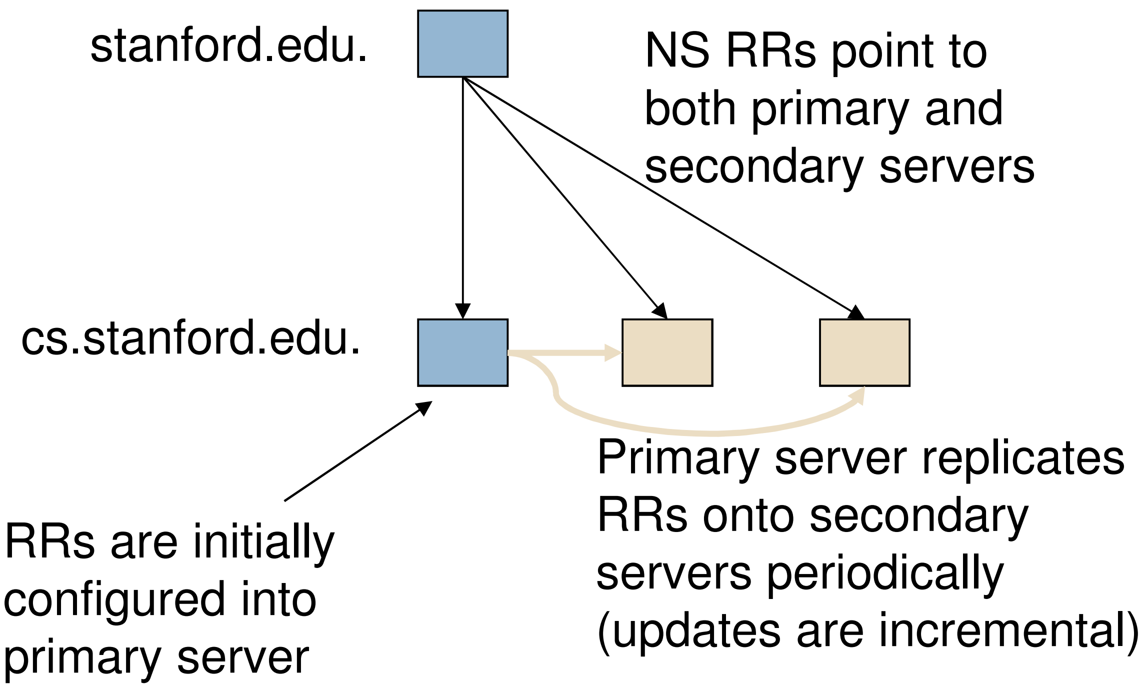
RRs in response to query

records for authoritative servers

additional "helpful" info that may be used (e.g., RR with IP address of Alice's mail server that I queried for)



Primary and secondary servers (within an organization)



Inserting records into DNS

- Example: just created startup "Network Utopia"
- Register name networkuptopia.com at a registrar (e.g., Network Solutions)
  - Need to provide registrar with names and IP addresses of your authoritative name server (primary and secondary)
  - Registrar inserts two RRs into the com TLD server:

```
(networkutopia.com,
  dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1,
  A)
```

- Put in your authoritative server Type A record for www.networkuptopia.com and Type MX record for networkutopia.com
- How do people get the IP address of your Web site?

DNS cache management

- All RRs have Time-to-live (TTL) values
- When TTL expires, cache entries are removed
- NS RRs tend to have long TTLs
  - Cached for a long time
  - Reduces load on higher level servers
- A RRs may have very short TTLs
  - Order of one minute for some web services
  - Order of one day for typical hosts

Why is DNS iterative and not recursive?

- Tanenbaum and van Steen state that recursive is more efficient
  - Better caching characteristics
    - Caches in servers, not just resolvers
  - Smaller response times
- However, high-performance recursive server much harder to implement
  - Maintain state for thousands of concurrent queries
  - Manage cache
- Recursive server prone to DoS attacks

LDAP is another popular distributed directory service

- Richer and more general than DNS
  - Has generalized attribute/value scheme
  - Can search on attribute, not just name
- Simpler and more efficient than a full relational database
- Not a global directory service, though namespace is global
  - Its predecessor, X.500, was meant to be
  - But “local” LDAP services can point to each other
- Commonly used for personnel RR databases, subscriber databases

URLs, URNs, and URIs

- Uniform Resource <Locator, Name, Identifier>
- URL tells a computer where and how to reach a resource
  - These came first
- URN is a true identifier
  - Unique, persistent, location-independent
  - E.g., urn:isbn:054140523
  - urn:ietf:rfc:3187
- URI refers to both URLs and URNs
  - Defines syntax for current and future URLs and URNs
- *For now we only really care about URLs*

URL

- Consists of:

<scheme>:<scheme-specific-part>

URL

- Consists of:

<scheme>:<scheme-specific-part>

A protocol

Information the protocol needs

URL examples

- HTTP (web)
  - http://www.cnn.com/news/story.html
- Email
  - mailto://mema@di.uoa.gr
- Newsgroups
  - news:cornell/class/cs514
- SIP (Session Initiation Protocol)
  - sip://service@phone.verizon.com
  - App-layer signaling protocol for multimedia sessions with 2 or more participants

Note the central role of DNS

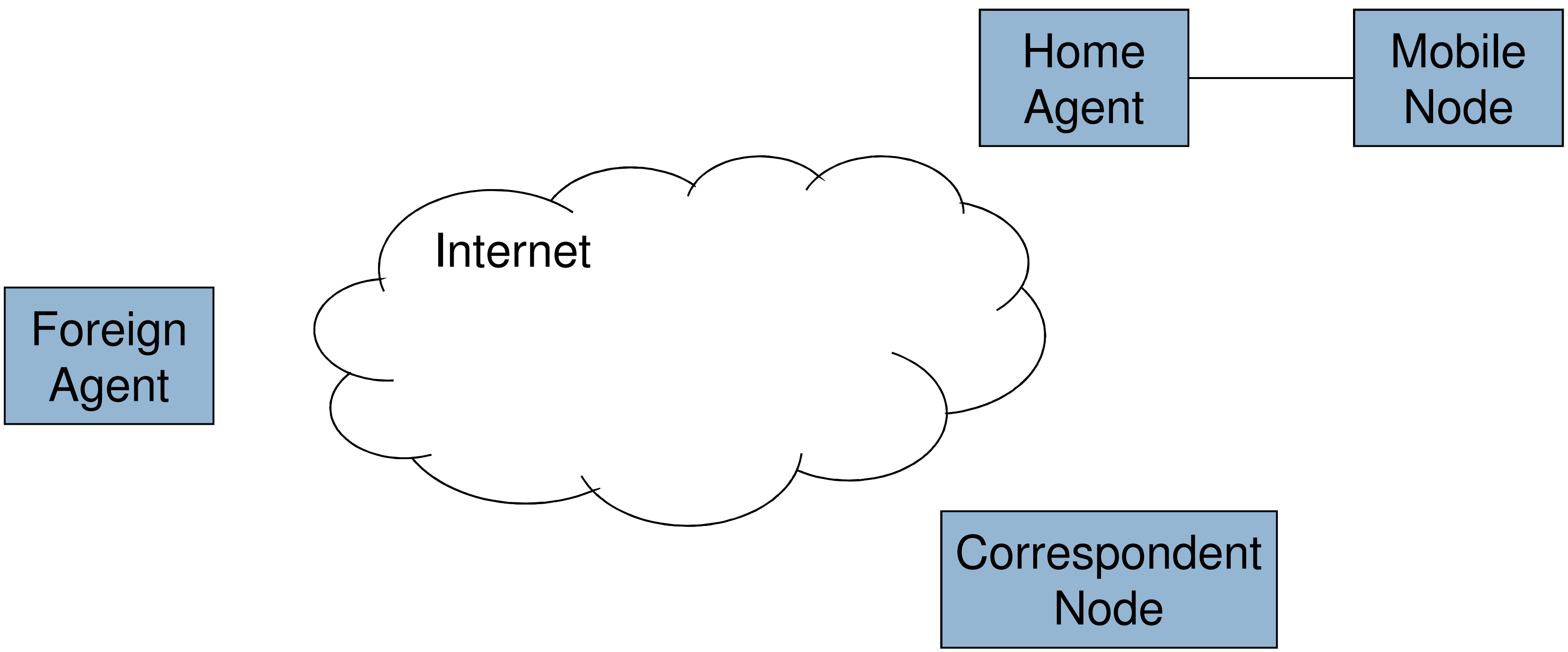
- HTTP (web)
  - http://**www.cnn.com**/news/story.html
- Email
  - mailto://mema@**di.uoa.gr**
- Newsgroups
  - news:cornell/class/cs514
- SIP (Session Initiation Protocol)
  - sip://service@**phone.verizon.com**

Locating mobile entities

- What is a mobile entity?
- From naming perspective, it is an entity whose address changes often
- This doesn't require physical mobility!
  - Every time you dial up/connect from home, you may get a new address
- So, "mobility" existed well before laptops became common
  - Though laptops create more mobility
- What happens if I change IP addr? Should DNS be notified?
  - Yes, via DNS update standard
  - No, Mobile IP instead

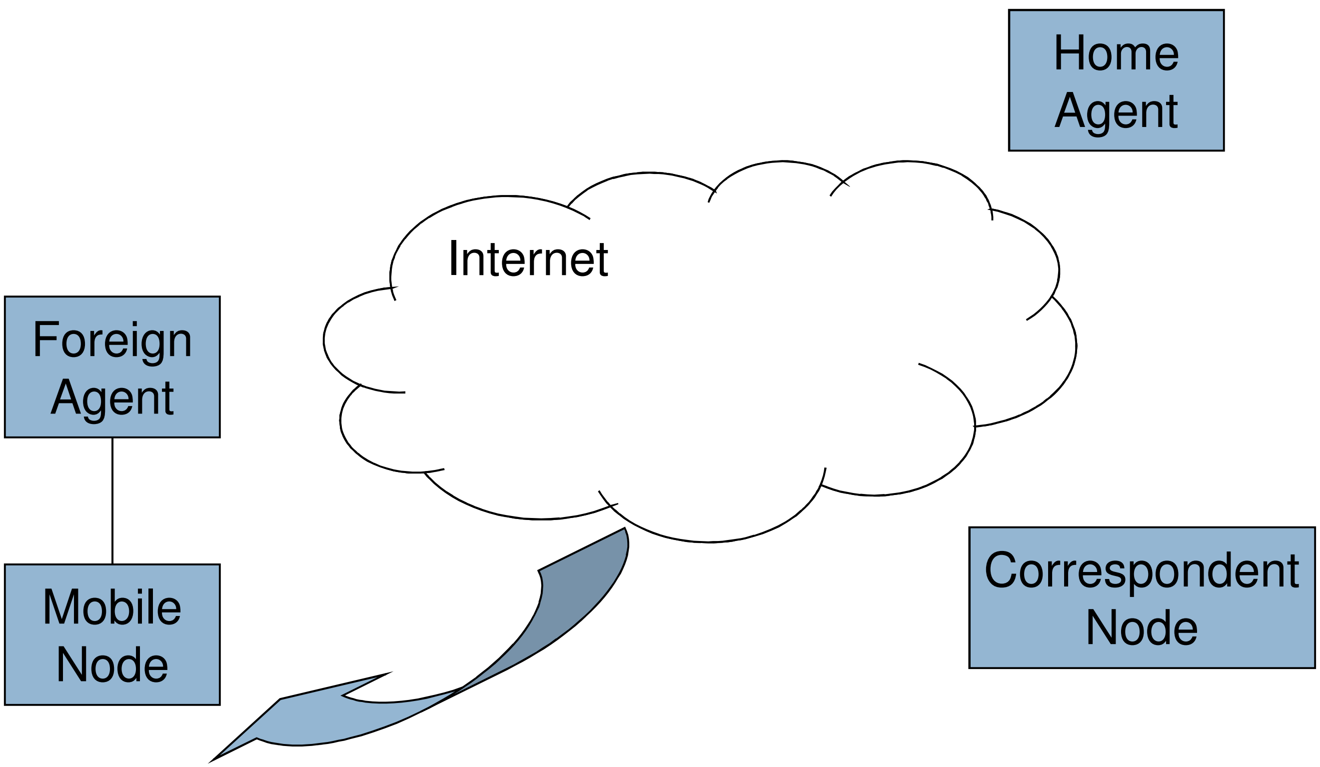
Mobile IP uses an IP-level registration

Mobile Node has a stable home address at its home network



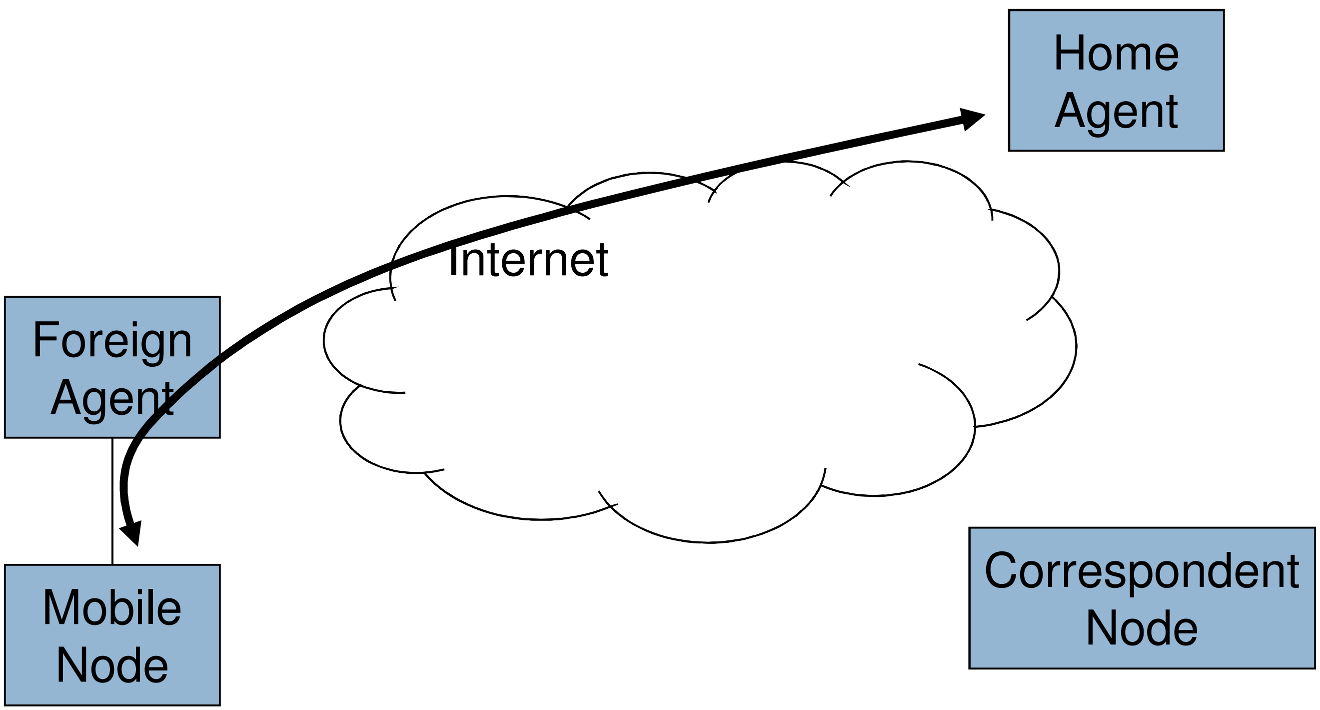
Mobile IP uses an IP-level registration

Mobile Node moves to foreign network, gets a Care-of Address



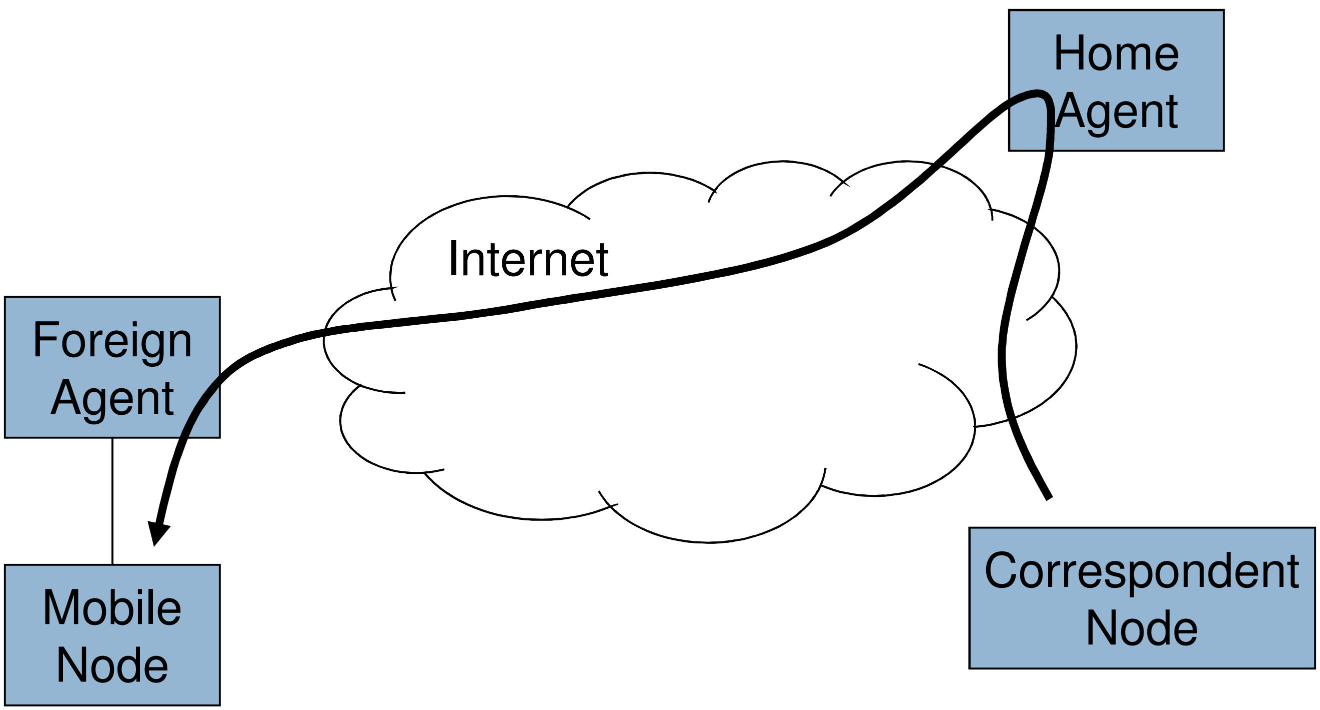
Mobile IP uses an IP-level registration

Mobile Node registers with Home Agent, creates IP tunnel

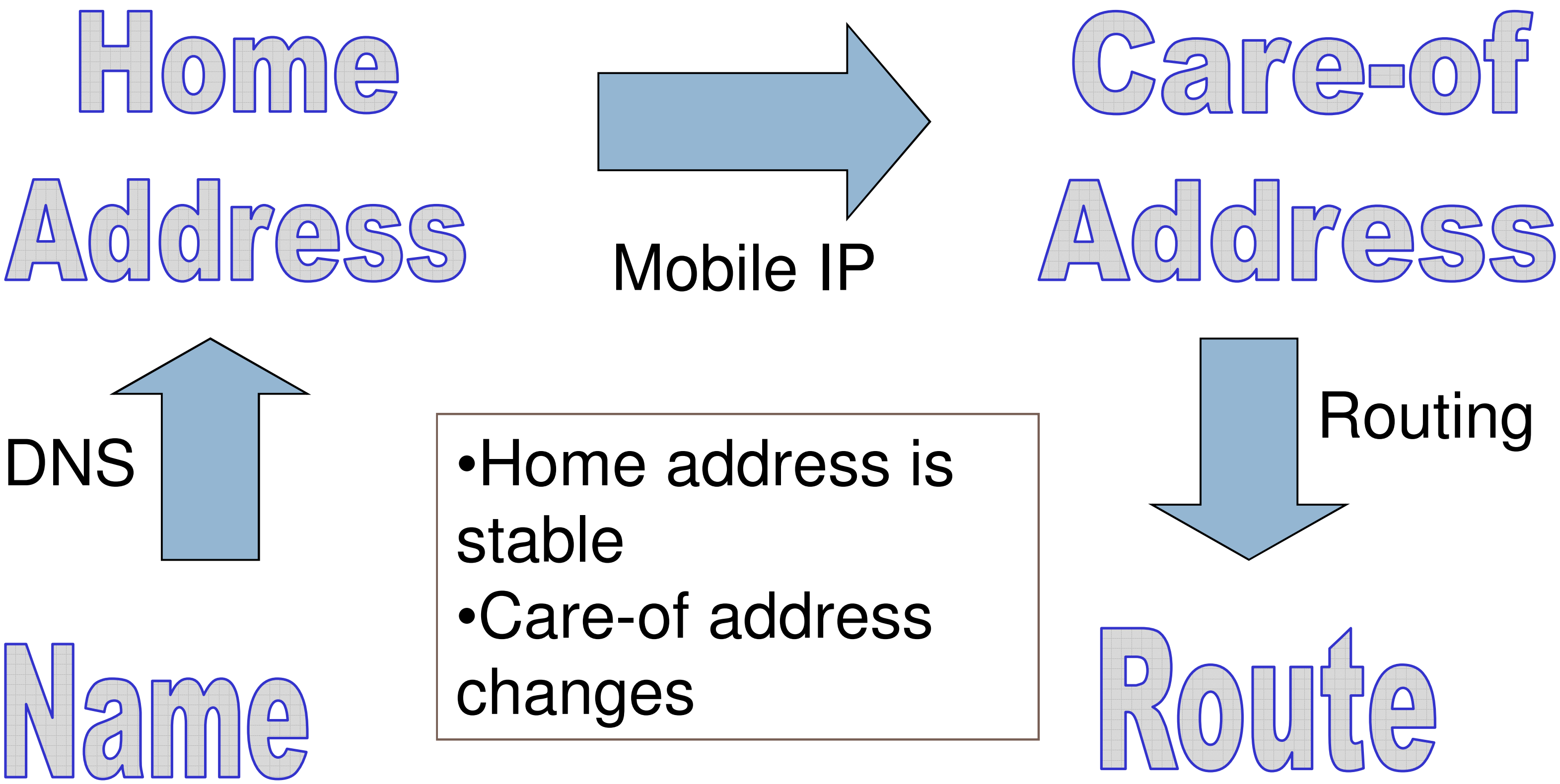


Mobile IP uses an IP-level registration

Connection initiated by Correspondent Node will be tunneled to Mobile Node



Mobile IP adds a layer of indirection



This solution comes with some cost – kernel-level changes, cost to tunnel communication for transparency, etc.

Is mobility a problem for DNS?

- Not really
  - Even though DNS was designed with relatively stable IP addresses in mind
- Because mobility only affects leaf DNS servers
  - Recall: A RR TTL is short, but NS RR TTL is long
- Note: *non-mobile* web server's A RRs often have very short TTLs
  - To allow quick failover to another web server
  - So DNS already handling dynamism

Is mobility a problem at all?

- Less than you'd think
- Most mobile systems are clients; servers are rarely mobile
  - Clients are initiators of connections, not recipients
  - Therefore, there is no client locating problem
- What about email, instant messaging, and VoIP (Voice over IP)?
  - Clients receive emails, instant messages, and phone calls

Application specific registration as a mobility solution

- To receive email, client connects to an email server
- To do instant messaging, client registers with an IM server
- To do VoIP, client registers with a SIP server

This is an adequate solution to 90% of mobility issues

  - This is why Mobile IP hasn't gotten traction (i.e. Microsoft has not implemented it)

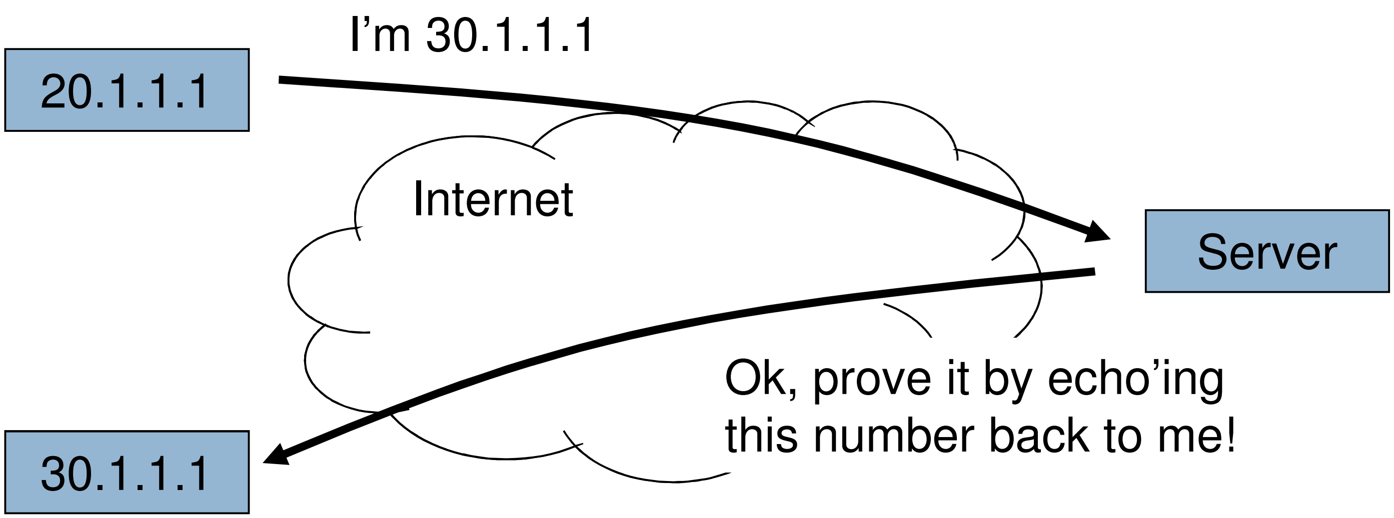
Client identification

- Servers cannot locate clients, but often must be able to identify them
- HTTP cookies serve this role
- HTTP cookies also contain many attributes about the client or session
- They also typically contain some kind of signature
  - To prevent tampering

Identifiers must be made hard to spoof

- That is why driver's licenses have pictures and credit cards have signatures
- In networking, two ways:
  1. Identifier is also a locator
     - Reverse routability
  2. Some kind of secret-protected signature

Reverse routability: DoS and Mobile IP



Since challenge doesn't go back to 20.1.1.1 (i.e. is not reverse routable), 20.1.1.1 cannot spoof 30.1.1.1

Summary of Lecture

Introduction to Naming

- In DS, we need to be able to identify entities, then use their id to locate them
- Naming basics:
  - Names, Addresses, Routes
  - Identifiers and Locators
- DNS is *the* global directory service
  - LDAP is a popular local directory service
- URLs build on DNS (and also URIs and URNs)
- Mobility is not much of a problem
- Identifiers must be hard to spoof
  - Reverse routability, cryptographic signatures