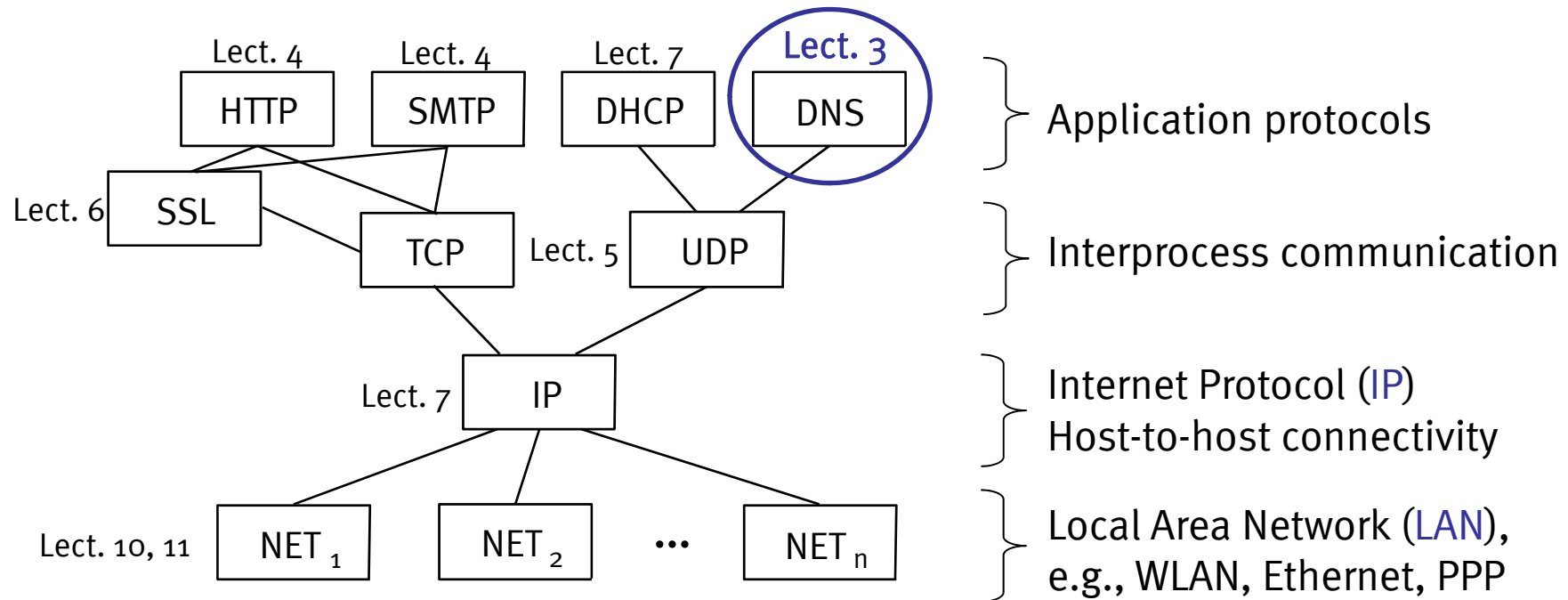




WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

Computer Networks, Winter Term 2009/2010

Domain Name System (DNS)

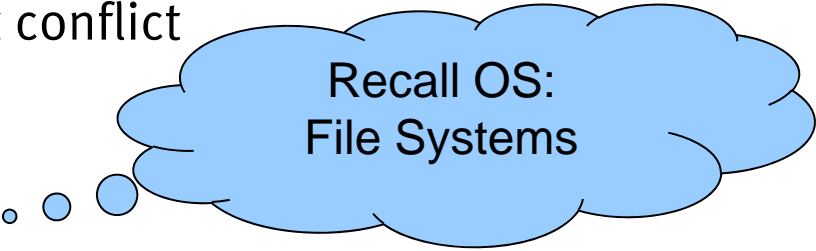


- DNS lookup is first step in Internet communication
 - Get IP address for DNS name

- How to interpret names like `xlx.uni-muenster.de`?
- What is the DNS?
 - How does it work?

- Names
- DNS
- Name Resolution
- Functionality and Examples
- DynDNS

- Sets of possible names
 - Flat or hierarchical
 - Potentially infinite
- Often define structures
 - Similar name parts without conflict
 - Grouping of related names
 - Typically directed graphs
 - Leafs: Named entities
 - Inner nodes: Directories
- Enable transparent restructuring



Recall OS:
File Systems

- Character string to identify entities
- RFC 3986, <http://tools.ietf.org/html/rfc3986>
- Examples from RFC 3986 (DNS names in blue)
 - ftp://[ftp.is.co.za](ftp://ftp.is.co.za)/rfc/rfc1808.txt
 - http://www.ietf.org/rfc/rfc2396.txt
 - ldap://[2001:db8::7]/c=GB?objectClass=one
 - mailto:John.Doe@[example.com](mailto:John.Doe@example.com)
 - news:comp.infosystems.www.servers.unix
 - tel:+1-816-555-1212
 - telnet://192.0.2.16:80/
 - urn:oasis:names:specification:docbook:dtd:xml:4.1.2

- Uniform
 - Different types of IDs under consistent format
- Syntax for absolute URIs
 - `<scheme>:<scheme-specific-part>`
 - scheme-specific-part often structured:
`<scheme>://<authority><path>?<query>`
(e.g., `http://www.google.de/search?q=uri+urn+url`)
- Subsume URLs and URNs

- Clarification at <http://tools.ietf.org/html/rfc3305>
- Uniform Resource Locator (URL)
 - “URL is a useful but informal concept”
 - Identification of web resources via primary access mechanism
 - Network location, address of access point
 - Scalable
 - Address, thus potentially invalid
- Uniform Resource Name (URN)
 - Permanent, location independent name of web resource
 - Registration of URN and URL for resource with URN-service
 - urn:... (RFC 2141)
 - E.g., urn:nbn:de:1111-200606299
 - Deutsche Nationalbibliothek
 - <http://www.nbn-resolving.de/>

- Names
- **DNS**
- Name Resolution
- Functionality and Examples
- DynDNS

- Specified in RFC 1034 and RFC 1035
 - Updated/augmented by various other RFCs
 - <http://www.dns.net/dnsrd/rfc>
- DNS is an application and a protocol

- User's view

- Function or OS call to invoke local **resolver**
 - Input: Domain name (e.g., xlx.uni-muenster.de)
 - Output: IP address (e.g., 128.176.159.171)

- Resolver's view

- Domain system composed of unknown number of **name servers**

- Name server's view

- Domain system consists of **zones**
 - Some cached locally
- Servers respond to (and issue) **queries**
 - Formats of names, queries, and messages specified by protocol

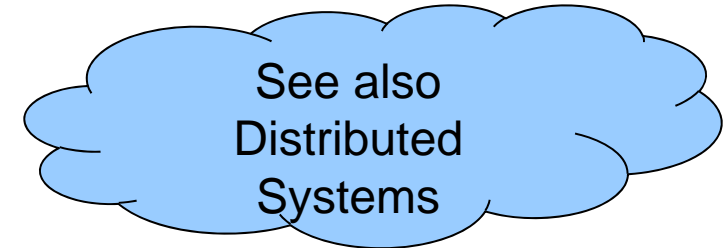
- DNS follows **request/reply** communication pattern
 - Queries and answers are network messages
 - “Small,” fit into single IP datagram
- Delivery either via UDP or TCP
 - DNS uses well-known **port 53**
 - Typically, UDP is used
 - Best effort service model
 - Why is that good enough?
- **No** security mechanisms
 - Cache poisoning, ID spoofing
- DNSSEC
 - <http://www.dnssec.net/rfc>



- **dig** (see example below) or **nslookup**
- **nslookup** (monitor with Wireshark, see following slide)
 - Query for IP address
 - `www.gmail.com`
 - Query for mail server
 - `set type=MX`
 - `gmail.com`
 - Query for name server, ask specific server
 - `set type=NS`
 - `gmail.com 128.176.0.12`
- Further info on domain names: **whois**
 - `whois gmail.com`

- <http://www.wireshark.org/>
- Monitoring and analysis of network traffic
- Capture Options
 - Interface
 - Promiscuous or not
 - Capture Filter
 - Monitoring of network traffic based on libpcap
 - See http://www.tcpdump.org/tcpdump_man.html for detailed syntax
 - Three qualifiers: **type** (host, net, port), **dir** (src, dst), **proto** (ip, tcp, udp, arp, ether, ...)
 - Boolean combinations with “and”, “or”, “not”, “(…)”
 - port 53
 - All traffic involving source or destination port 53
 - host dbms.uni-muenster.de
 - Either source or destination host has given name
 - Also IP address instead of name
 - dst host 128.176.0.12 and udp dst port 53
- Beware: Security **risks**
 - Capture as admin, long history of exploitable holes
 - May capture to file (tcpdump, dumpcap) as admin, analyze as user

- DNS names
 - dbms.uni-muenster.de
 - uni-muenster.de
 - de (Country Code Top Level Domain)
 - . (Root)
 - Reserved names: RFC 2606 (e.g., .localhost, .example.com)
- DNS is a distributed **name service**
 - Manages attributes (e.g., IP addresses) for DNS names
 - Name structure reflects administrative structure of Internet
 - Various registries (e.g., DENIC for .de)
- Scalable (today's Internet)
 - Caching
 - Partitioned database
- Fault tolerance via replication



- Assumptions

- Identifiable resources
- State stable for some time, valid for different clients

- Cache contains copies of resources

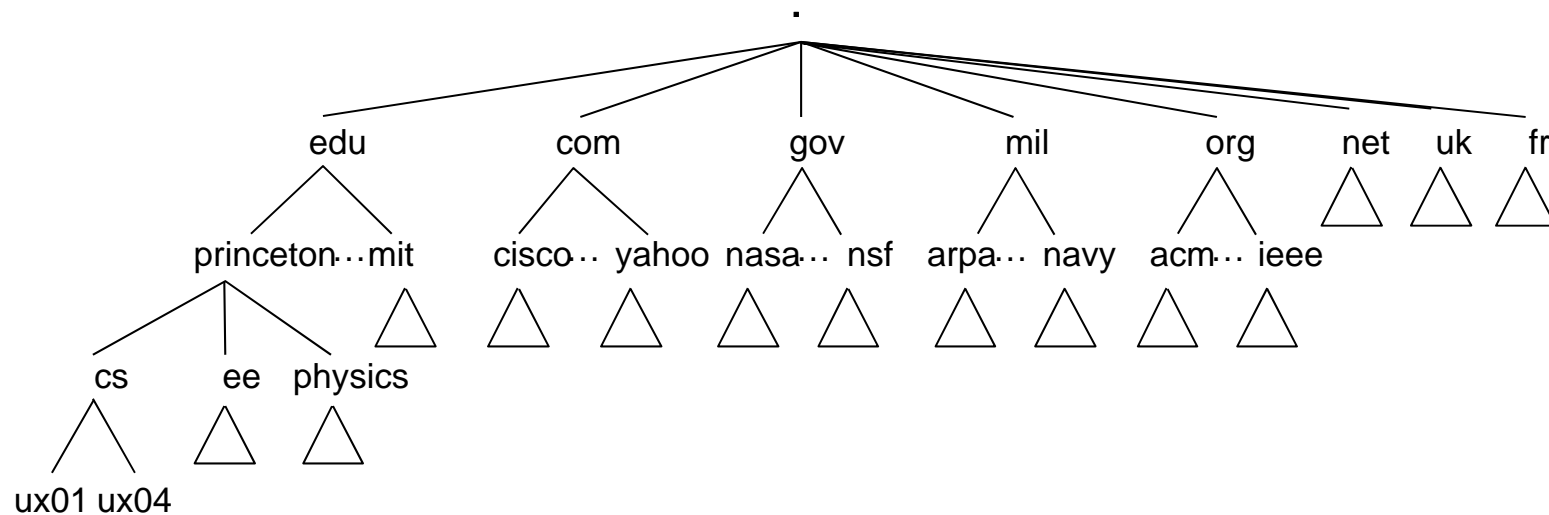
- If cache contains requested resource (cache hit), it is served immediately
- Otherwise (cache miss), the resource is requested from the server

- Caches reduce

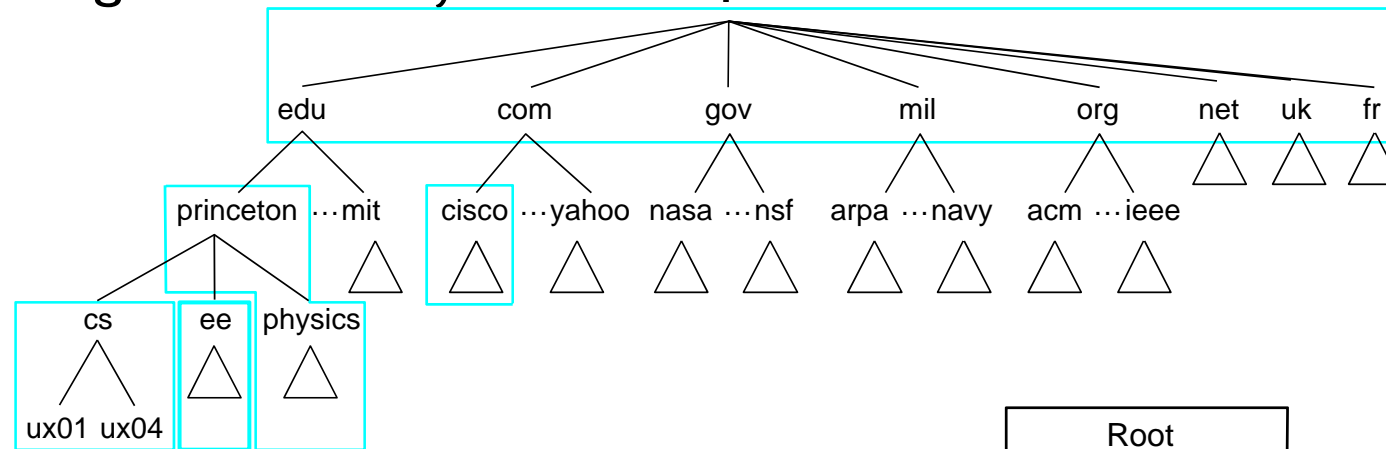
- Redundant transfers
- Latency
- Bottlenecks
- Server load

- (Recall dissemination pattern)

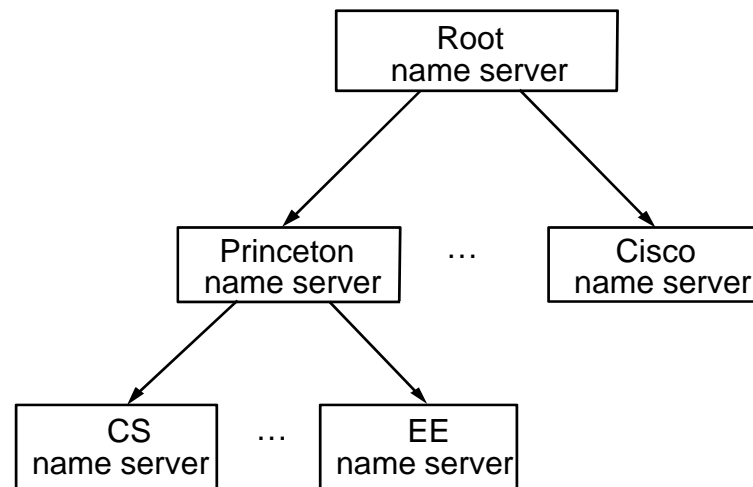




- Partitioning of hierarchy in zones



- Each zone implemented by two or more **name servers**

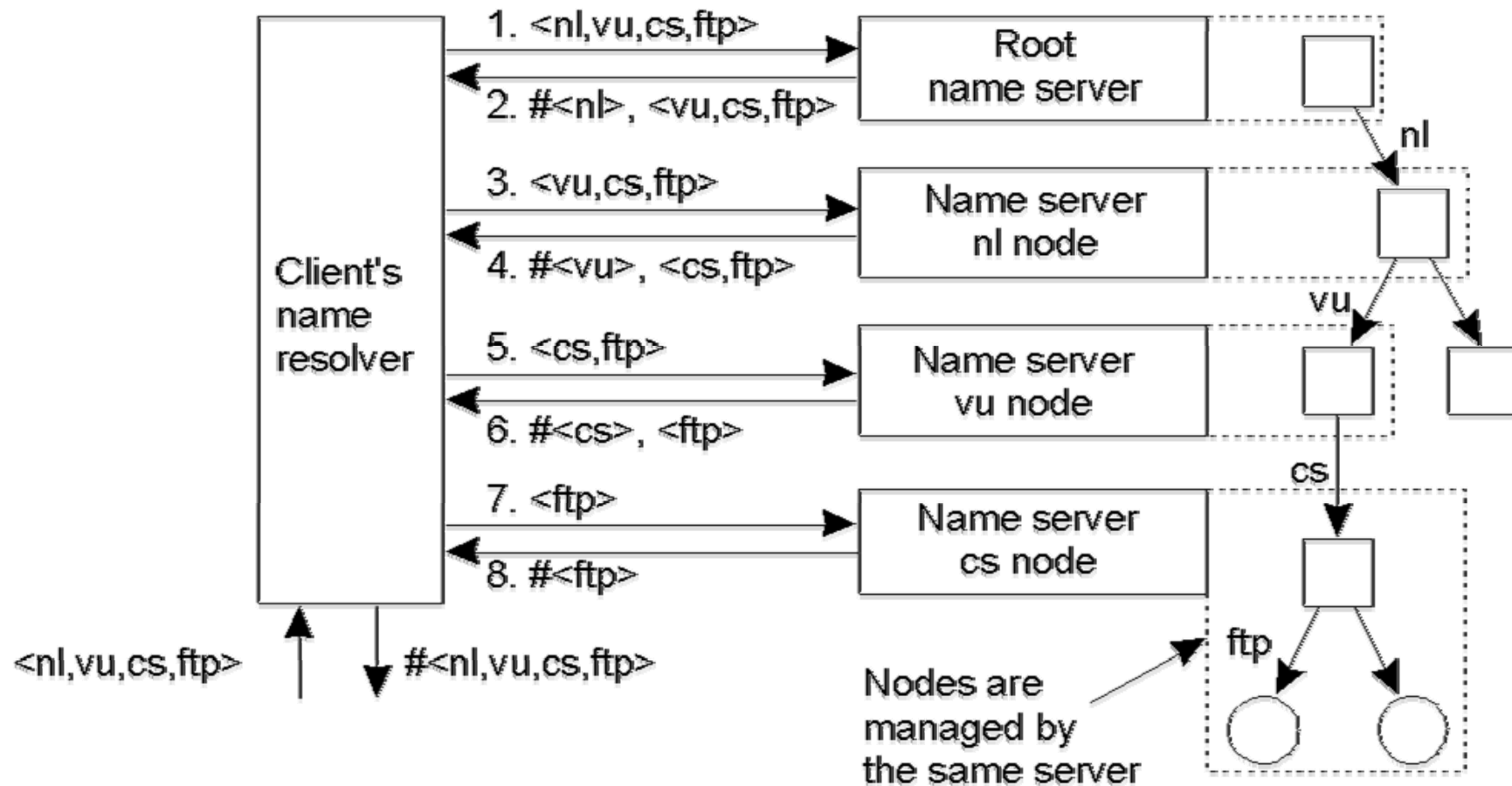


Slide based on: Instructor's Manual for Peterson and Davie:
Computer Networks (2nd ed.), © Morgan-Kaufmann Publishers 2000

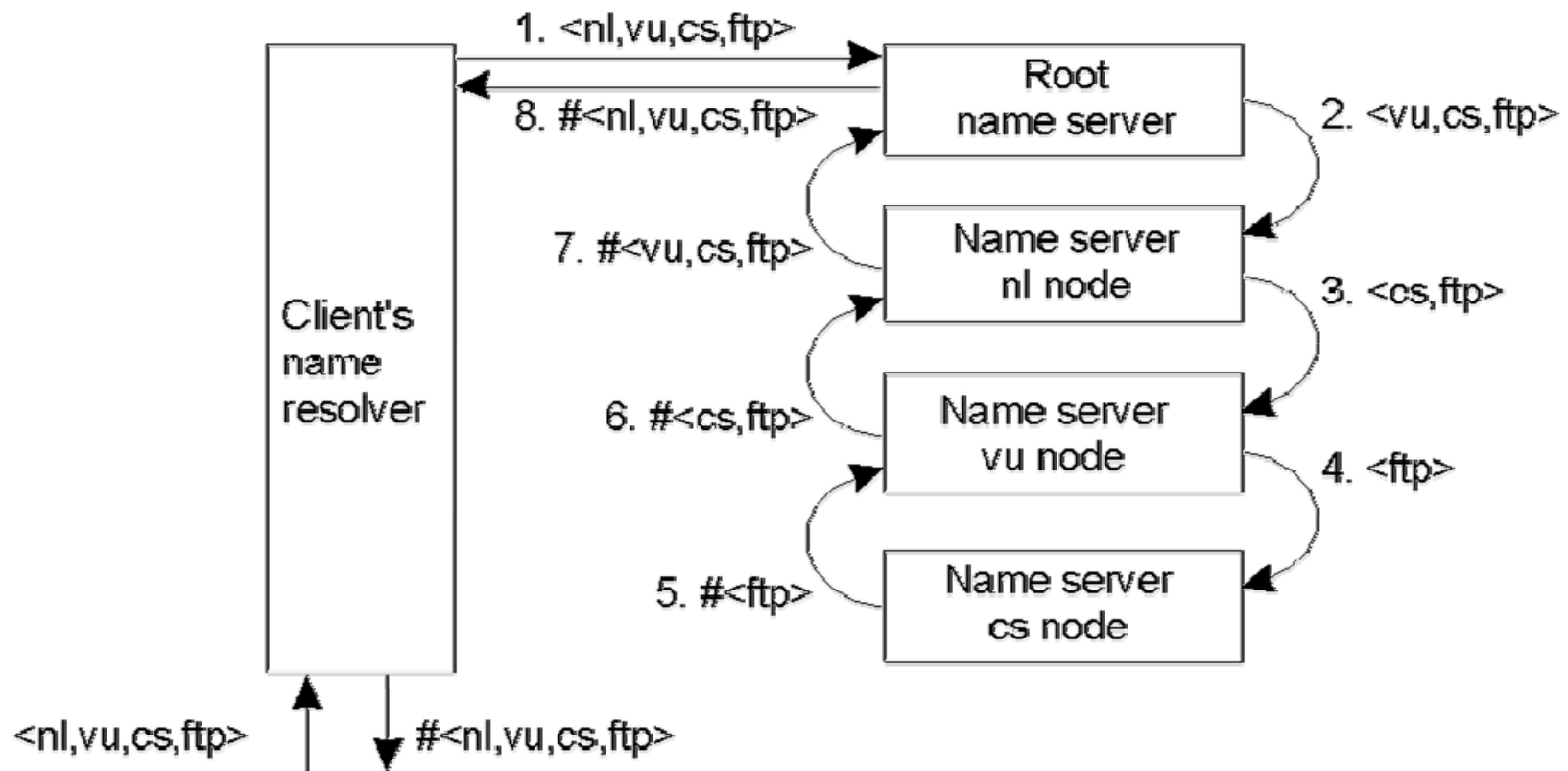
- 13 redundant entry points for DNS hierarchy
 - A – M
 - Compatibility issues
 - `dig @a.root-servers.net . soa`
 - Response with 493 bytes
 - UDP replies restricted to 512 bytes
 - Several replicated at different locations
 - <http://root-servers.org/>

- Names
- DNS
- **Name Resolution**
- Functionality and Examples
- DynDNS

Iterative Name Resolution



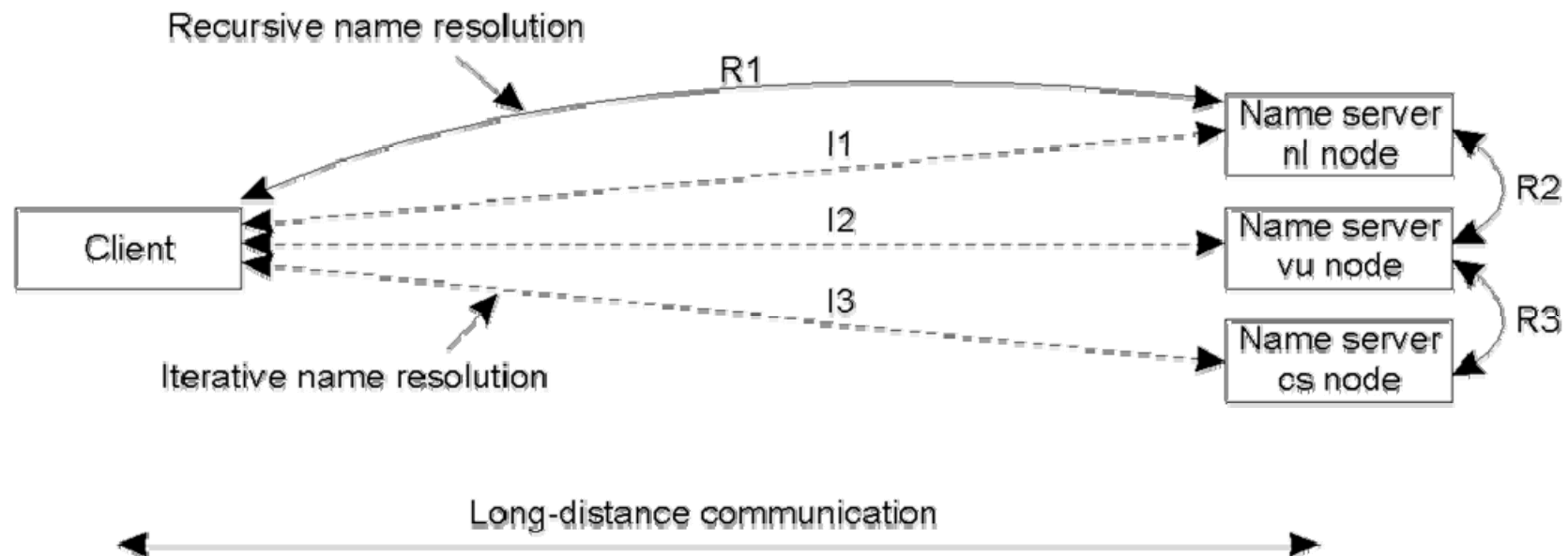
Recursive Name Resolution



Caching with Recursive Name Resolution

Server for node	Should resolve	Looks up	Passes to child	Receives and caches	Returns to requester
cs	<ftp>	#<ftp>	--	--	#<ftp>
vu	<cs,ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs, ftp>
nl	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
root	<nl,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

Name Resolution: Communication Costs



- Names
- DNS
- Name Resolution
- **Functionality and Examples**
- DynDNS

- Main functions
 - Resolution of DNS names to IP addresses
 - Identification of **mail hosts** for domain
 - Results cached for “time to live” (TTL)
- Further functions
 - Reverse resolution: Determine DNS name for IP address
 - Host information: Hardware and OS
 - Well-known services: Services offered by host
 - Optional attributes
- Additional applications (some pointers, not exhaustive)
 - Certificate directory (RFC 4398, DomainKeys Identified Mail (DKIM), Opportunistic Encryption for IPsec)
 - Sender Policy Framework (SPF)
 - Covert channel (DNS Tunneling (IP over DNS))
 - <http://www.daemon.be/maarten/dnstunnel.html>

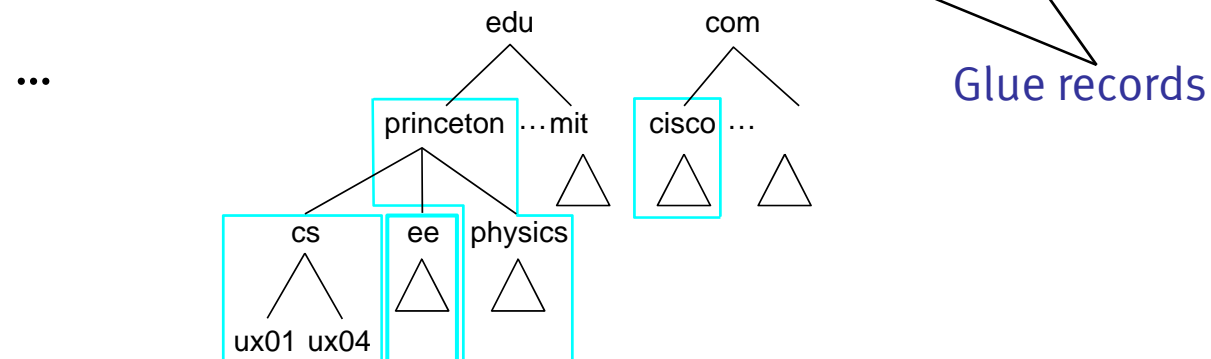
- Name servers manage **resource records**
 - (NAME, VALUE, TYPE, CLASS, TTL)
- NAME/VALUE: Not only names and IP addresses
- Sample TYPEs
 - **A**: DNS name and IPv4 address (IPv6 is AAAA)
 - **NS**: VALUE is DNS name for name server that is able to resolve names of domain NAME
 - **CNAME**: VALUE is canonical name; used for aliasing
 - **MX**: VALUE is DNS name for mail server of domain NAME
- **CLASS**: Constantly “IN” in Internet (omitted in following)
- **TTL**: Time to live of resource record

(princeton.edu, cit.princeton.edu, NS)

(cit.princeton.edu, 128.196.128.233, A)

(cisco.com, thumper.cisco.com, NS)

(thumper.ciscoe.com, 128.96.32.20, A)



- (cs.princeton.edu, optima.cs.princeton.edu, NS)
- (optima.cs.princeton.edu, 192.12.69.5, A)

- (ee.princeton.edu, helios.ee.princeton.edu, NS)
- (helios.ee.princeton.edu, 128.196.28.166, A)

- (jupiter.physics.princeton.edu, 128.196.4.1, A)
- (saturn.physics.princeton.edu, 128.196.4.2, A)
- (mars.physics.princeton.edu, 128.196.4.3, A)
- (venus.physics.princeton.edu, 128.196.4.4, A)

- (cs.princeton.edu, optima.cs.princeton.edu, MX)
- (optima.cs.princeton.edu, 192.12.69.5, A)
- (opt.cs.princeton.edu, optima.cs.princeton.edu, CNAME)

- (cheltenham.cs.princeton.edu, 192.12.69.60, A)
- (che.cs.princeton.edu, cheltenham.cs.princeton.edu, CNAME)

- (baskerville.cs.princeton.edu, 192.12.69.35, A)
- (bas.cs.princeton.edu, baskerville.cs.princeton.edu, CNAME)

- Suppose the country of Kinakuta decides that `xlx.uni-muenster.de` hosts subversive content
- They consider DNS blocking to restrict access
- Discussion
 - What is “DNS Blocking”?
 - Where and how to block what?
 - Efficacy?
 - Side effects?
 - Alternatives?
 - What if we replace “Kinakuta” with “Germany” and “subversive” with “illegal”?



- Names
- DNS
- Name Resolution
- Functionality and Examples
- **DynDNS**

- Aim

- Permanent DNS name for **dynamic** IP address

- Idea

- Special DNS servers allow **updates in real-time**
- TTL of 60s
- Special client software to notify DNS server of changed IP address
 - Sometimes integrated into DSL router/modem

- DNS as prime example for
 - Name service
 - Distributed system
 - Global scalability via distribution, partitioning, caching
- Binding of names and addresses (and further attributes)
 - Names resolved upon request
 - Iterative vs recursive resolution
- Bindings are static
 - DynDNS for dynamic IP addresses
- Security issues

- Explain URIs, URLs, URNs
- Explain assumptions and advantages of caching (ongoing)
- Discuss (dis-) advantages of iterative and recursive DNS
- Discuss effects of regional DNS blocking on Web page accessibility
- Perform DNS lookups for different resource types via nslookup
- Use Wireshark to monitor network traffic