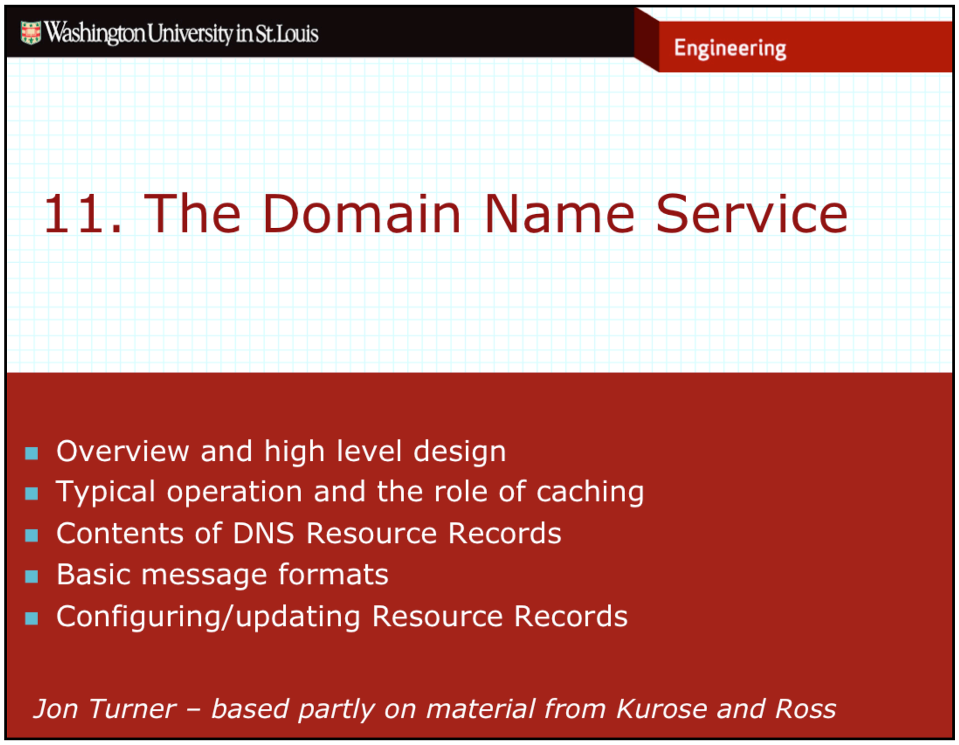


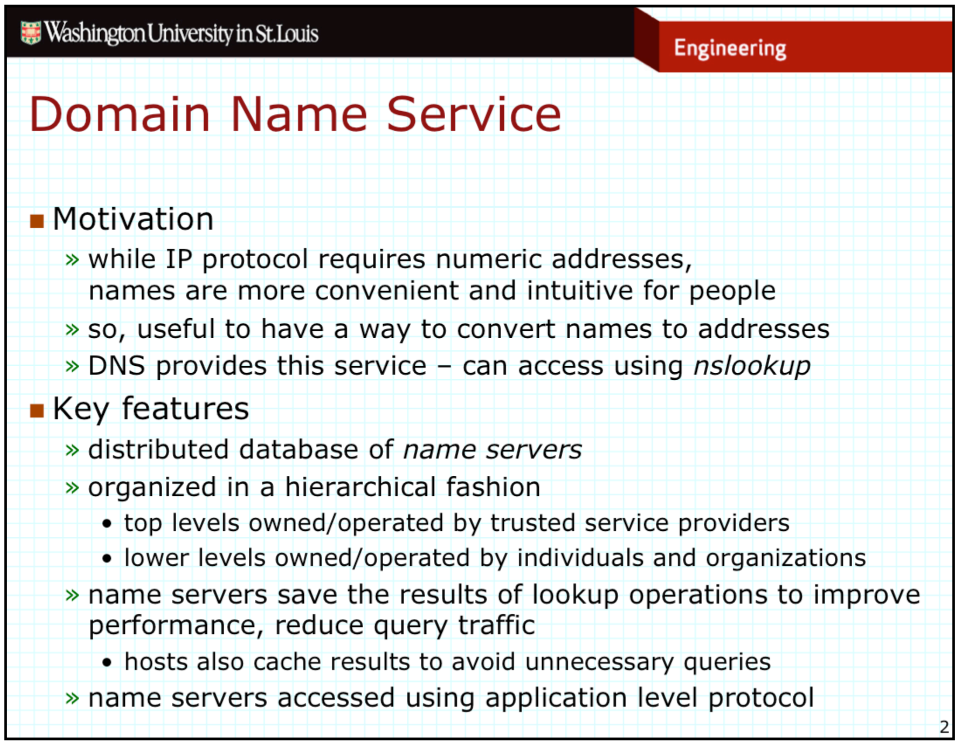
Washington University in St.Louis

Engineering

11. The Domain Name Service

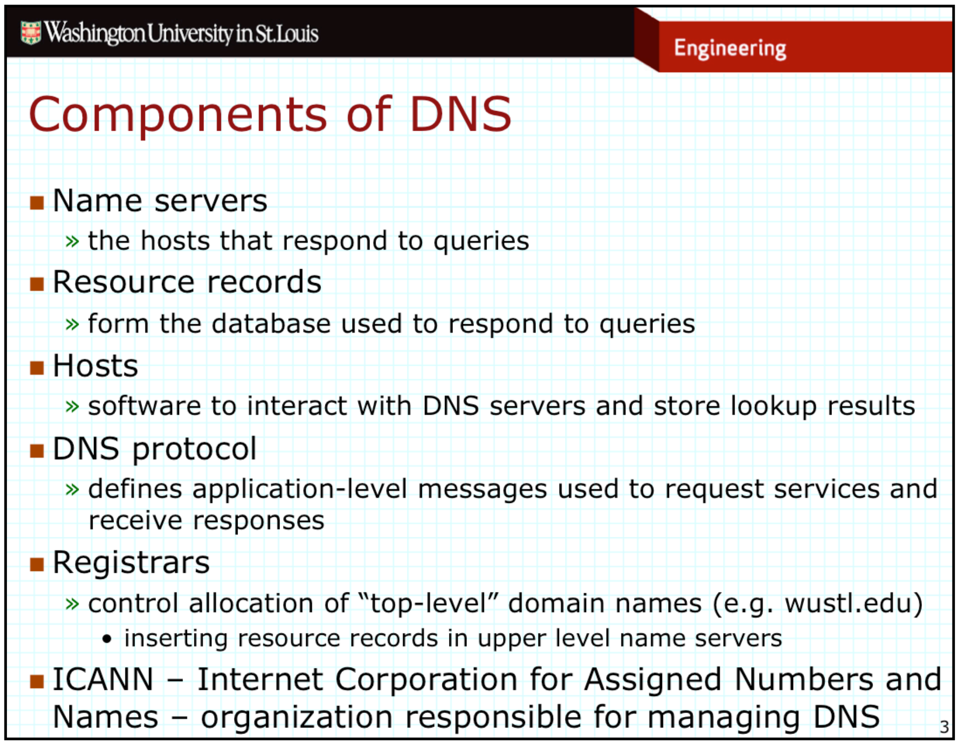
- Overview and high level design
- Typical operation and the role of caching
- Contents of DNS Resource Records
- Basic message formats
- Configuring/updating Resource Records

Jon Turner – based partly on material from Kurose and Ross



Domain Name Service

- Motivation
  - » while IP protocol requires numeric addresses, names are more convenient and intuitive for people
  - » so, useful to have a way to convert names to addresses
  - » DNS provides this service – can access using *nslookup*
- Key features
  - » distributed database of *name servers*
  - » organized in a hierarchical fashion
    - top levels owned/operated by trusted service providers
    - lower levels owned/operated by individuals and organizations
  - » name servers save the results of lookup operations to improve performance, reduce query traffic
    - hosts also cache results to avoid unnecessary queries
  - » name servers accessed using application level protocol



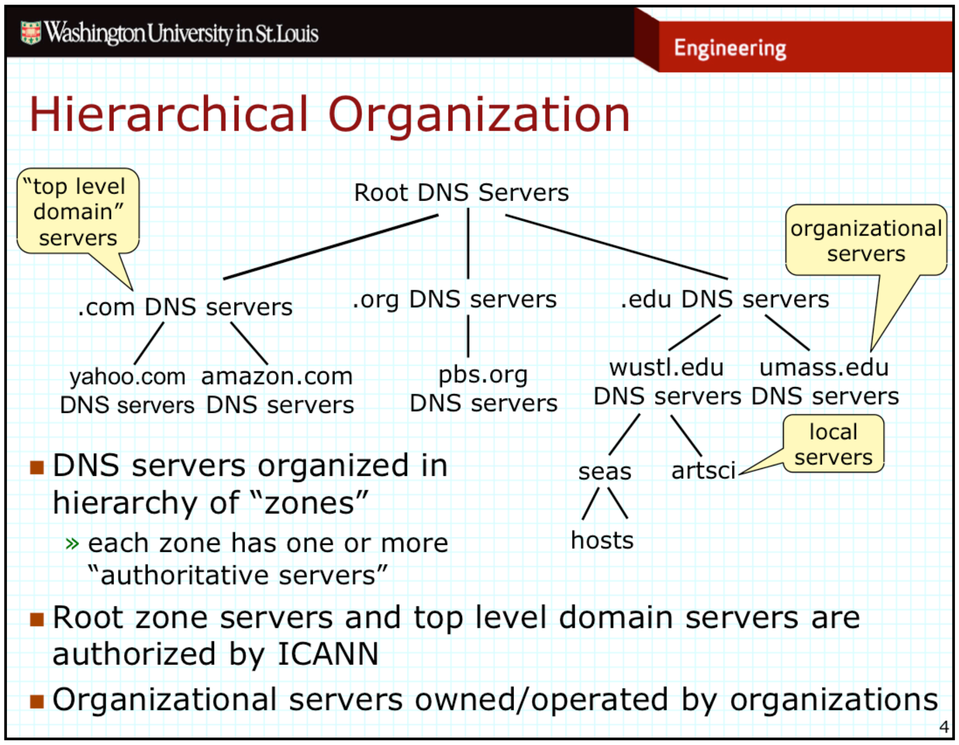
Washington University in St.Louis

Engineering

Components of DNS

- Name servers
  - » the hosts that respond to queries
- Resource records
  - » form the database used to respond to queries
- Hosts
  - » software to interact with DNS servers and store lookup results
- DNS protocol
  - » defines application-level messages used to request services and receive responses
- Registrars
  - » control allocation of "top-level" domain names (e.g. wustl.edu)
    - inserting resource records in upper level name servers
- ICANN – Internet Corporation for Assigned Numbers and Names – organization responsible for managing DNS

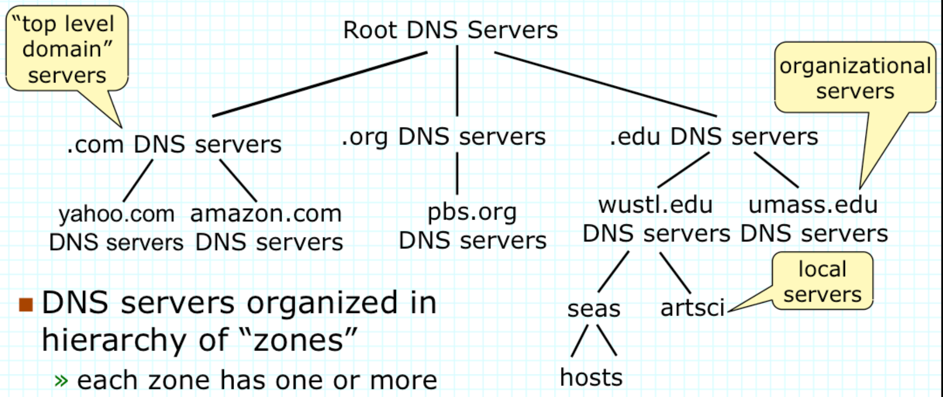
3



Washington University in St.Louis

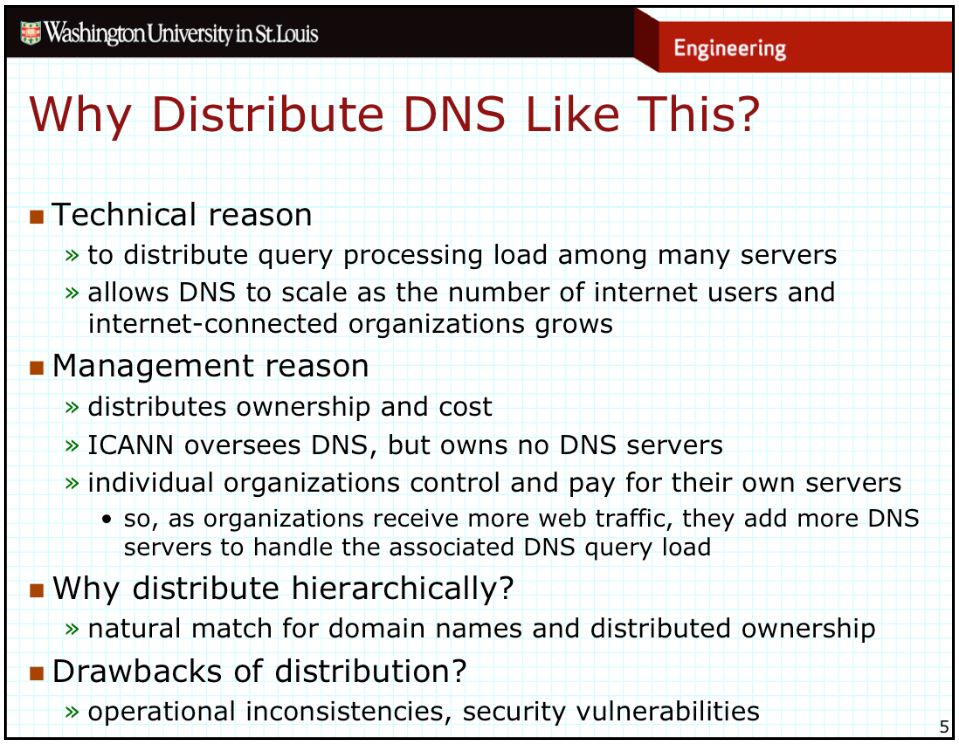
Engineering

Hierarchical Organization



- DNS servers organized in hierarchy of "zones"
  - » each zone has one or more "authoritative servers"
- Root zone servers and top level domain servers are authorized by ICANN
- Organizational servers owned/operated by organizations

4



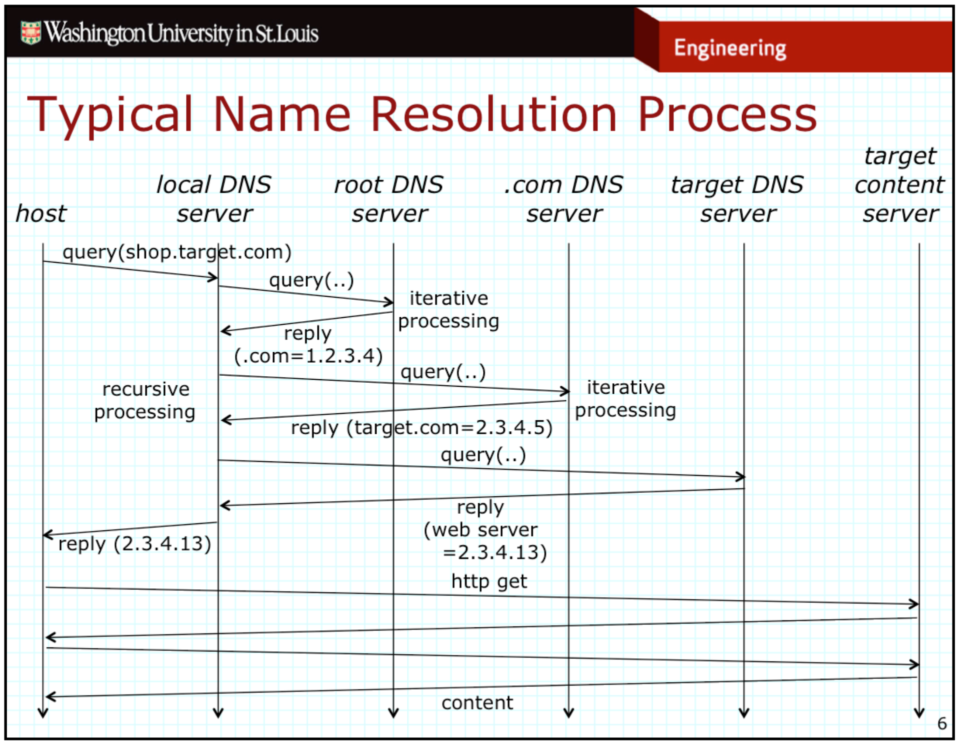
Washington University in St.Louis

Engineering

Why Distribute DNS Like This?

- Technical reason
  - » to distribute query processing load among many servers
  - » allows DNS to scale as the number of internet users and internet-connected organizations grows
- Management reason
  - » distributes ownership and cost
  - » ICANN oversees DNS, but owns no DNS servers
  - » individual organizations control and pay for their own servers
    - so, as organizations receive more web traffic, they add more DNS servers to handle the associated DNS query load
- Why distribute hierarchically?
  - » natural match for domain names and distributed ownership
- Drawbacks of distribution?
  - » operational inconsistencies, security vulnerabilities

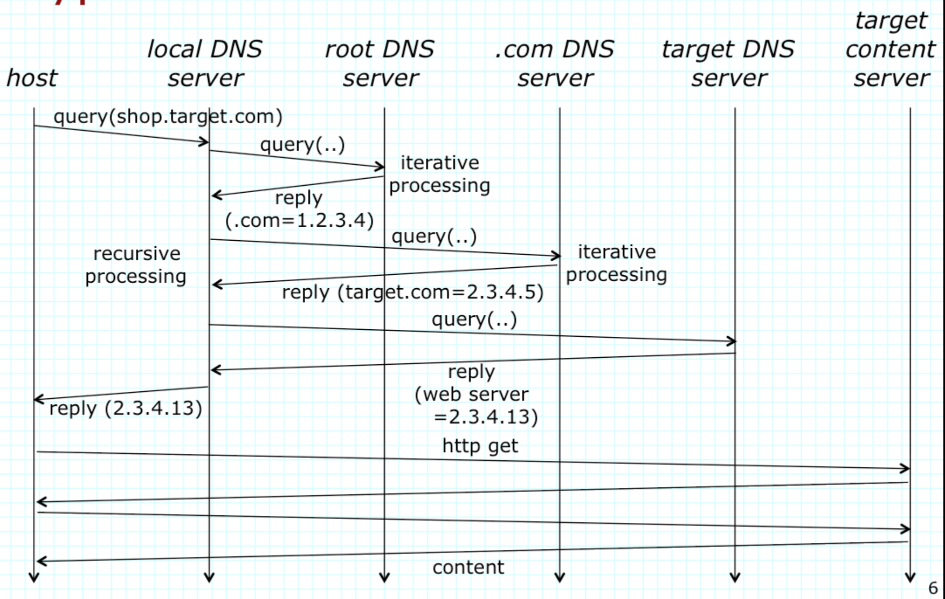
5



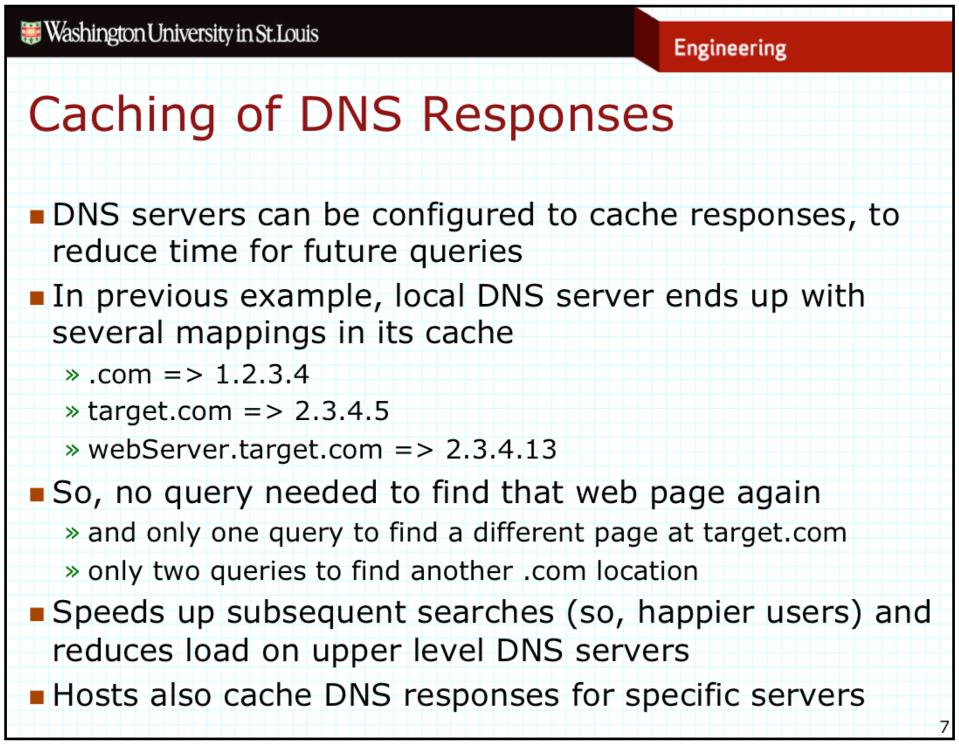
Washington University in St. Louis

Engineering

Typical Name Resolution Process



6



Washington University in St.Louis

Engineering

Caching of DNS Responses

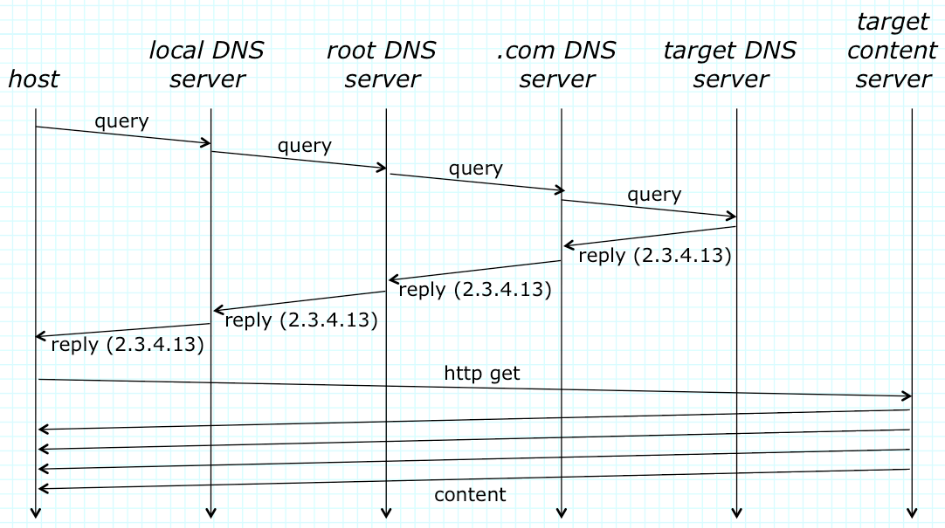
- DNS servers can be configured to cache responses, to reduce time for future queries
- In previous example, local DNS server ends up with several mappings in its cache
  - » .com => 1.2.3.4
  - » target.com => 2.3.4.5
  - » webServer.target.com => 2.3.4.13
- So, no query needed to find that web page again
  - » and only one query to find a different page at target.com
  - » only two queries to find another .com location
- Speeds up subsequent searches (so, happier users) and reduces load on upper level DNS servers
- Hosts also cache DNS responses for specific servers

7

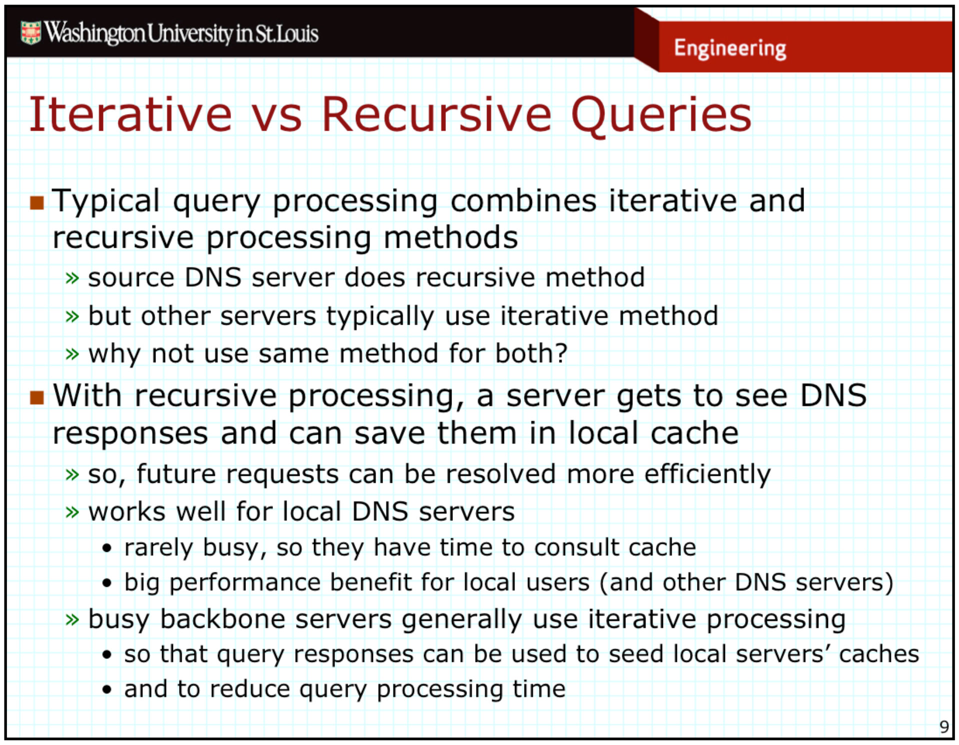
Washington University in St.Louis

Engineering

Pure Recursive Resolution Process



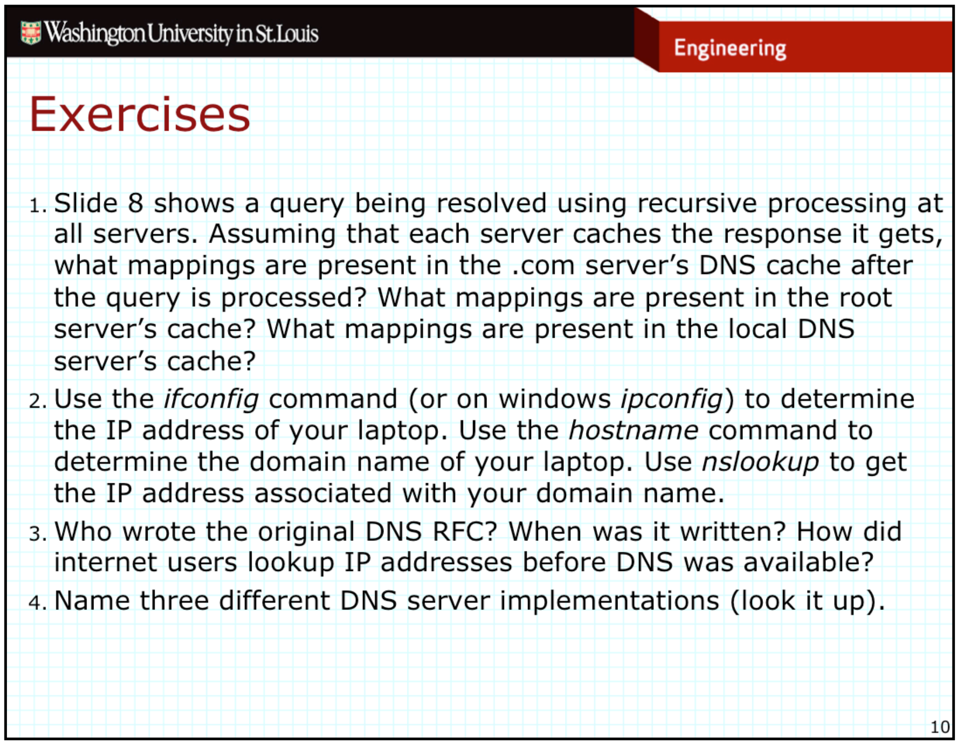
8



Iterative vs Recursive Queries

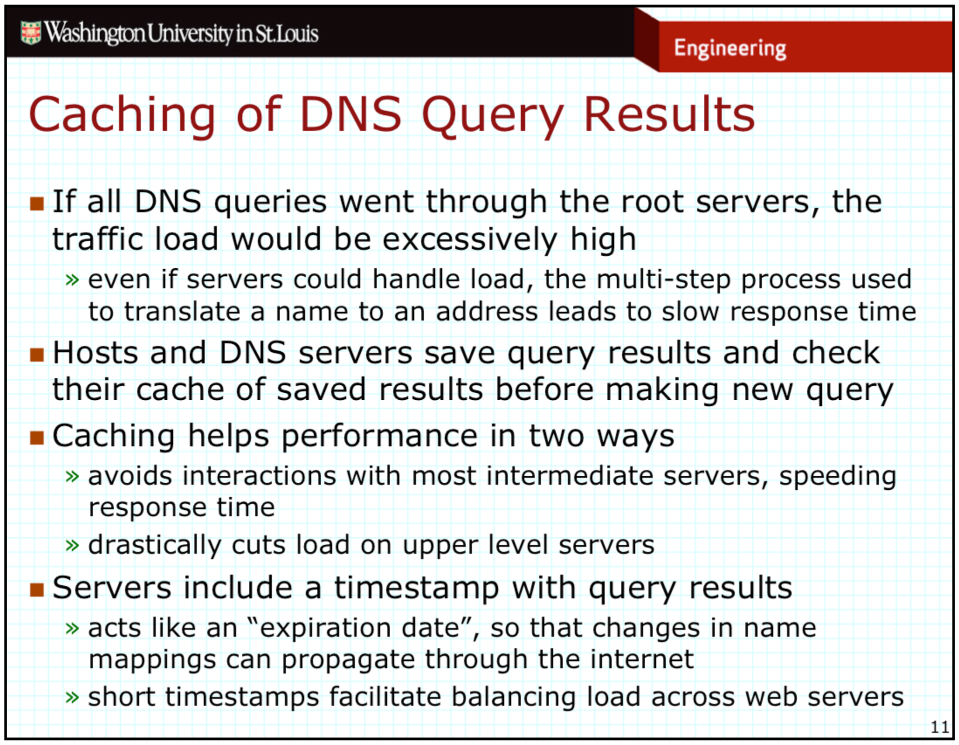
- Typical query processing combines iterative and recursive processing methods
  - » source DNS server does recursive method
  - » but other servers typically use iterative method
  - » why not use same method for both?
- With recursive processing, a server gets to see DNS responses and can save them in local cache
  - » so, future requests can be resolved more efficiently
  - » works well for local DNS servers
    - rarely busy, so they have time to consult cache
    - big performance benefit for local users (and other DNS servers)
  - » busy backbone servers generally use iterative processing
    - so that query responses can be used to seed local servers’ caches
    - and to reduce query processing time

9



Exercises

1. Slide 8 shows a query being resolved using recursive processing at all servers. Assuming that each server caches the response it gets, what mappings are present in the .com server's DNS cache after the query is processed? What mappings are present in the root server's cache? What mappings are present in the local DNS server's cache?
2. Use the *ifconfig* command (or on windows *ipconfig*) to determine the IP address of your laptop. Use the *hostname* command to determine the domain name of your laptop. Use *nslookup* to get the IP address associated with your domain name.
3. Who wrote the original DNS RFC? When was it written? How did internet users lookup IP addresses before DNS was available?
4. Name three different DNS server implementations (look it up).

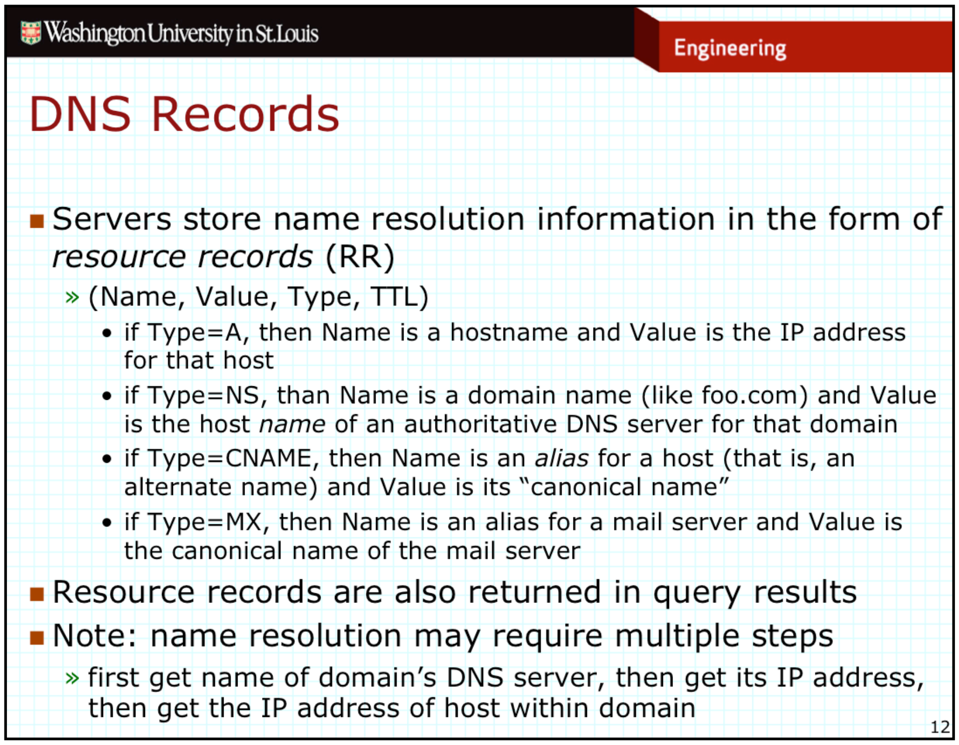

Washington University in St.Louis

Engineering

Caching of DNS Query Results

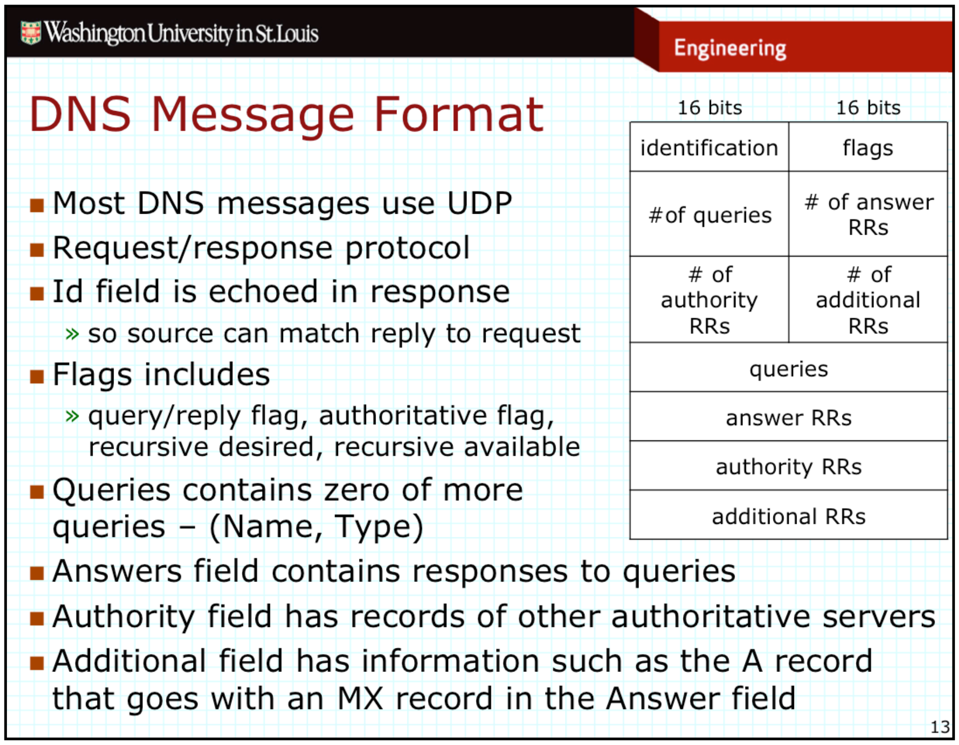
- If all DNS queries went through the root servers, the traffic load would be excessively high
  - » even if servers could handle load, the multi-step process used to translate a name to an address leads to slow response time
- Hosts and DNS servers save query results and check their cache of saved results before making new query
- Caching helps performance in two ways
  - » avoids interactions with most intermediate servers, speeding response time
  - » drastically cuts load on upper level servers
- Servers include a timestamp with query results
  - » acts like an "expiration date", so that changes in name mappings can propagate through the internet
  - » short timestamps facilitate balancing load across web servers

11



DNS Records

- Servers store name resolution information in the form of *resource records* (RR)
  - » (Name, Value, Type, TTL)
    - if Type=A, then Name is a hostname and Value is the IP address for that host
    - if Type=NS, than Name is a domain name (like foo.com) and Value is the host *name* of an authoritative DNS server for that domain
    - if Type=CNAME, then Name is an *alias* for a host (that is, an alternate name) and Value is its "canonical name"
    - if Type=MX, then Name is an alias for a mail server and Value is the canonical name of the mail server
- Resource records are also returned in query results
- Note: name resolution may require multiple steps
  - » first get name of domain's DNS server, then get its IP address, then get the IP address of host within domain



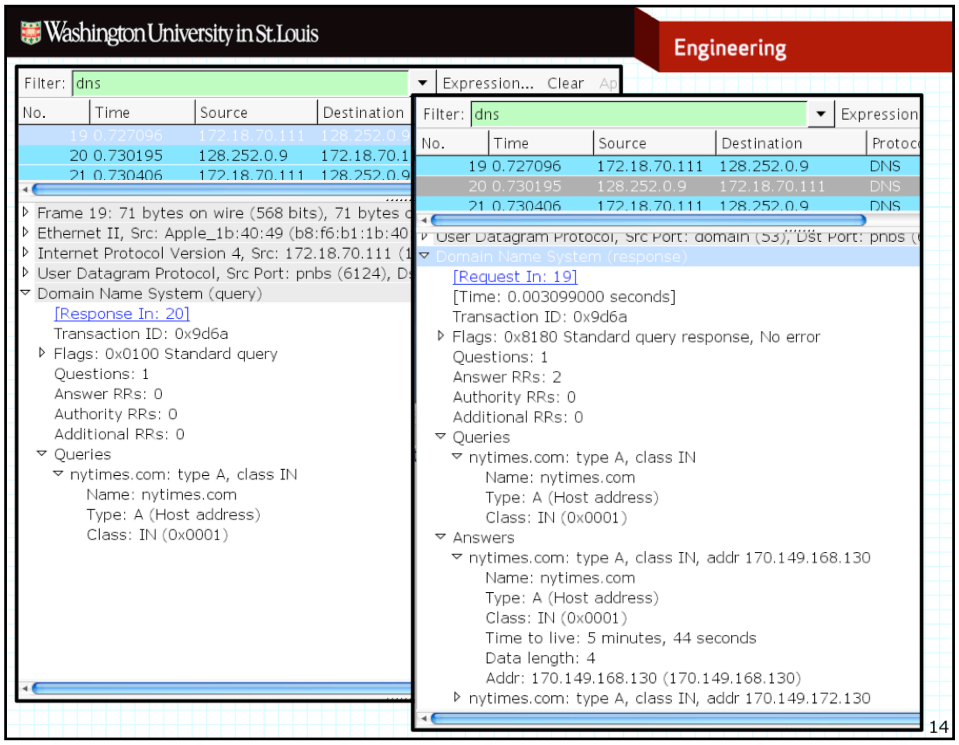
Washington University in St.Louis — Engineering

DNS Message Format

16 bits	16 bits
identification	flags
#of queries	# of answer RRs
# of authority RRs	# of additional RRs
queries	
answer RRs	
authority RRs	
additional RRs	

- Most DNS messages use UDP
- Request/response protocol
- Id field is echoed in response
  - » so source can match reply to request
- Flags includes
  - » query/reply flag, authoritative flag, recursive desired, recursive available
- Queries contains zero of more queries – (Name, Type)
- Answers field contains responses to queries
- Authority field has records of other authoritative servers
- Additional field has information such as the A record that goes with an MX record in the Answer field

13



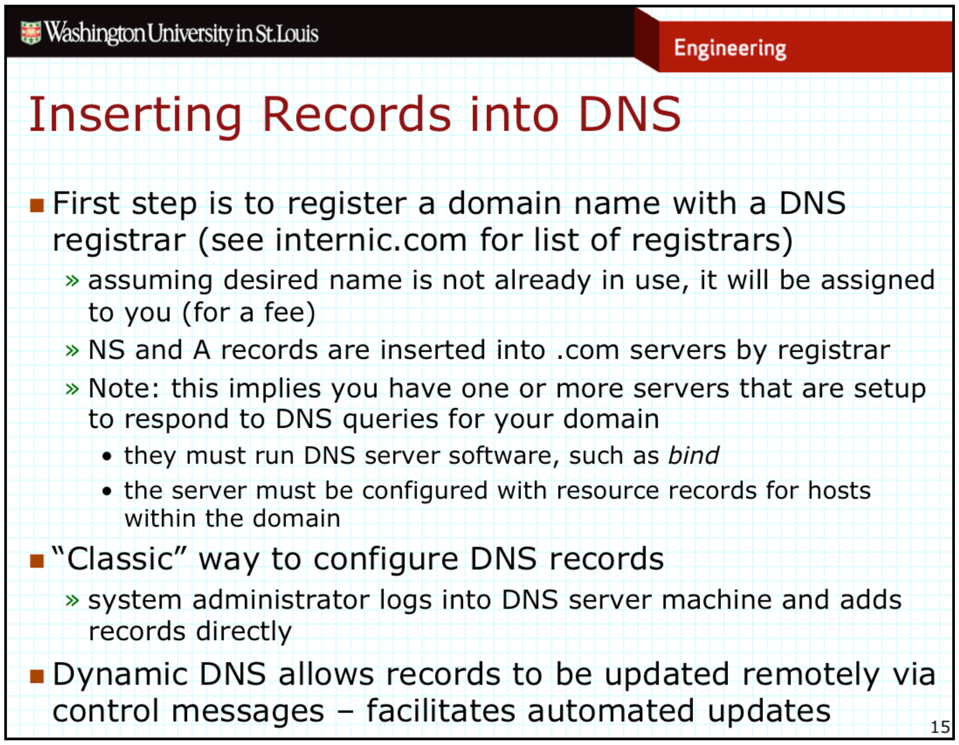
```
Filter: dns                                  Expression...  Clear  Ap
No.      Time        Source          Destination
   19 0.727096    172.18.70.111   128.252.0.9
   20 0.730195    128.252.0.9     172.18.70.1
   21 0.730406    172.18.70.111   128.252.0.9

▷ Frame 19: 71 bytes on wire (568 bits), 71 bytes c
▷ Ethernet II, Src: Apple_1b:40:49 (b8:f6:b1:1b:40
▷ Internet Protocol Version 4, Src: 172.18.70.111 (1
▷ User Datagram Protocol, Src Port: pnbs (6124), D
▽ Domain Name System (query)
    [Response In: 20]
    Transaction ID: 0x9d6a
  ▷ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ nytimes.com: type A, class IN
        Name: nytimes.com
        Type: A (Host address)
        Class: IN (0x0001)
```

```
Filter: dns                                  Expression
No.      Time        Source          Destination     Protoc
   19 0.727096    172.18.70.111   128.252.0.9      DNS
   20 0.730195    128.252.0.9     172.18.70.111    DNS
   21 0.730406    172.18.70.111   128.252.0.9      DNS

▷ User Datagram Protocol, Src Port: domain (53), Dst Port: pnbs (
▽ Domain Name System (response)
    [Request In: 19]
    [Time: 0.003099000 seconds]
    Transaction ID: 0x9d6a
  ▷ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ nytimes.com: type A, class IN
        Name: nytimes.com
        Type: A (Host address)
        Class: IN (0x0001)
  ▽ Answers
    ▽ nytimes.com: type A, class IN, addr 170.149.168.130
        Name: nytimes.com
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 5 minutes, 44 seconds
        Data length: 4
        Addr: 170.149.168.130 (170.149.168.130)
    ▷ nytimes.com: type A, class IN, addr 170.149.172.130
```

14

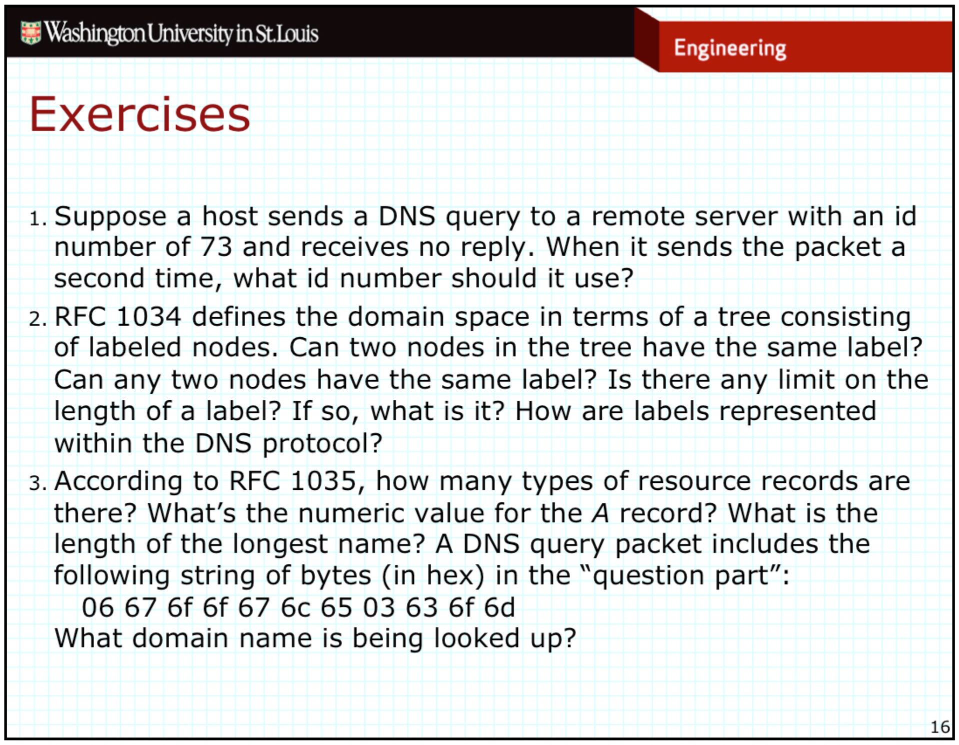


Washington University in St. Louis | Engineering

Inserting Records into DNS

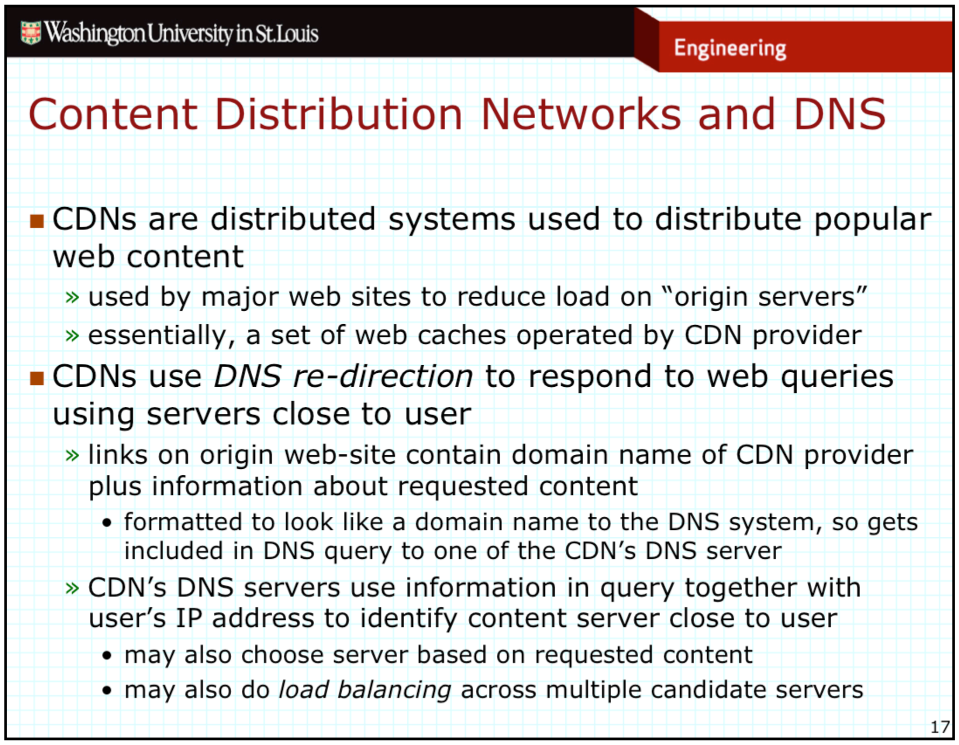
- First step is to register a domain name with a DNS registrar (see internic.com for list of registrars)
  - » assuming desired name is not already in use, it will be assigned to you (for a fee)
  - » NS and A records are inserted into .com servers by registrar
  - » Note: this implies you have one or more servers that are setup to respond to DNS queries for your domain
    - they must run DNS server software, such as *bind*
    - the server must be configured with resource records for hosts within the domain
- "Classic" way to configure DNS records
  - » system administrator logs into DNS server machine and adds records directly
- Dynamic DNS allows records to be updated remotely via control messages – facilitates automated updates

15



Exercises

1. Suppose a host sends a DNS query to a remote server with an id number of 73 and receives no reply. When it sends the packet a second time, what id number should it use?
2. RFC 1034 defines the domain space in terms of a tree consisting of labeled nodes. Can two nodes in the tree have the same label? Can any two nodes have the same label? Is there any limit on the length of a label? If so, what is it? How are labels represented within the DNS protocol?
3. According to RFC 1035, how many types of resource records are there? What's the numeric value for the *A* record? What is the length of the longest name? A DNS query packet includes the following string of bytes (in hex) in the "question part":
   06 67 6f 6f 67 6c 65 03 63 6f 6d
   What domain name is being looked up?



Washington University in St. Louis

Engineering

Content Distribution Networks and DNS

- CDNs are distributed systems used to distribute popular web content
  - » used by major web sites to reduce load on "origin servers"
  - » essentially, a set of web caches operated by CDN provider
- CDNs use *DNS re-direction* to respond to web queries using servers close to user
  - » links on origin web-site contain domain name of CDN provider plus information about requested content
    - formatted to look like a domain name to the DNS system, so gets included in DNS query to one of the CDN's DNS server
  - » CDN's DNS servers use information in query together with user's IP address to identify content server close to user
    - may also choose server based on requested content
    - may also do *load balancing* across multiple candidate servers

17