Unit 5: Applications and NATs

HyperText

History [edit source | edit beta]

The term HyperText was coined by Ted Nelson who in turn was inspired by Vannevar Bush's microfilm-based "memex". Tim Berners-Lee first proposed the "WorldWideWeb" project — now known as the World Wide Web. Berners-Lee and his team are credited with inventing the original HTTP along with HTML and the associated technology for a web server and a text-based web browser. The first version of the protocol had only one method, namely GET, which would request a page from a server.[3] The response from the server was always an HTML page.[4]

Tim Berners-Lee

The first documented version of HTTP was **HTTP V0.9** (1991). Dave Raggett led the HTTP Working Group (HTTP WG) in 1995 and wanted to expand the protocol with extended operations, extended negotiation, richer meta-information, tied with a security protocol which became more efficient by adding additional methods and header fields.[5][6] RFC 1945 officially introduced and recognized HTTP V1.0 in 1996.

The HTTP WG planned to publish new standards in December 1995[7] and the support for pre-standard HTTP/1.1 based on the then developing RFC 2068 (called HTTP-NG) was rapidly adopted by the major browser developers in early 1996. By March 1996, pre-standard HTTP/1.1 was supported in Arena,[8] Netscape 2.0,[8] Netscape Navigator Gold 2.01,[8] Mosaic 2.7,[citation needed] Lynx 2.5[citation needed], and in Internet Explorer 2.0[citation needed]. End-user adoption of the new browsers was rapid. In March 1996, one web hosting company reported that over 40% of browsers in use on the Internet were HTTP 1.1 compliant.[citation needed] That same web hosting company reported that by June 1996, 65% of all browsers accessing their servers were HTTP/1.1 compliant.[9] The HTTP/1.1 standard as defined in RFC 2068 was officially released in January 1997. Improvements and updates to the HTTP/1.1 standard were released under RFC 2616 in June 1999.

```
<h2><span class="mw-headline" id="History">History</span><span class="mw-editsection"><span class="mw-editsection-bracket">[</span><a href="/w/index.p
<div class="thumb tright">
<div class="thumbinner" style="width:192px;"><a href="/wiki/File:Tim_Berners-Lee_CP_2.jpg" class="image"><img alt="" src="//upload.wikimedia.org/wikip
<div class="thumbcaption">
<div class="magnify"><a href="/wiki/File:Tim_Berners-Lee_CP_2.jpg" class="internal" title="Enlarge"><img src="//bits.wikimedia.org/static-1.22wmf15/sk
<a href="/wiki/Tim_Berners-Lee" title="Tim Berners-Lee">Tim Berners-Lee</a></div>
</div>
</div>
<p>The term <a href="/wiki/HyperText" title="HyperText" class="mw-redirect">HyperText</a> was coined by <a href="/wiki/Ted_Nelson" title="Ted Nelson">
<p>The first documented version of HTTP was <b><a rel="nofollow" class="external text" href="http://www.w3.org/pub/WWW/Protocols/HTTP/AsImplemented.ht
<p>The HTTP WG planned to publish new standards in December 1995<sup id="cite_ref-7" class="reference"><a href="#cite_note-7"><span>[</span>7<span>]</
```

HyperText

History [edit source | edit beta]

The term HyperText was coined by Ted Nelson who in turn was inspired by Vannevar Bush's microfilm-based "memex". Tim Berners-Lee first proposed the "WorldWideWeb" project — now known as the World Wide Web. Berners-Lee and his team are credited with inventing the original HTTP along with HTML and the associated technology for a web server and a text-based web browser. The first version of the protocol had only one method, namely GET, which would request a page from a server.[3] The response from the server was always an HTML page.[4]

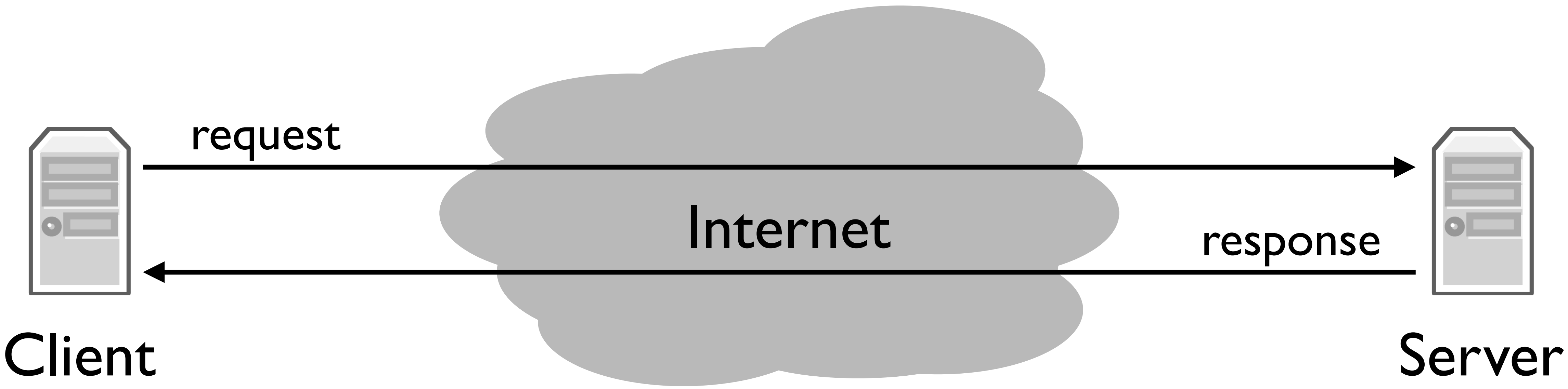
Tim Berners-Lee

The first documented version of HTTP was **HTTP V0.9** (1991). Dave Raggett led the HTTP Working Group (HTTP WG) in 1995 and wanted to expand the protocol with extended operations, extended negotiation, richer meta-information, tied with a security protocol which became more efficient by adding additional methods and header fields.[5][6] RFC 1945 officially introduced and recognized HTTP V1.0 in 1996.

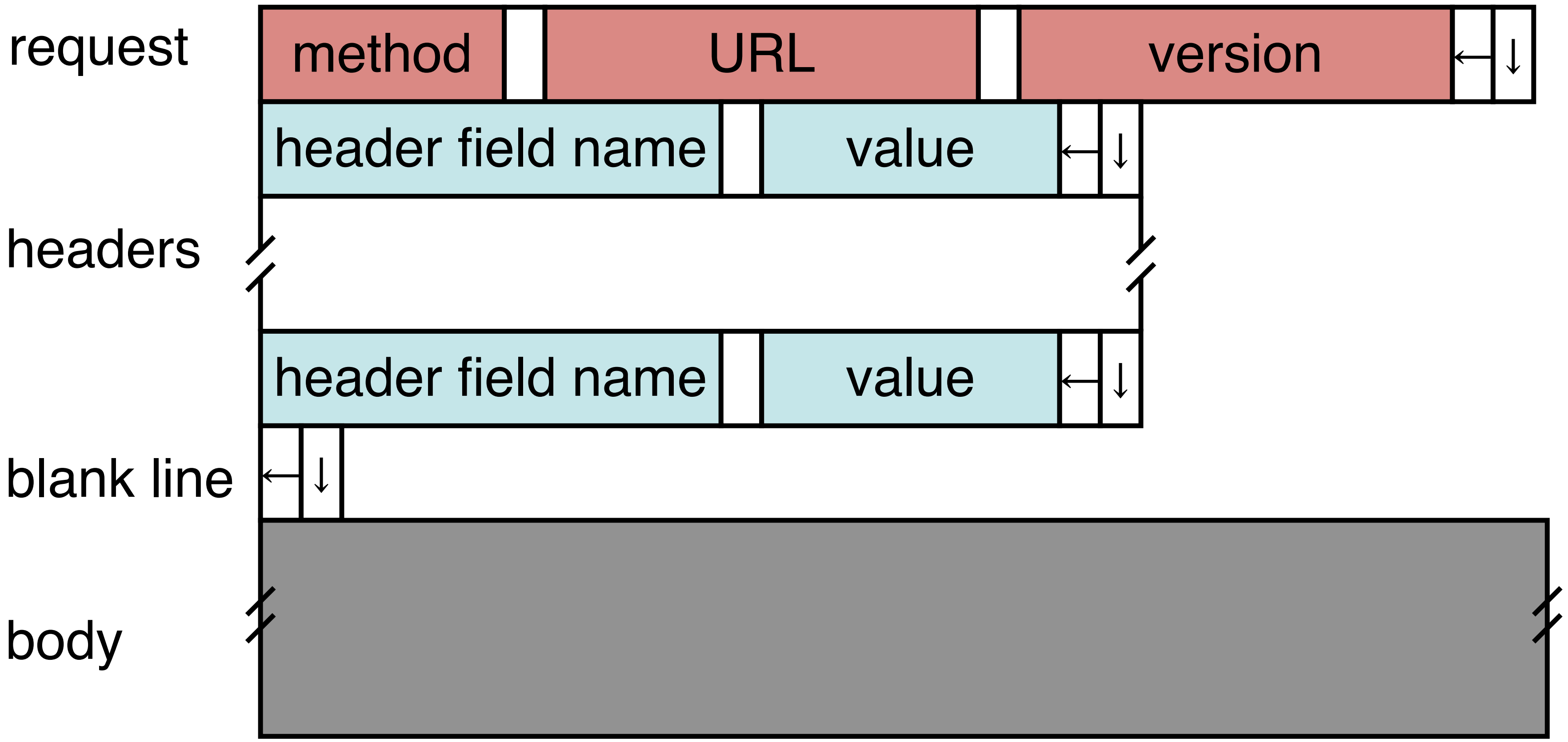
The HTTP WG planned to publish new standards in December 1995[7] and the support for pre-standard HTTP/1.1 based on the then developing RFC 2068 (called HTTP-NG) was rapidly adopted by the major browser developers in early 1996. By March 1996, pre-standard HTTP/1.1 was supported in Arena,[8] Netscape 2.0,[8] Netscape Navigator Gold 2.01,[8] Mosaic 2.7,[citation needed] Lynx 2.5[citation needed], and in Internet Explorer 2.0[citation needed]. End-user adoption of the new browsers was rapid. In March 1996, one web hosting company reported that over 40% of browsers in use on the Internet were HTTP 1.1 compliant.[citation needed] That same web hosting company reported that by June 1996, 65% of all browsers accessing their servers were HTTP/1.1 compliant.[9] The HTTP/1.1 standard as defined in RFC 2068 was officially released in January 1997. Improvements and updates to the HTTP/1.1 standard were released under RFC 2616 in June 1999.

```
<h2><span class="mw-headline" id="History">History</span><span class="mw-editsection"><span class="mw-editsection-bracket">[</span><a href="/w/index.p
<div class="thumb tright">
<div class="thumbinner" style="width:192px;"><a href="/wiki/File:Tim_Berners-Lee_CP_2.jpg" class="image"><img alt="" src="//upload.wikimedia.org/wikip
<div class="thumbcaption">
<div class="magnify"><a href="/wiki/File:Tim_Berners-Lee_CP_2.jpg" class="internal" title="Enlarge"><img src="//bits.wikimedia.org/static-1.22wmf15/sk
<a href="/wiki/Tim_Berners-Lee" title="Tim Berners-Lee">Tim Berners-Lee</a></div>
</div>
</div>
<p>The term <a href="/wiki/HyperText" title="HyperText" class="mw-redirect">HyperText</a> was coined by <a href="/wiki/Ted_Nelson" title="Ted Nelson">
<p>The first documented version of HTTP was <b><a rel="nofollow" class="external text" href="http://www.w3.org/pub/WWW/Protocols/HTTP/AsImplemented.ht
<p>The HTTP WG planned to publish new standards in December 1995<sup id="cite_ref-7" class="reference"><a href="#cite_note-7"><span>[</span>7<span>]</
```

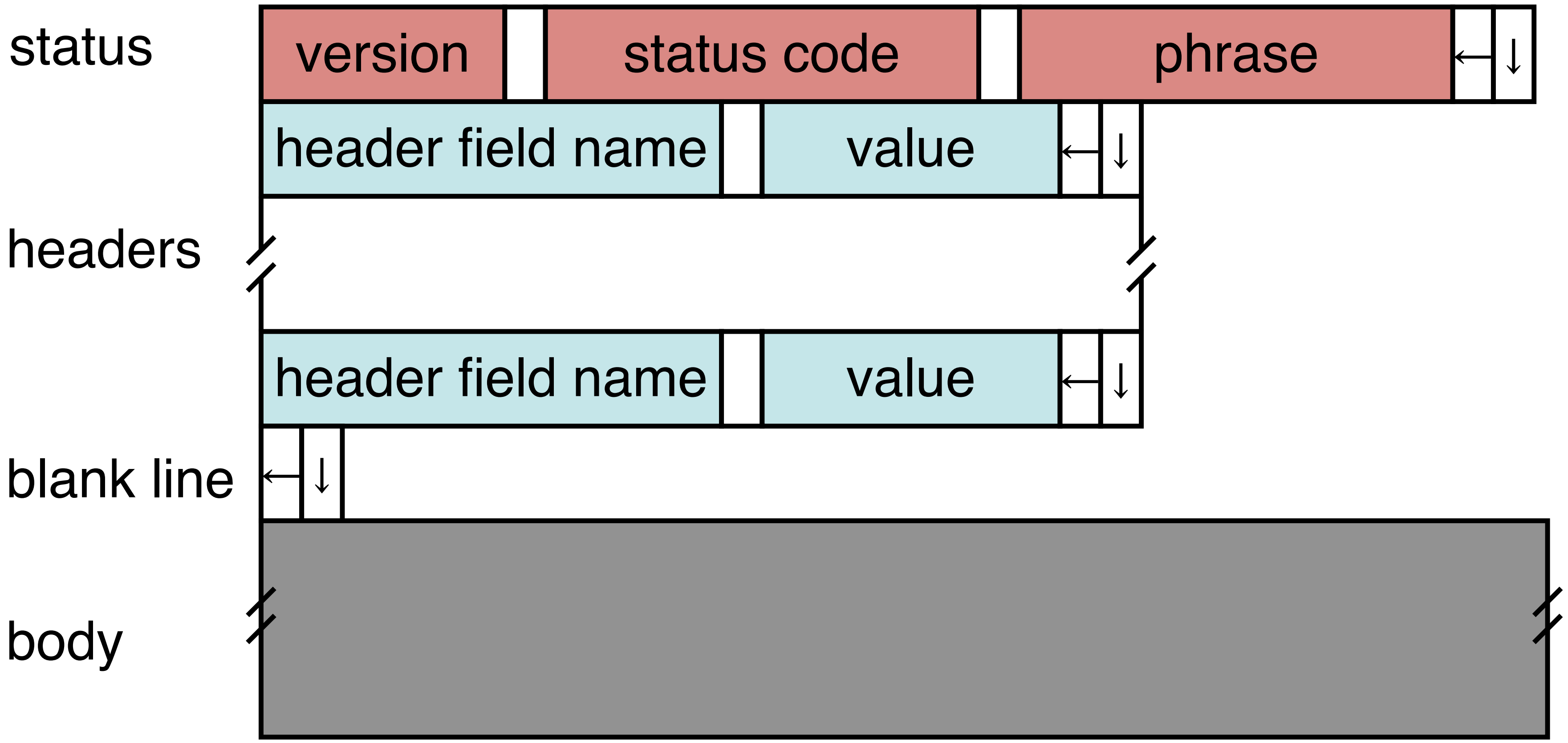
World Wide Web (HTTP)



HTTP Request Format

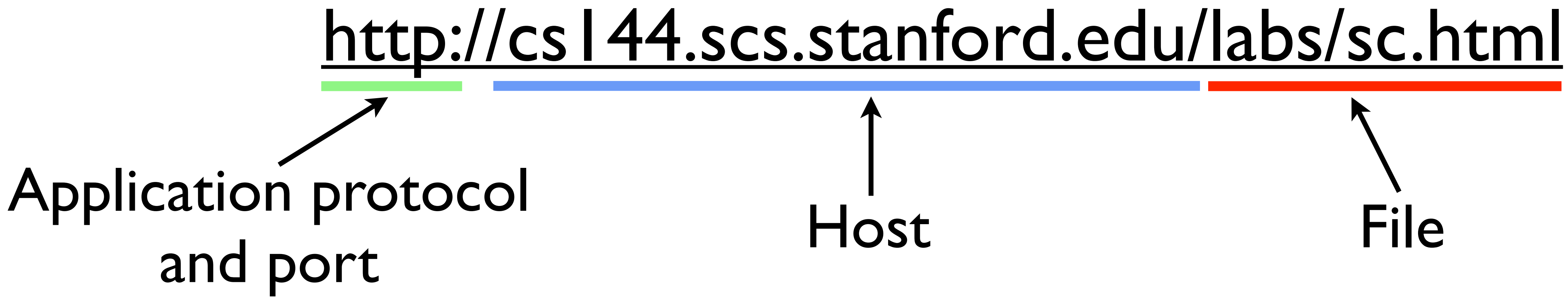


HTTP Response

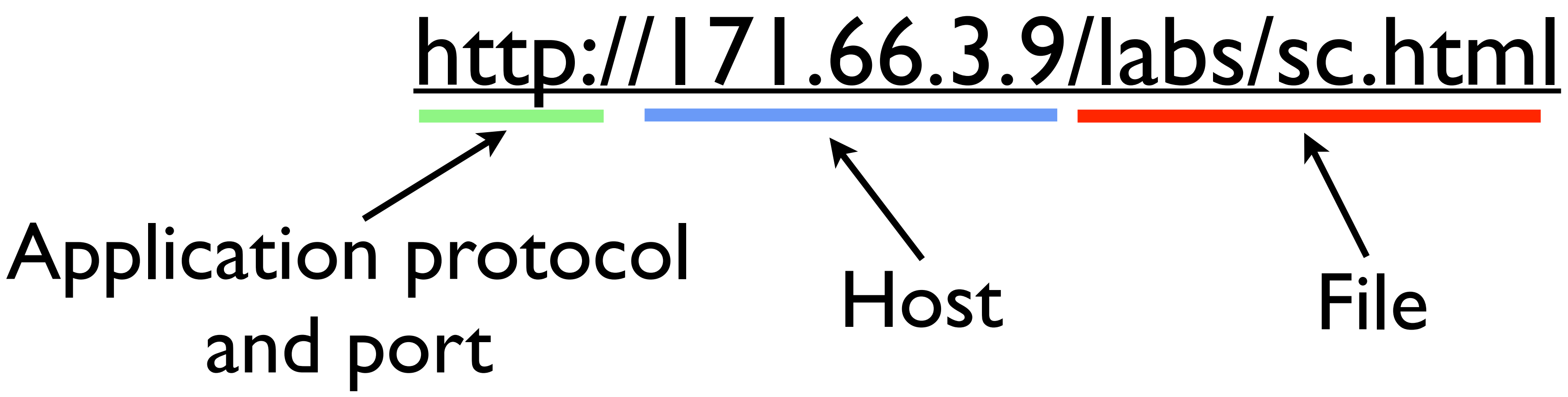


Domain Name System (DNS)

Parsing a URL



Parsing a URL



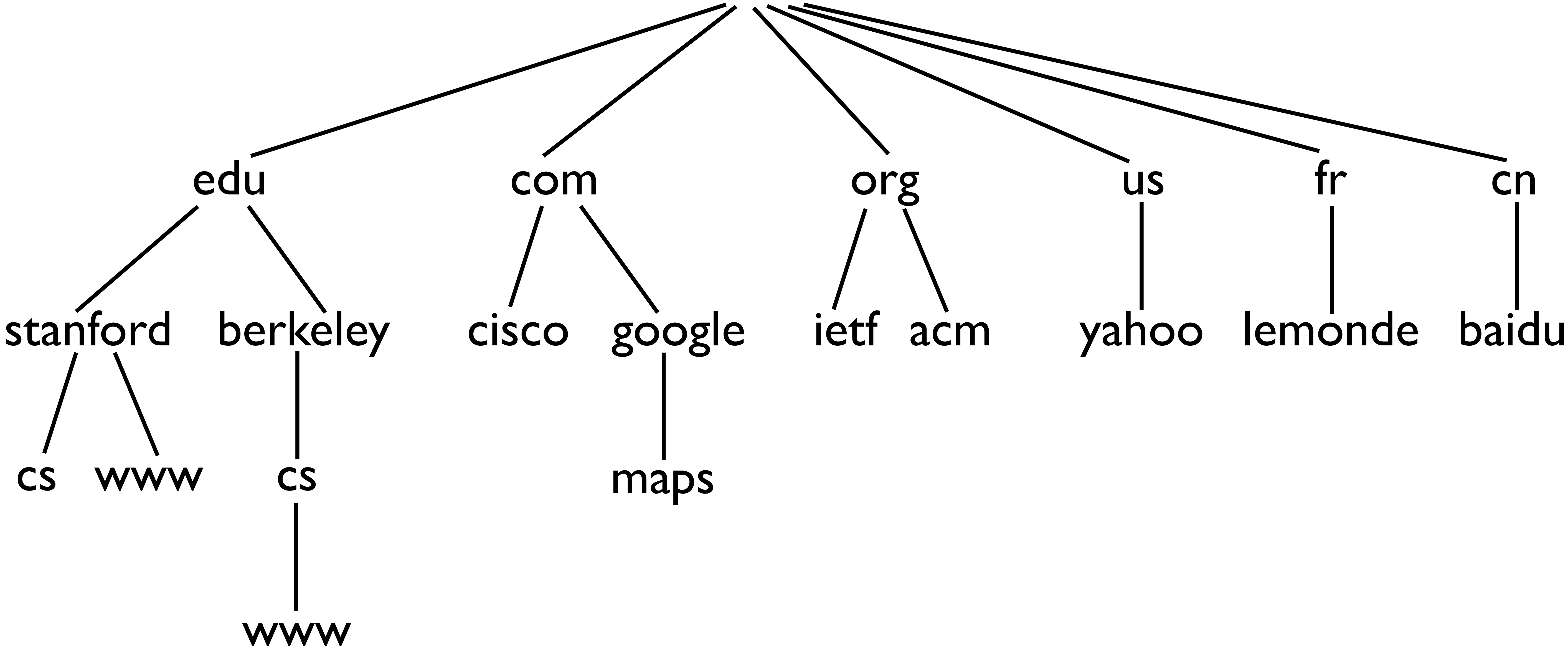
Domain Name System

- Map names to addresses (more generally, values)
- Must be able to handle *huge* number of records
- Must have distributed control: people can control their own names
- Must be robust to individual node failures

Domain Name System Design

- Two properties make DNS design feasible
  - Read-only or read-mostly database: hosts look up names much more often than update them
  - Loose consistency: changes can take a little while to propagate
- Two properties allow extensive caching
  - Look up a name, keep result for a long time

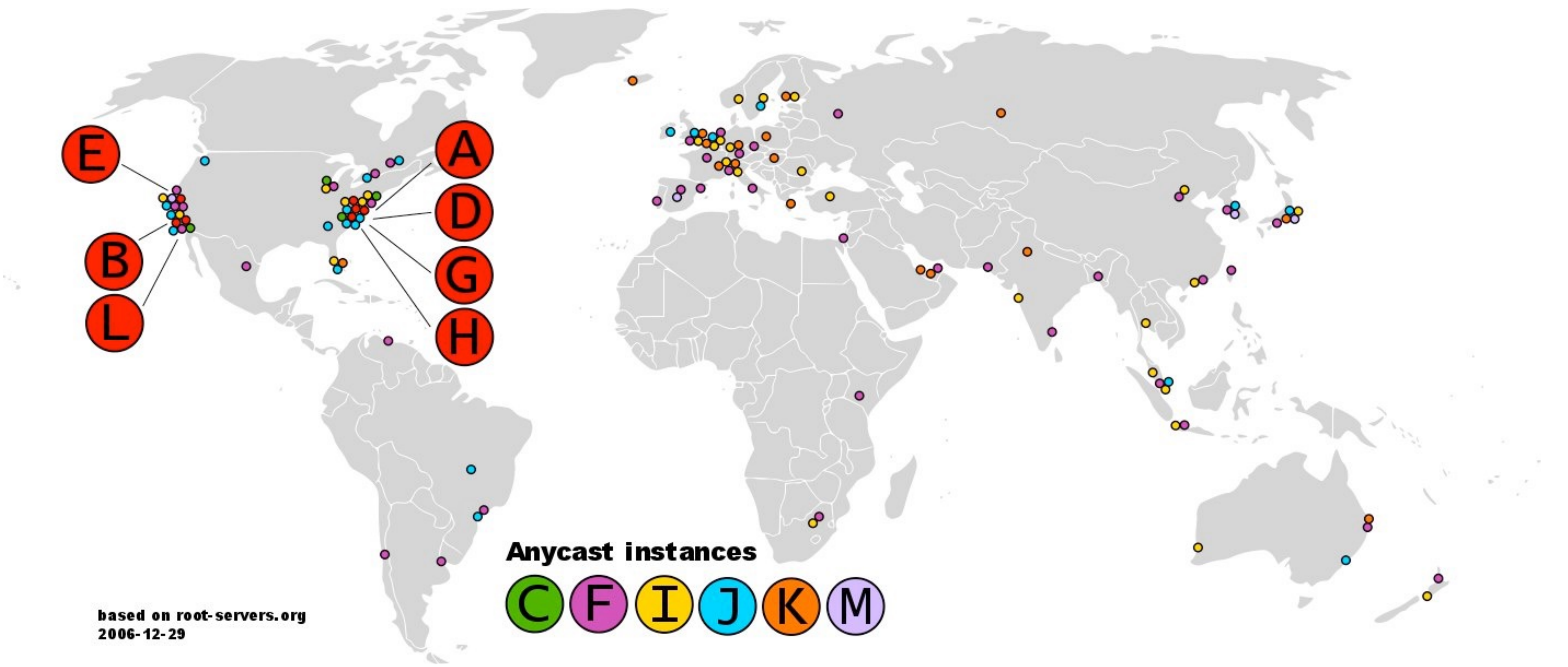
DNS Name Architecture



DNS Servers

- Hierarchical zones (“root” zone, edu, stanford, scs)
- Each zone can be separately administered
- Each zone served from several replicated servers
- Root zone: 13 servers, highly replicated (a, b, c, ... m)
  - Bootstrap: root server IPs are stored in a file on name server
  - Replicated through anycast (discussed later in course)

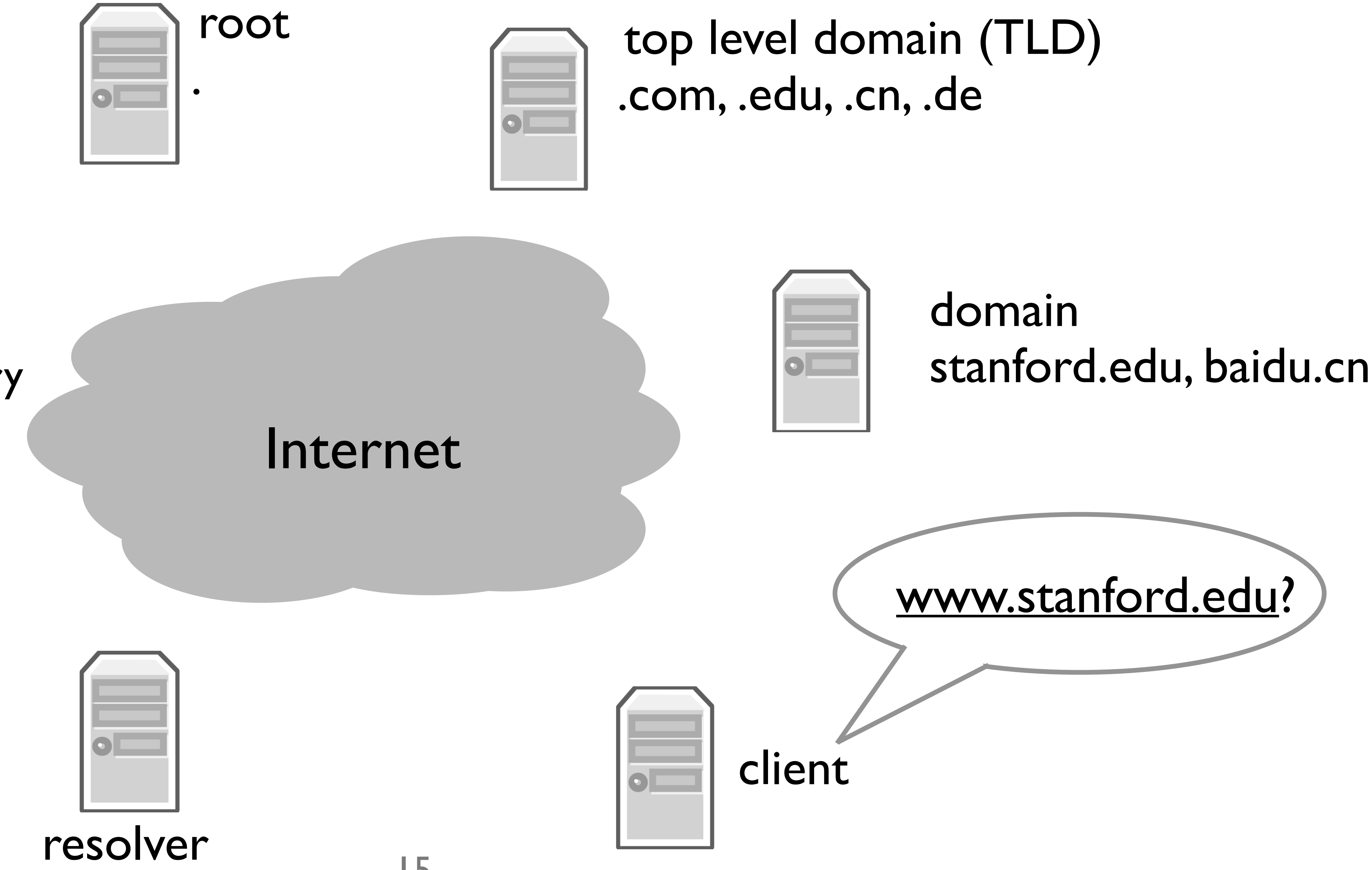
DNS Root Servers



http://en.wikipedia.org/wiki/File:Root-current.svg

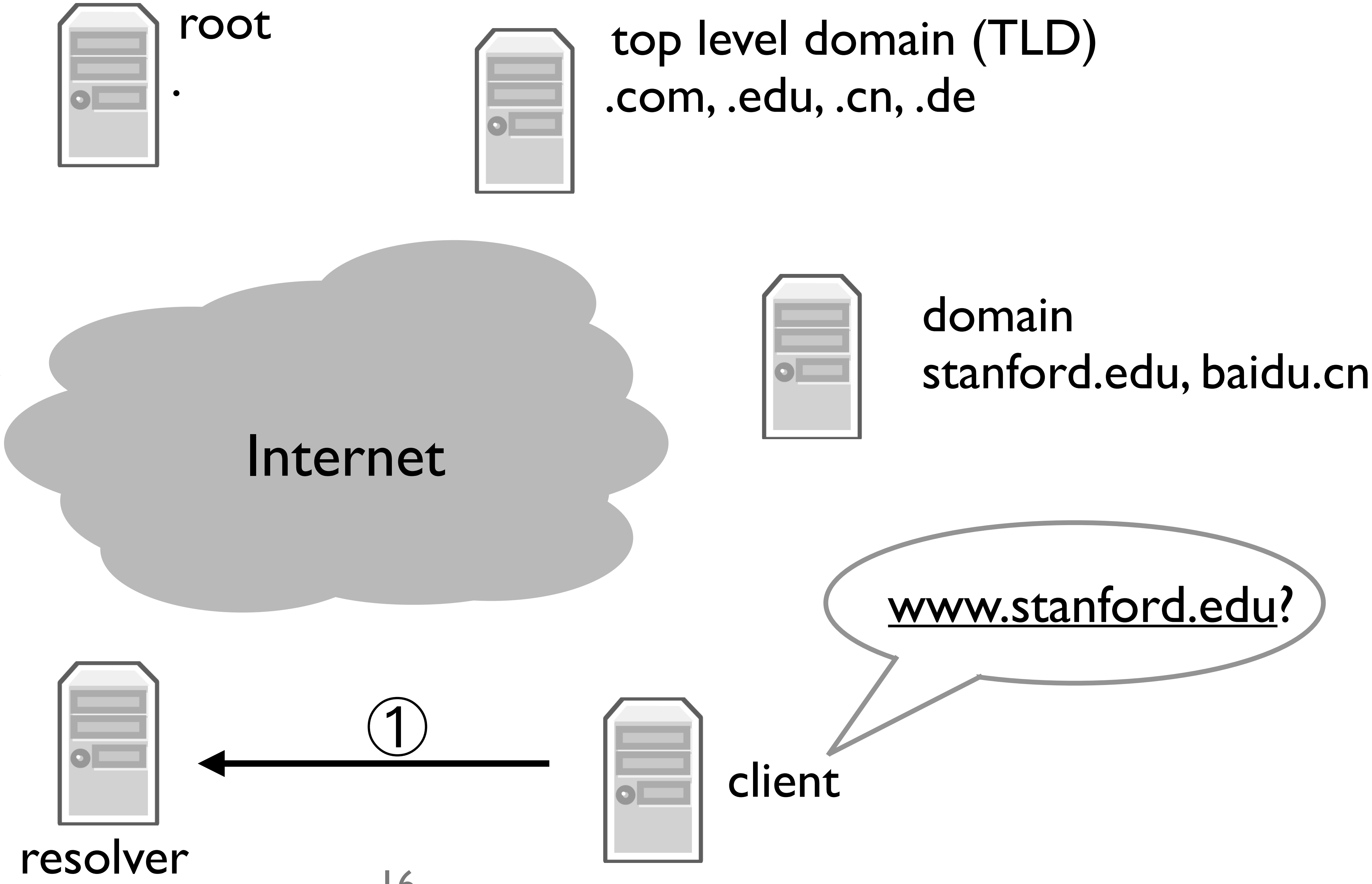
A DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



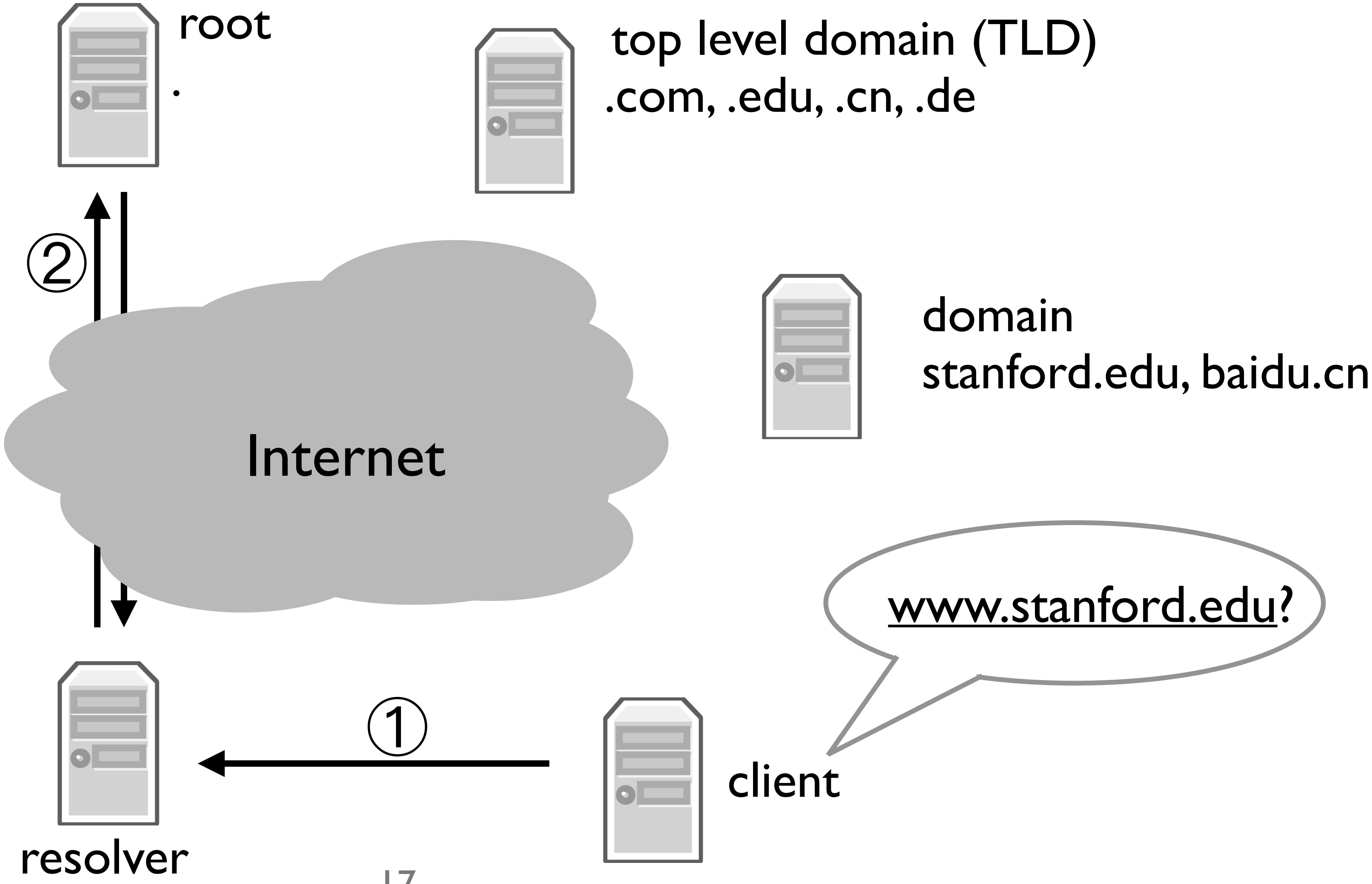
A DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



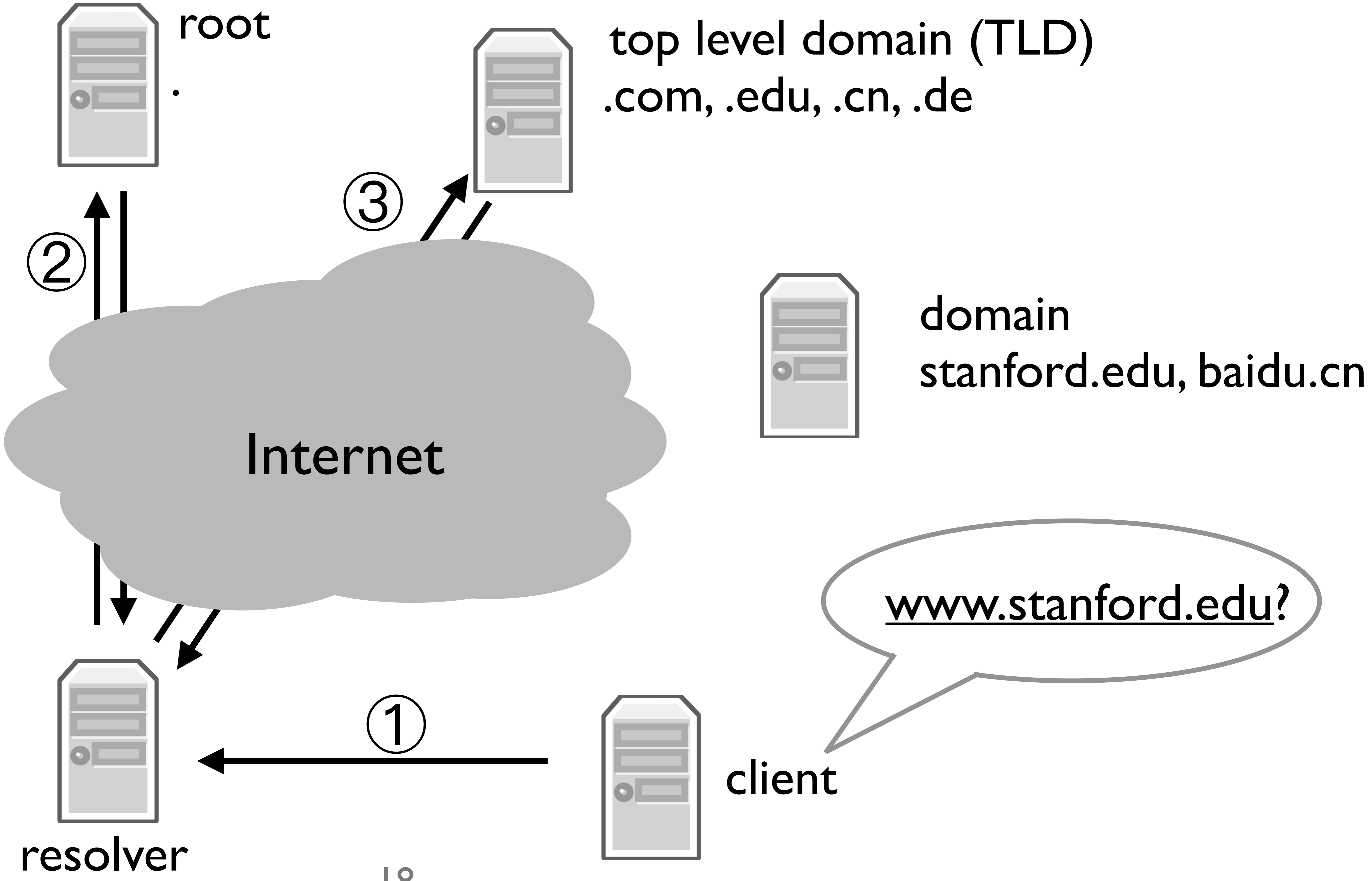
A DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



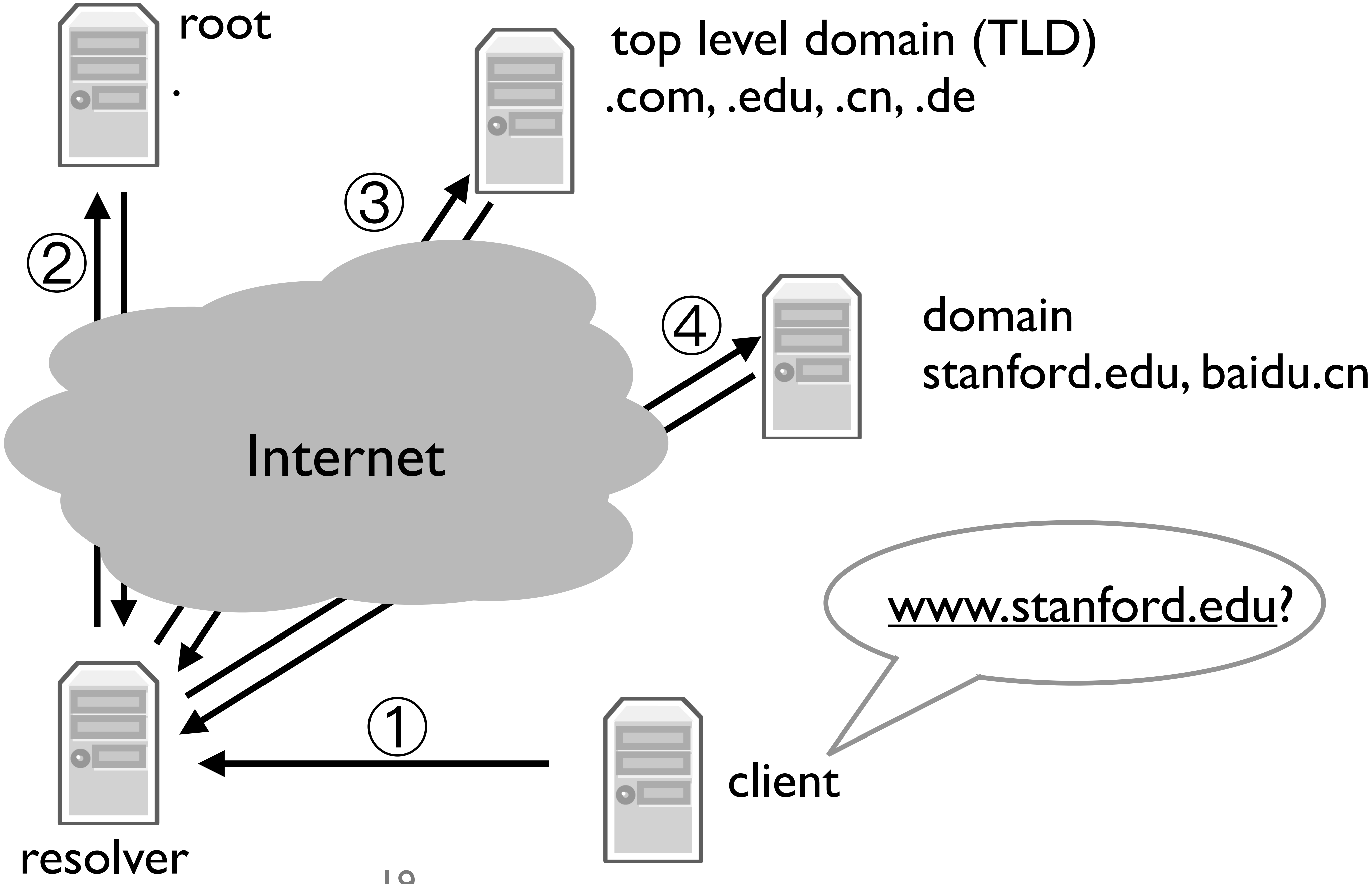
A DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



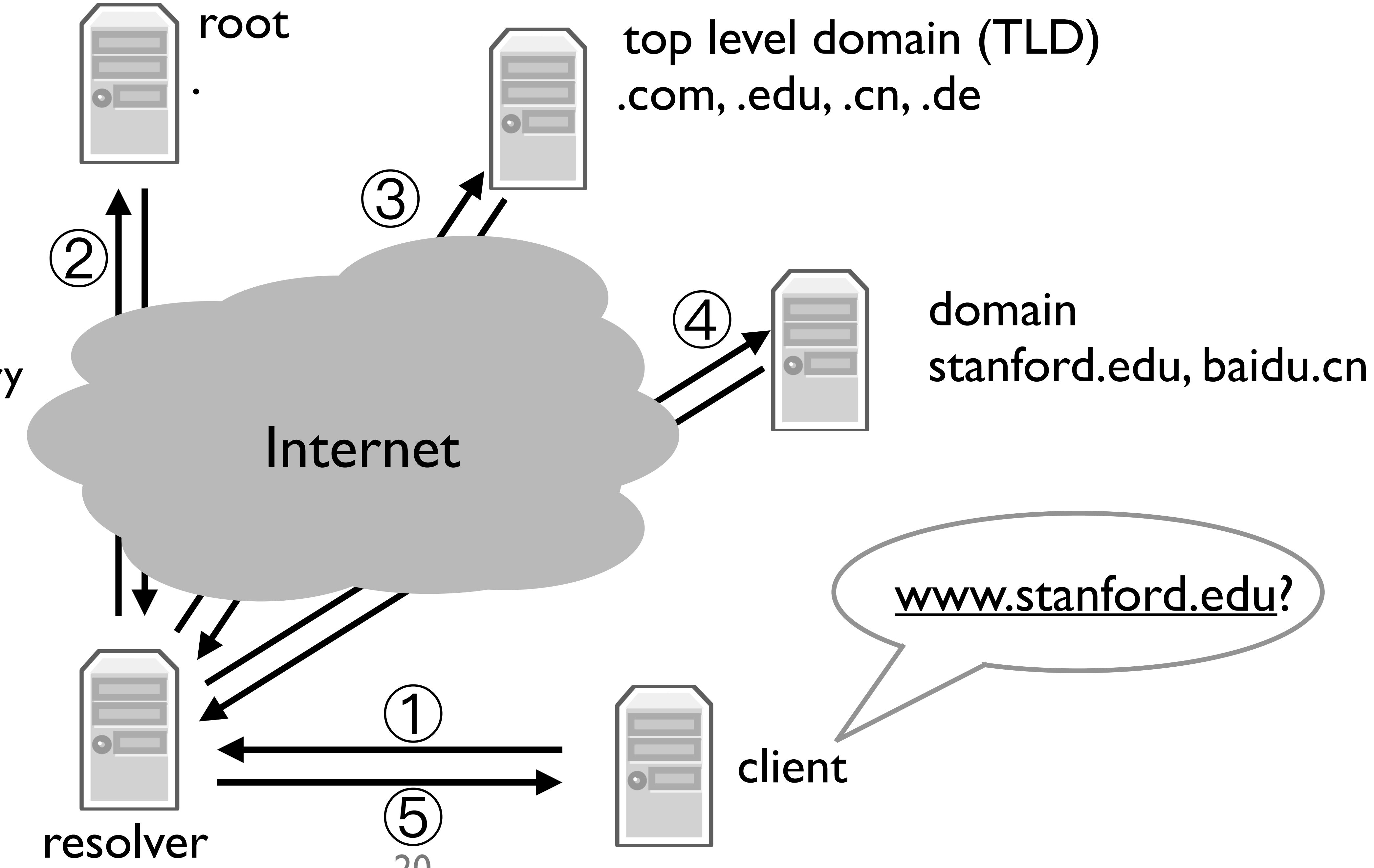
A DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



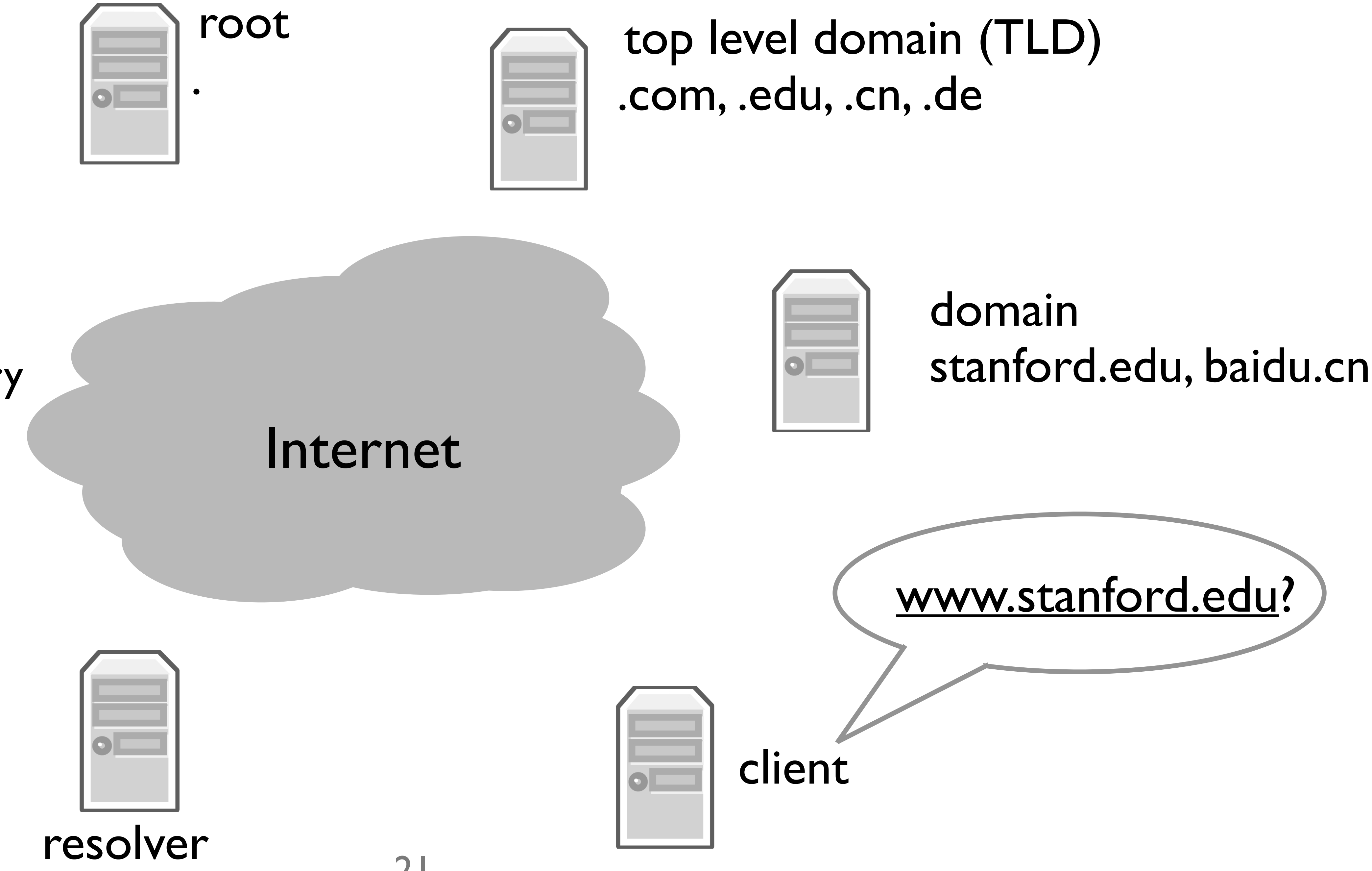
A DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



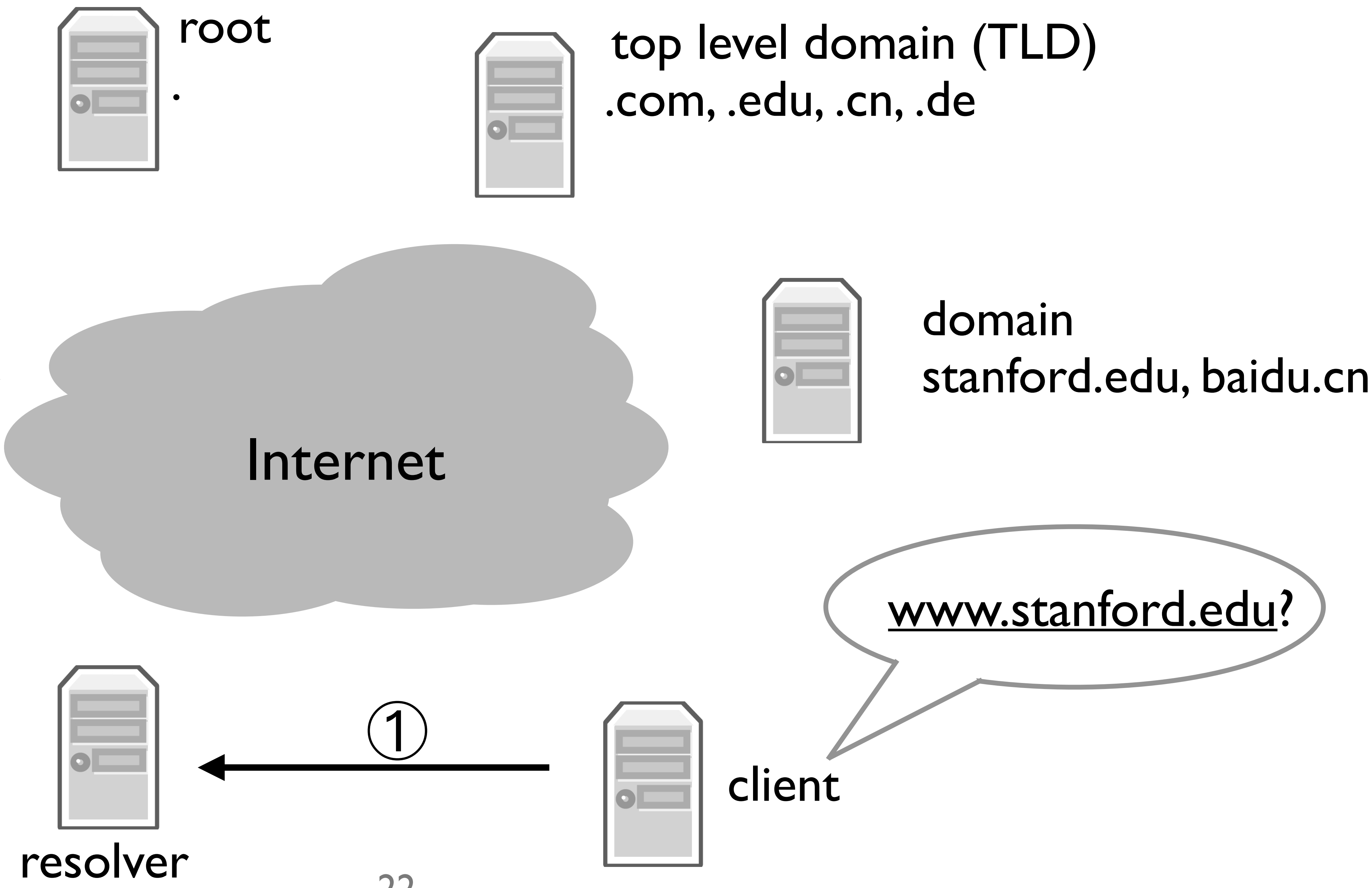
A Repeated DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



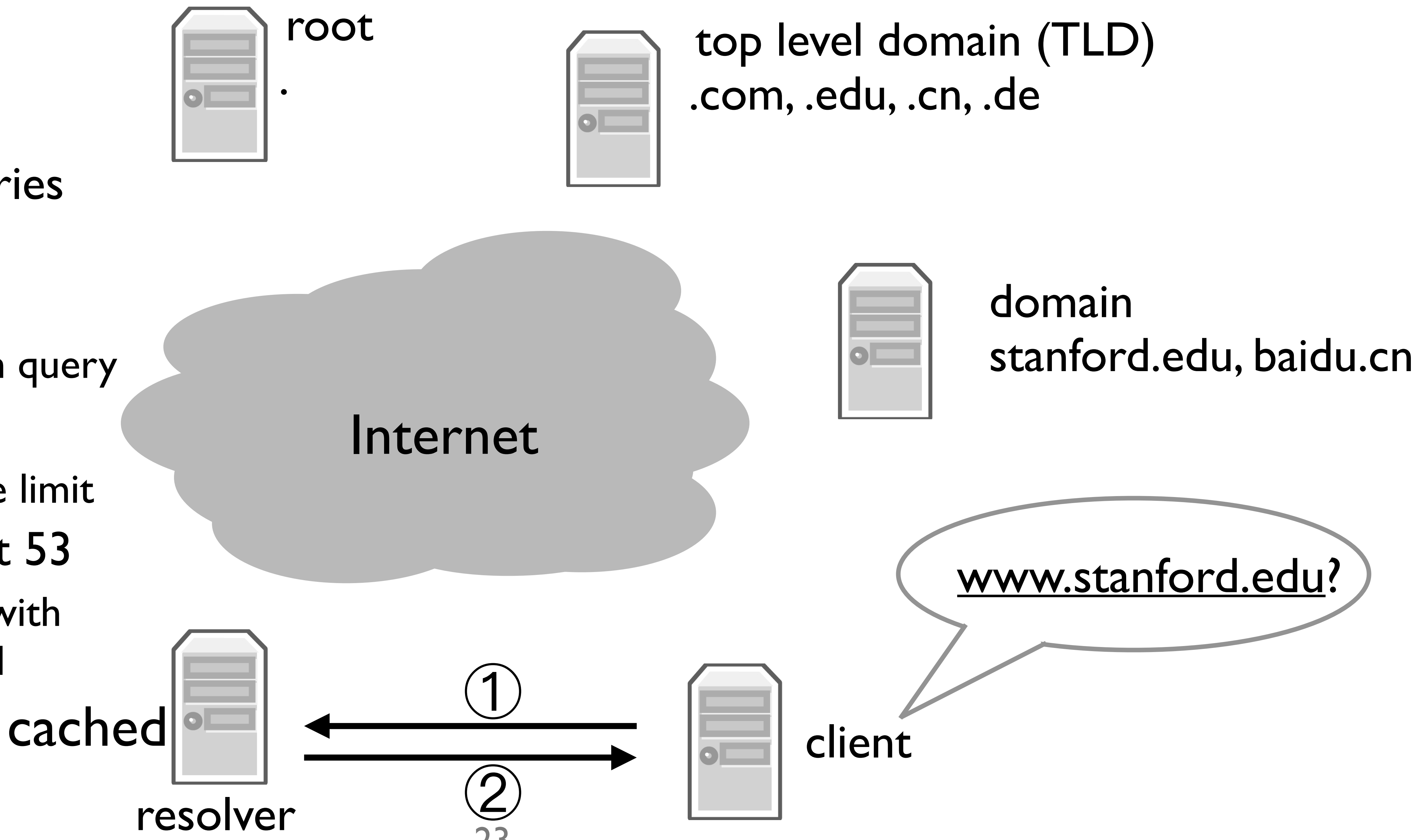
A Repeated DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



A Repeated DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



Resource Records

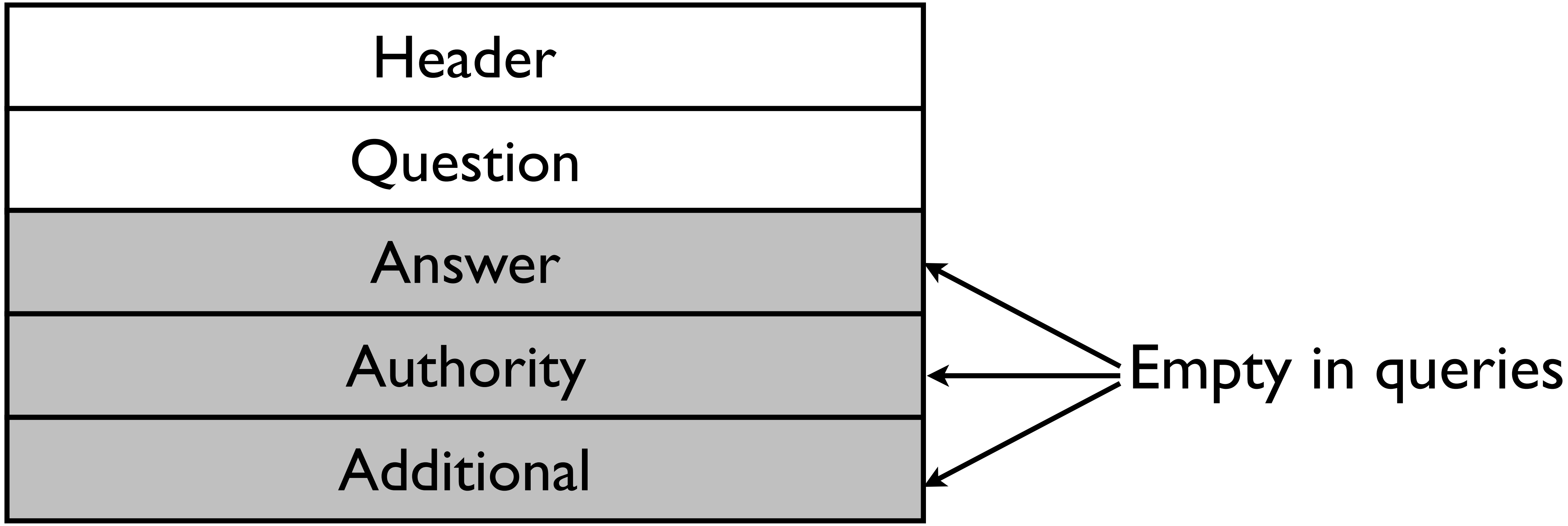
- All DNS information represented in Resource Records (RRs):

name [TTL] [class] type rdata

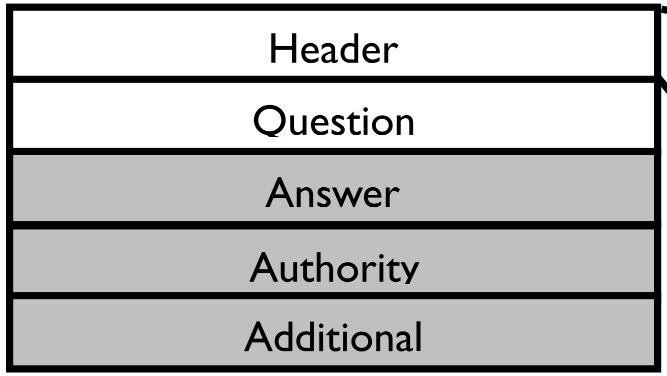
  - *name*: domain name (e.g., www.stanford.edu)
  - *TTL*: time to live (in seconds)
  - *class*: for extensibility, usually `IN 1` (Internet)
  - *type*: type of the record
  - *rdata*: resource data dependent on *type*

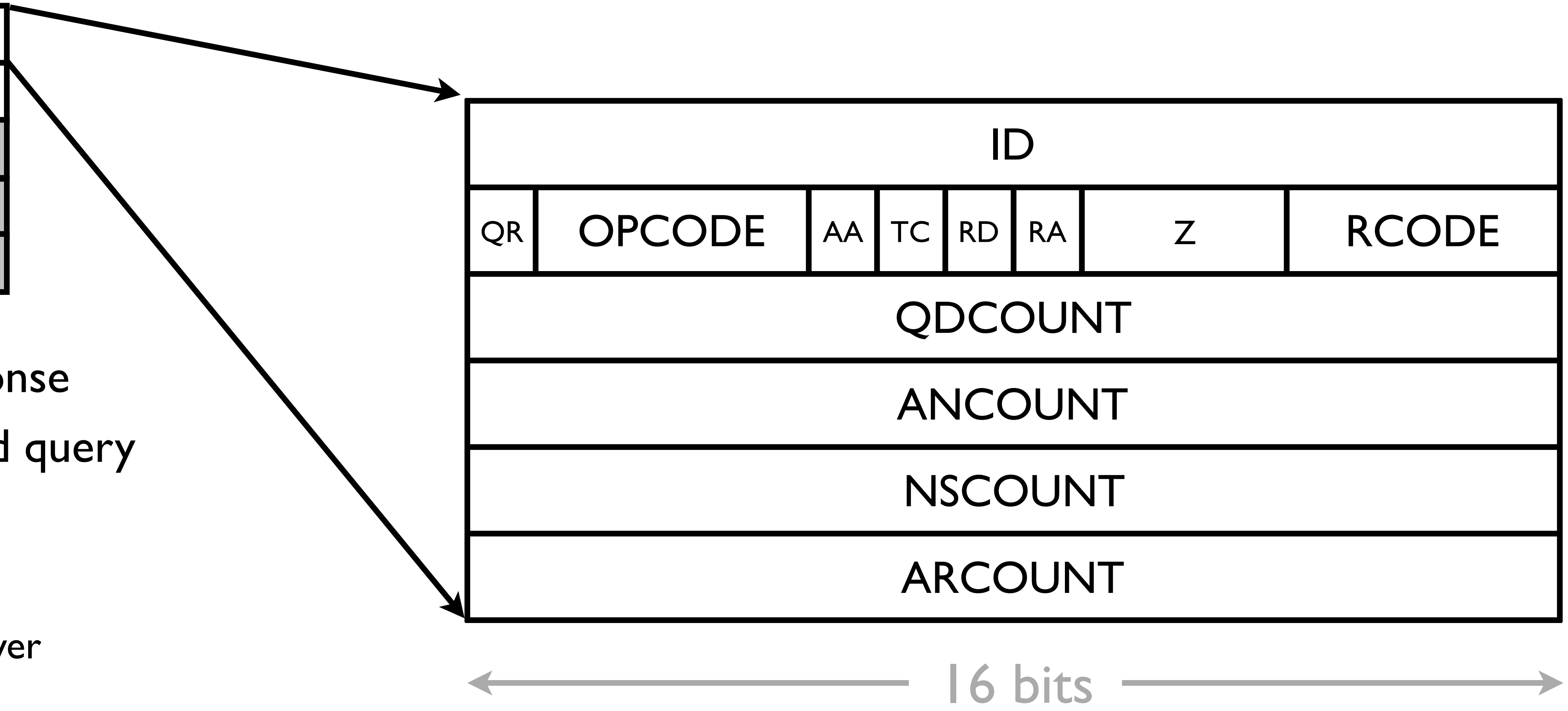
- Two critical RR types: A (IPv4 address) and NS (name server) records
- `dig` tool

DNS Message Structure (RFC1035)



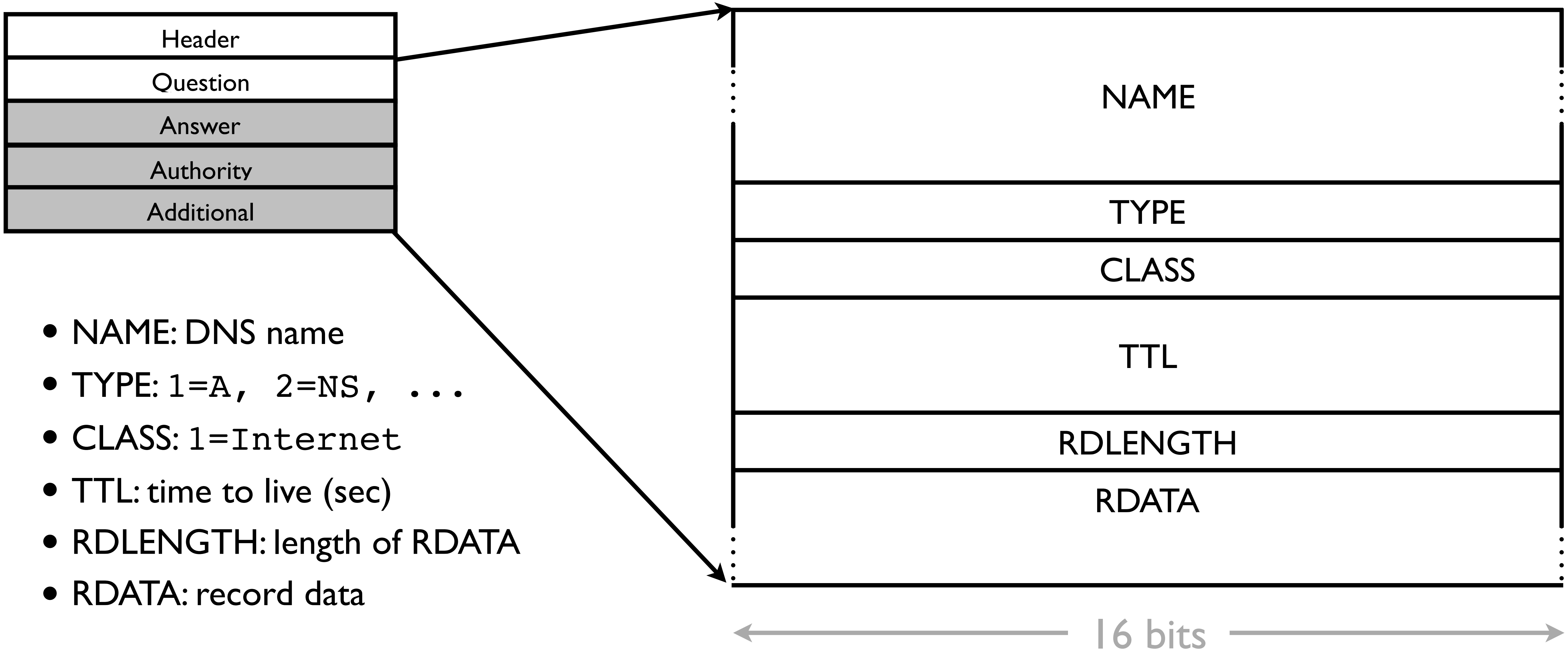
DNS Header Structure (RFC1035)





- QR: 0=query, 1=response
- OPCODE: 0=standard query
- RCODE: error code
- Flags
  - AA: authoritative answer
  - TC: truncated
  - RD: recursion desired
  - RA: recursion available

DNS RR Structure (RFC1035)

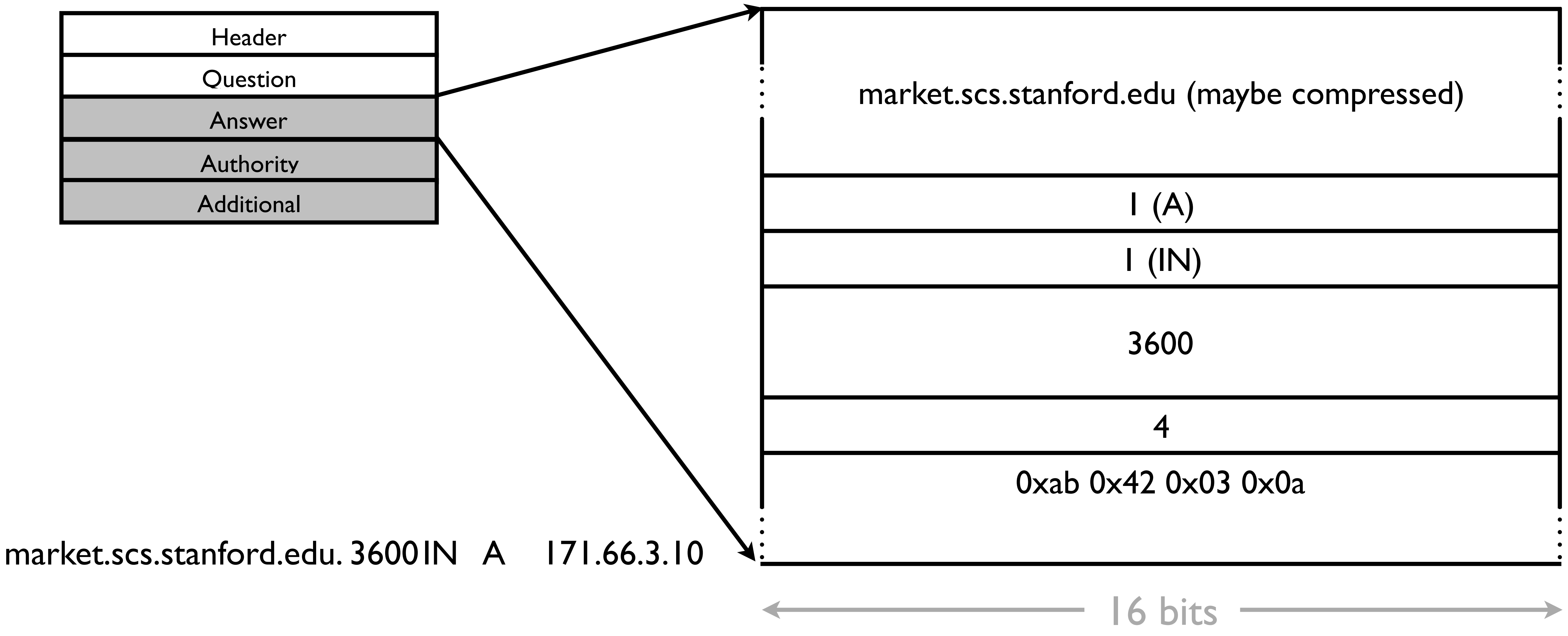


- NAME: DNS name
- TYPE: `1=A, 2=NS, ...`
- CLASS: `1=Internet`
- TTL: time to live (sec)
- RDLENGTH: length of RDATA
- RDATA: record data

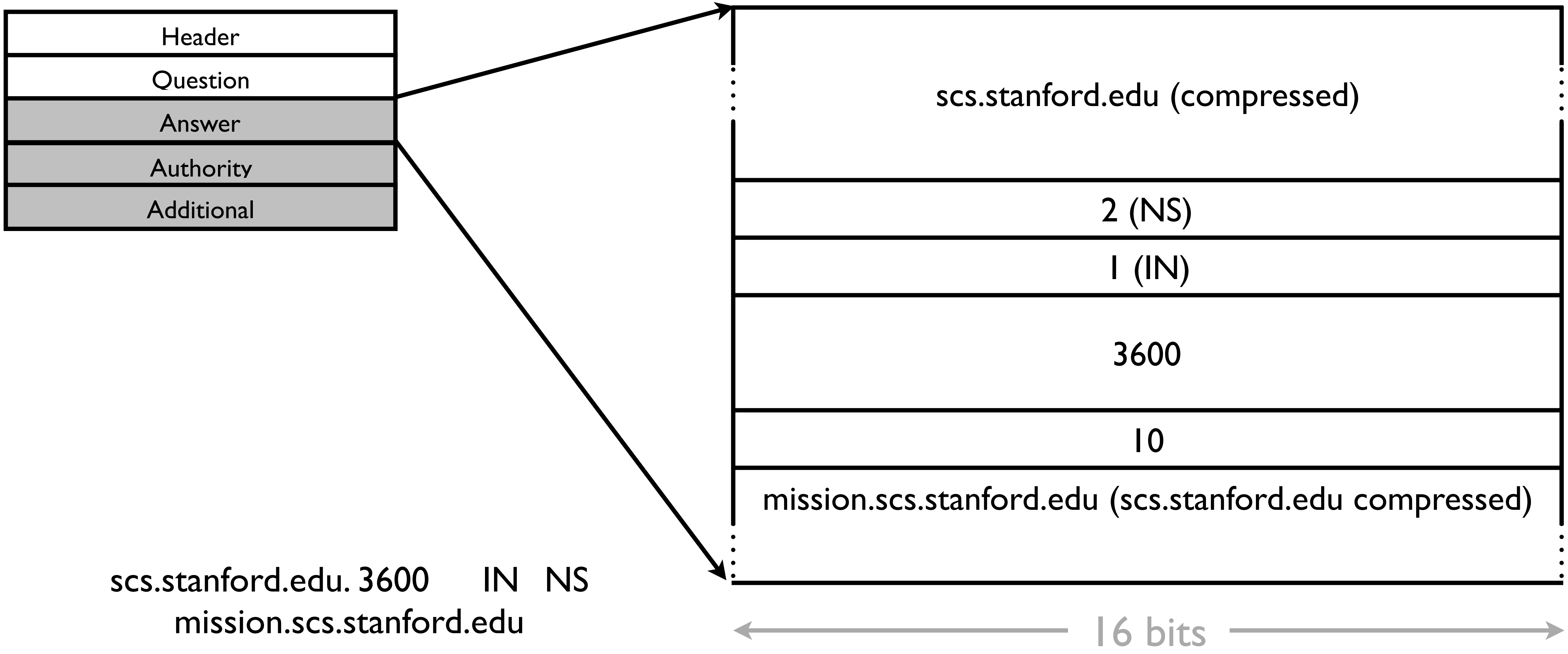
DNS Name Compression

- Names can be long and repeated several times in a packet
  - Query/answer
  - NS record/A record
- Break names into labels: www.stanford.edu is www, stanford, and edu
- Each label is encoded as length, text: 3www, 8stanford, 3edu
  - Length is binary
  - Text is ASCII: 3www is 0x0377 0x7777
- If length >= 192, next 14 bits specifies offset in packet of name
  - 0xc00c means name is at offset 0xc00c-0xc000 = 0x0c = 12

DNS A Record

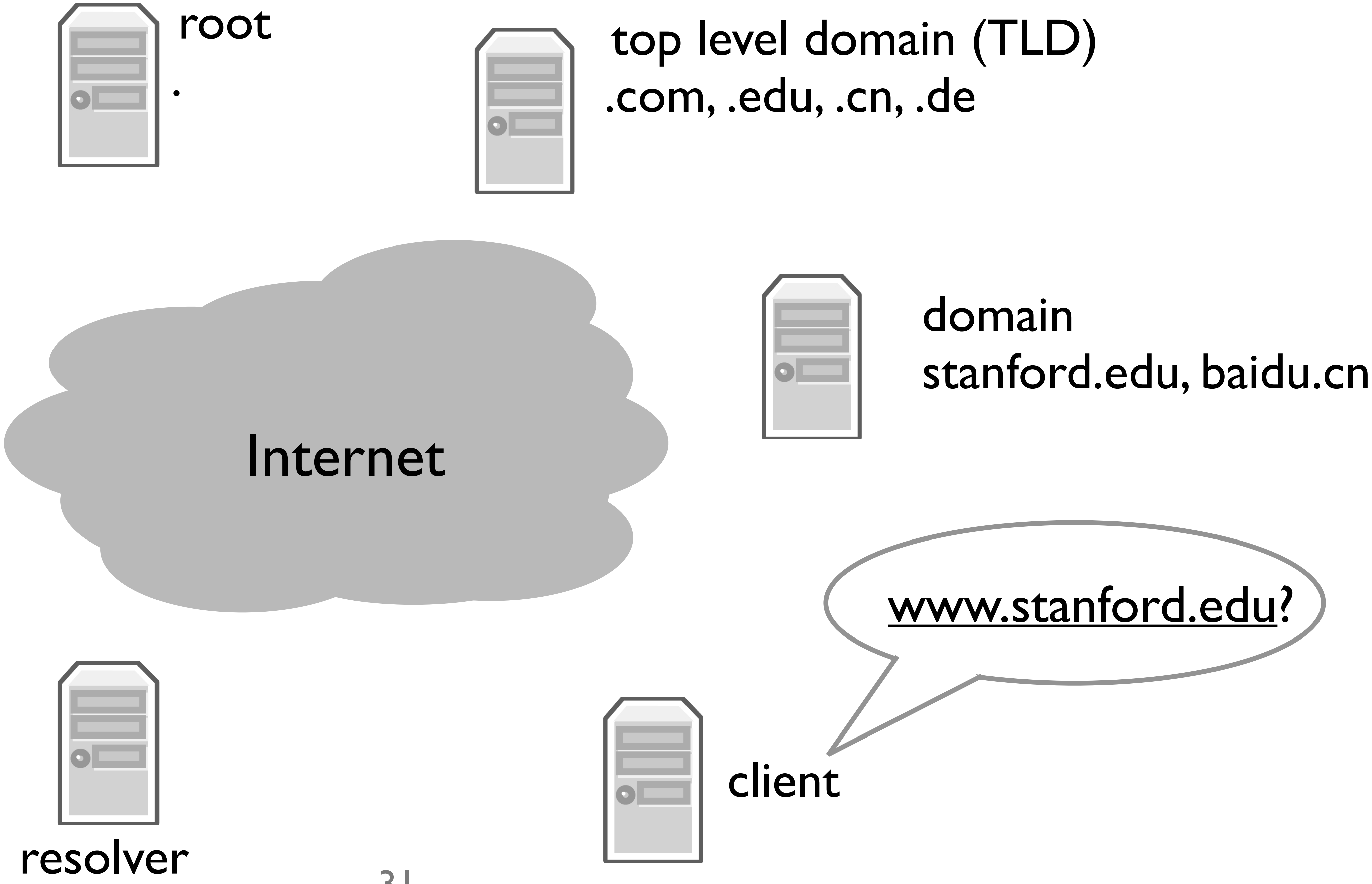

Header
Question
Answer
Authority
Additional
market.scs.stanford.edu (maybe compressed)
1 (A)
1 (IN)
3600
4
0xab 0x42 0x03 0x0a
market.scs.stanford.edu. 3600 IN A 171.66.3.10
16 bits

DNS NS Record



scs.stanford.edu. 3600 IN NS
mission.scs.stanford.edu

A DNS Query



- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field

Dyn Attack

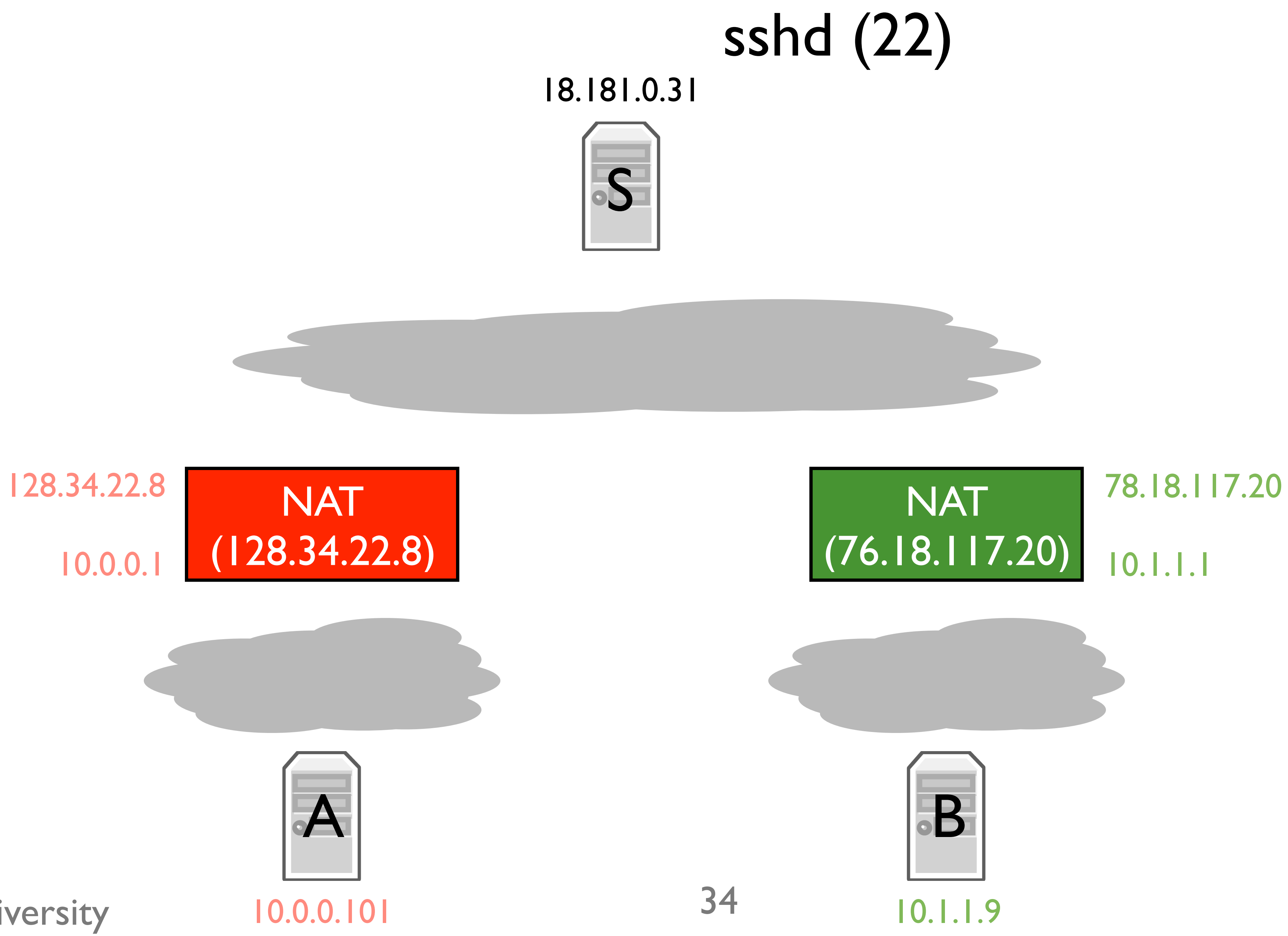
- Several major sites taken offline for east coast last Friday morning
  - Dyn is a DNS provider: clients couldn't query Dyn for A records
- "Drew says the attack consisted mainly of TCP SYN floods aimed directly at against port 53 of Dyn's DNS servers, but also a prepend attack, which is also called a subdomain attack. That's when attackers send DNS requests to a server for a domain for which they know the target is authoritative. But they tack onto the front of the domain name random prepends or subnet designations. The server won't have these in its cache so will have to look them up, sapping computational resources and effectively preventing the server from handling legitimate traffic, he says."

http://www.networkworld.com/article/3134057/security/how-the-dyn-ddos-attack-unfolded.html

Network Address Translator (NAT)

RFC1631

NAT Example

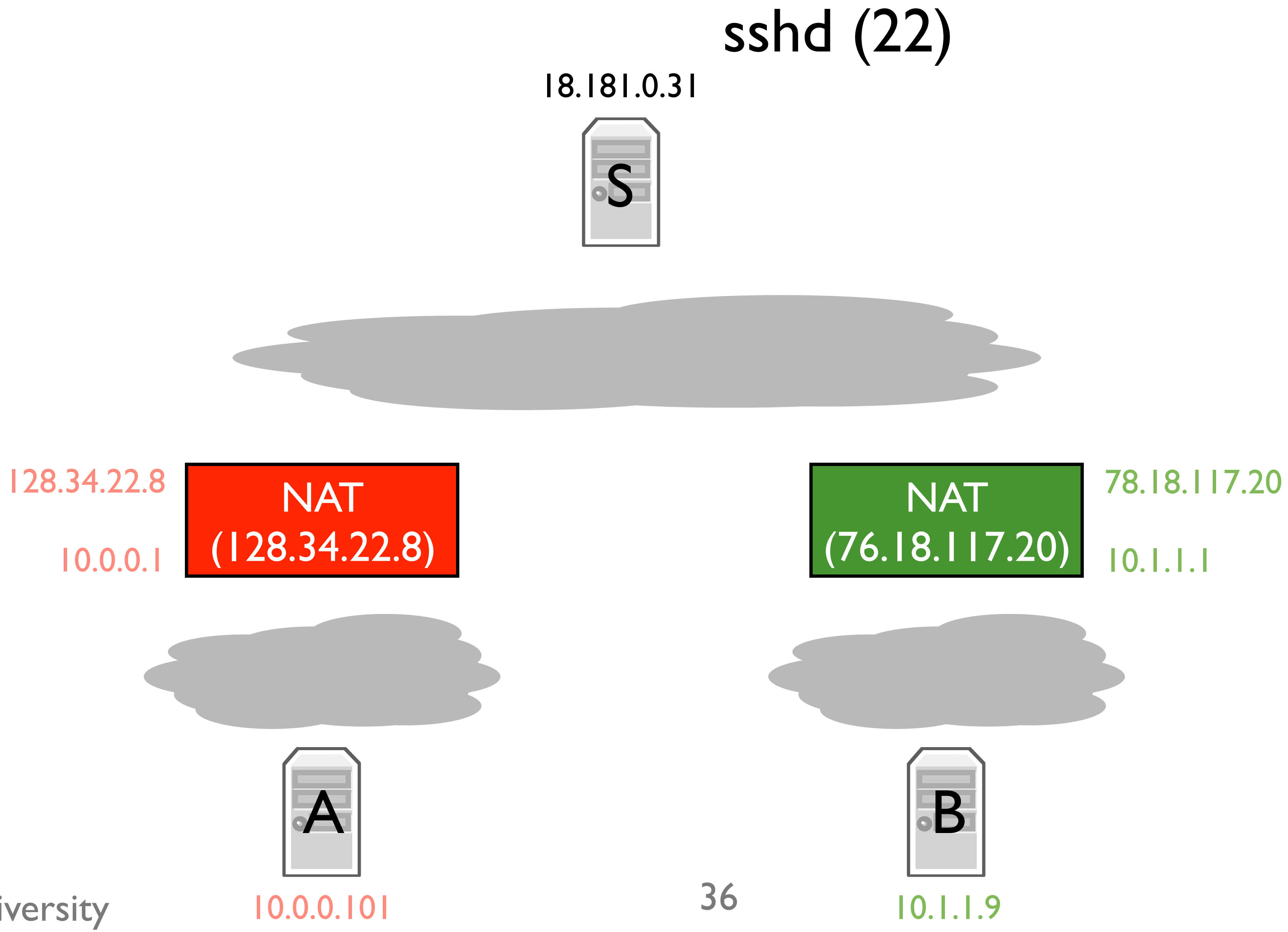


sshd (22)
18.181.0.31
S
128.34.22.8
10.0.0.1
NAT
(128.34.22.8)
NAT
(76.18.117.20)
78.18.117.20
10.1.1.1
A
B
10.0.0.101
10.1.1.9

NAT Benefits and Complications

- Benefits
  - Can use private addresses: there are only 2^{32} IP addresses
  - Firewalls for security
- Complications
  - Breaks end-to-end (node does not know its external IP)
  - Node might not even know if it's behind a NAT
  - Incoming connections break easily
  - NAT must be aware of transport layer
- RFC 4787/BCP 127 defines recommended behavior

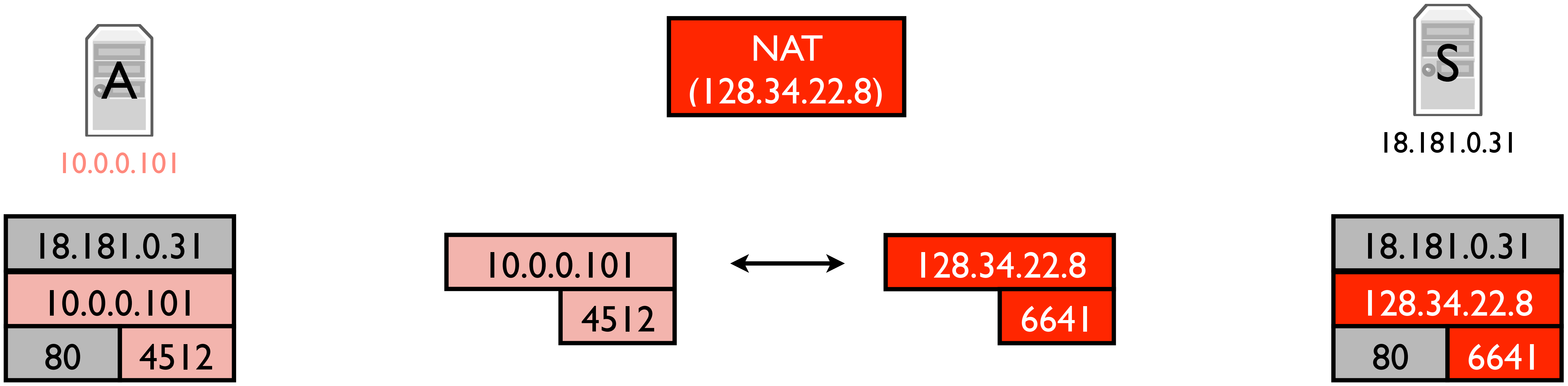
NAT Example


sshd (22)
18.181.0.31
S
128.34.22.8
10.0.0.1
NAT
(128.34.22.8)
NAT
(76.18.117.20)
78.18.117.20
10.1.1.1
A
B
10.0.0.101
10.1.1.9

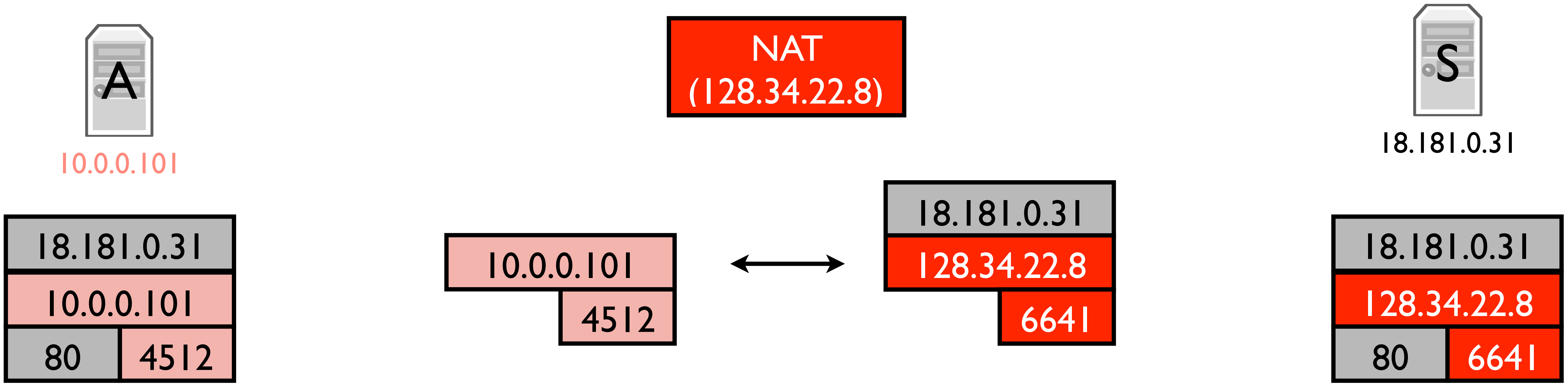
Two Questions

- What packets does a NAT allow to traverse mappings?
- How and when does a NAT assign mappings?
- NAT terminology/classification in RFC3489

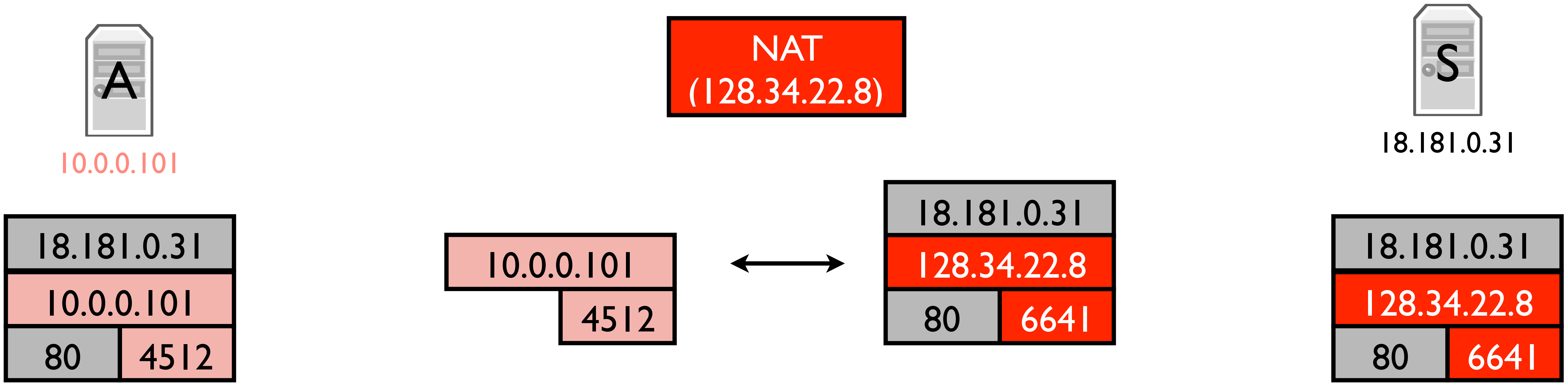
Full Cone NAT



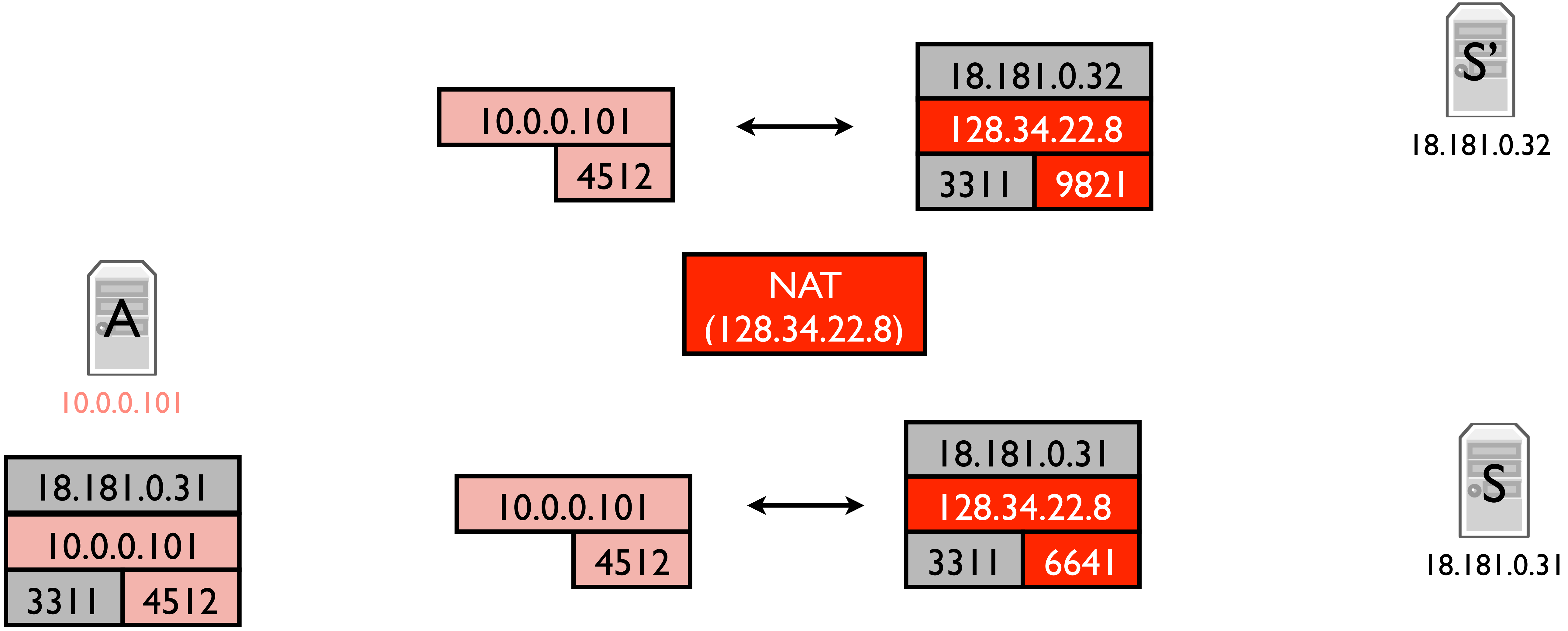
Restricted Cone NAT



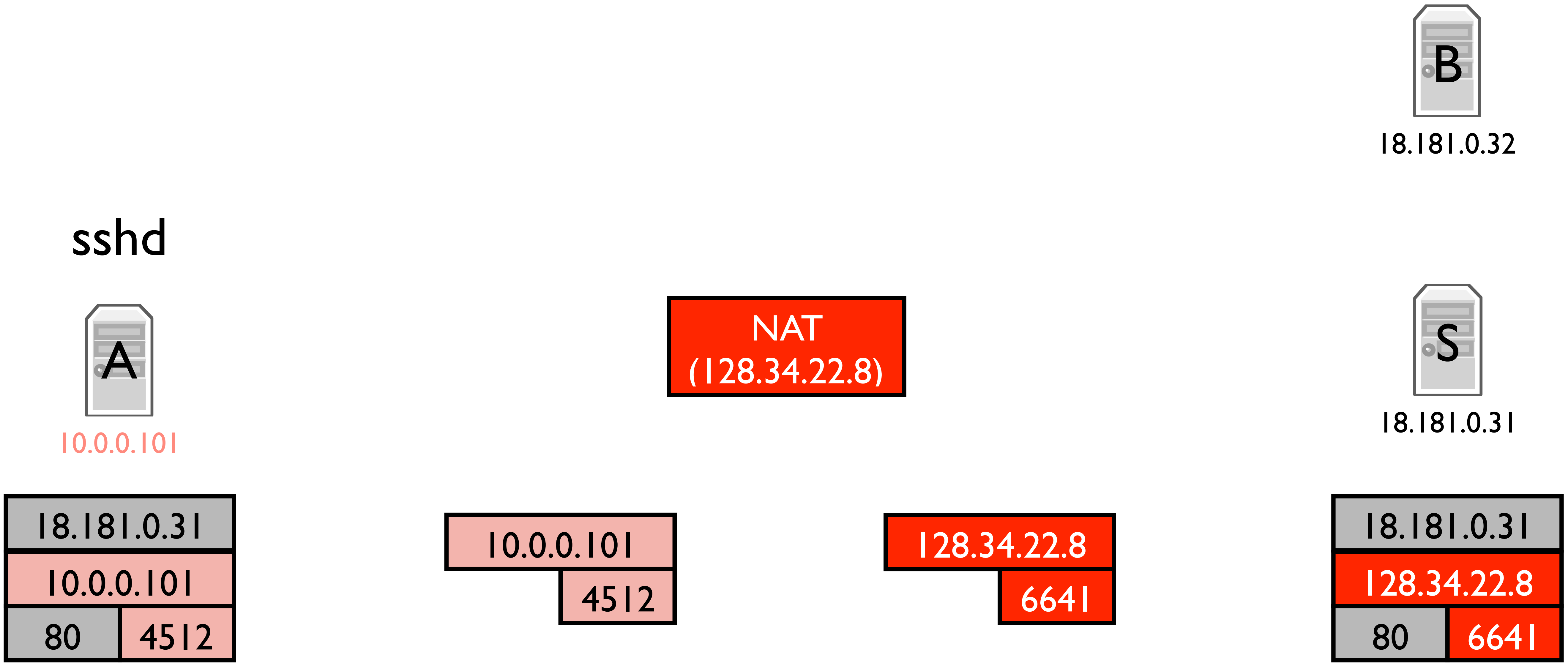
Port Restricted NAT



Symmetric NAT

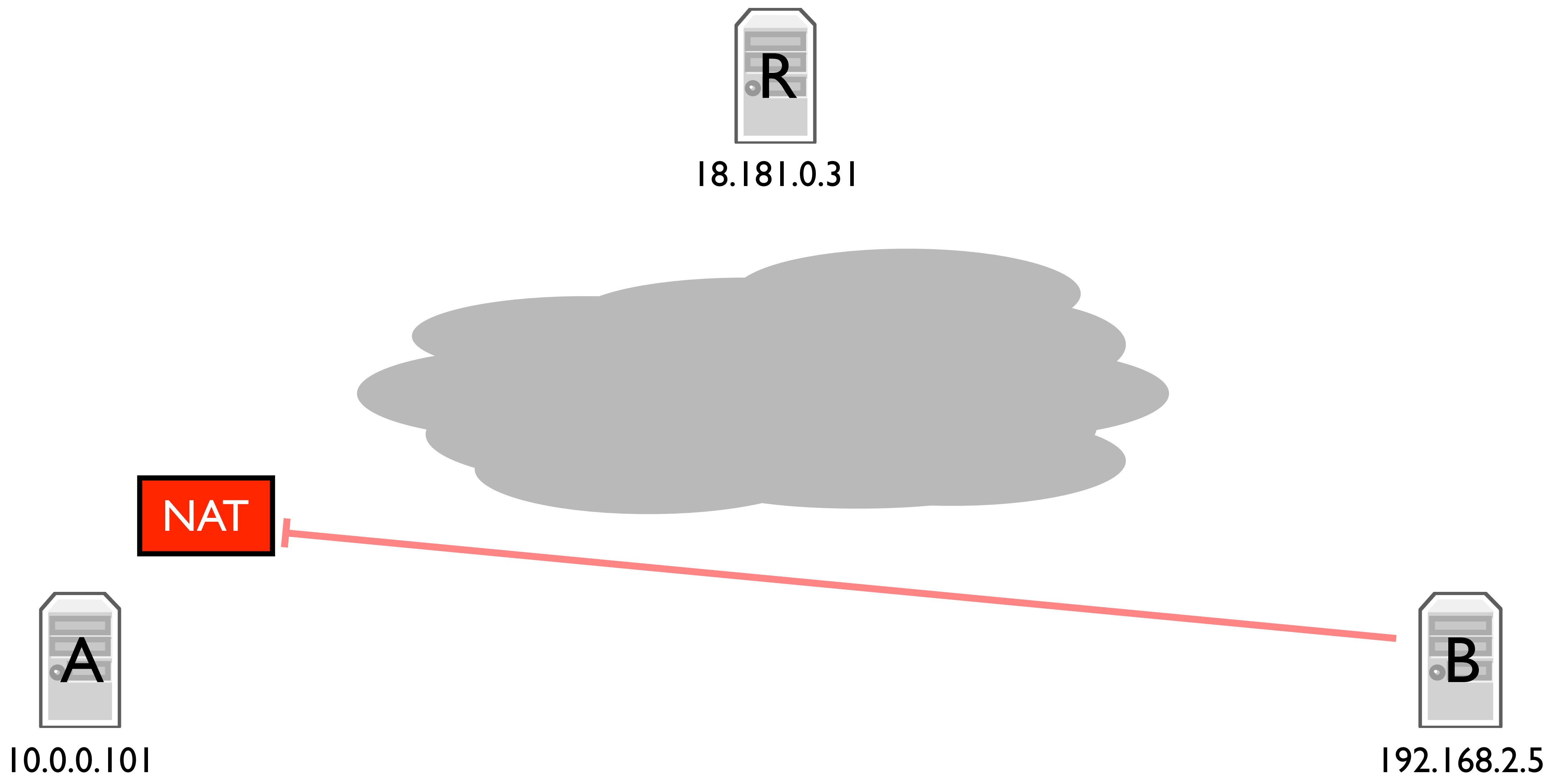


Applications: Incoming Connections

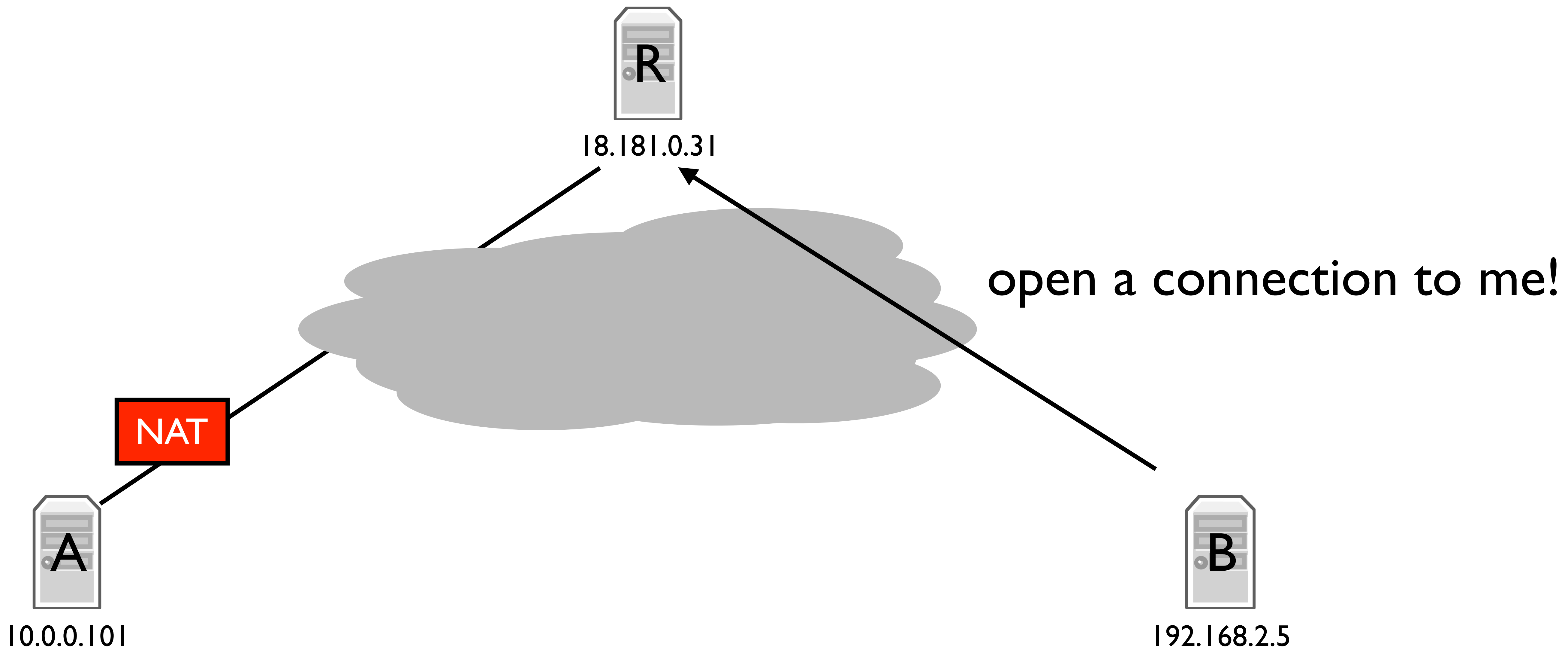


Connection Reversal

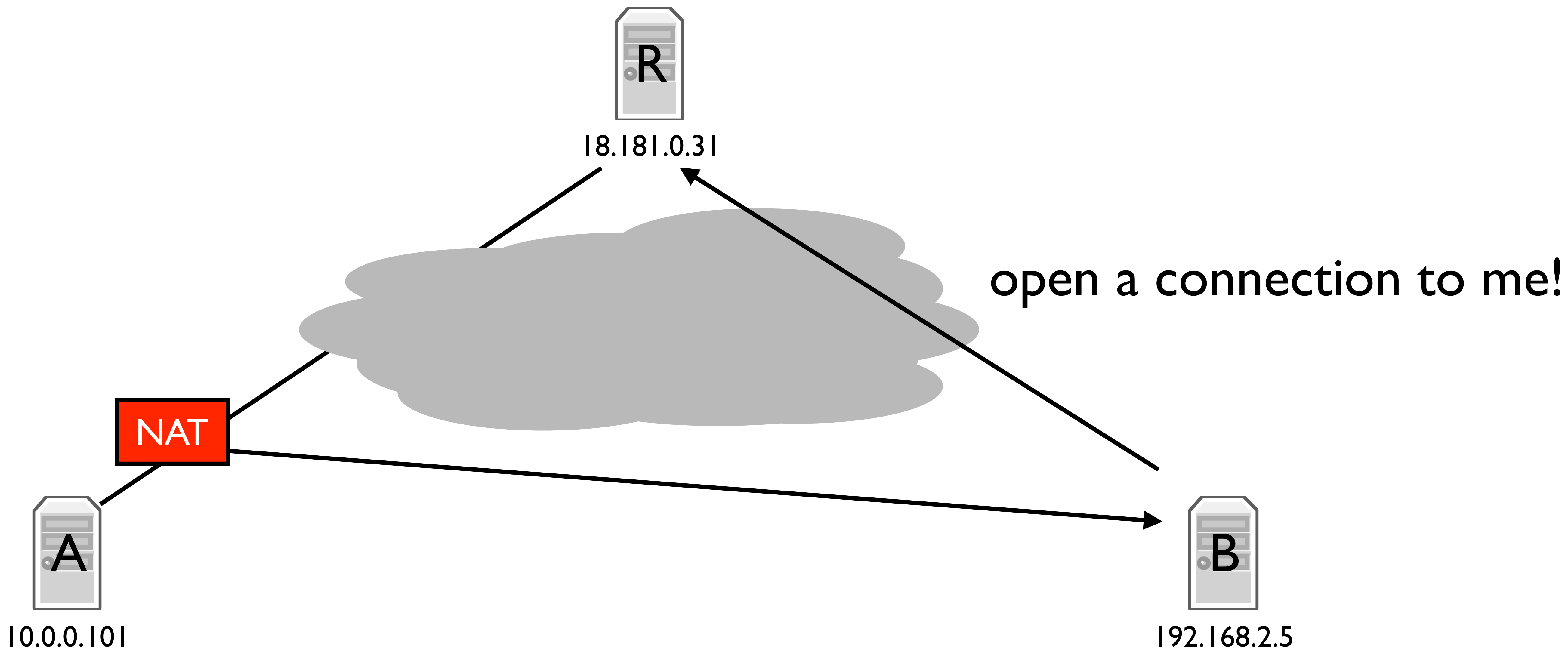
Connection Reversal



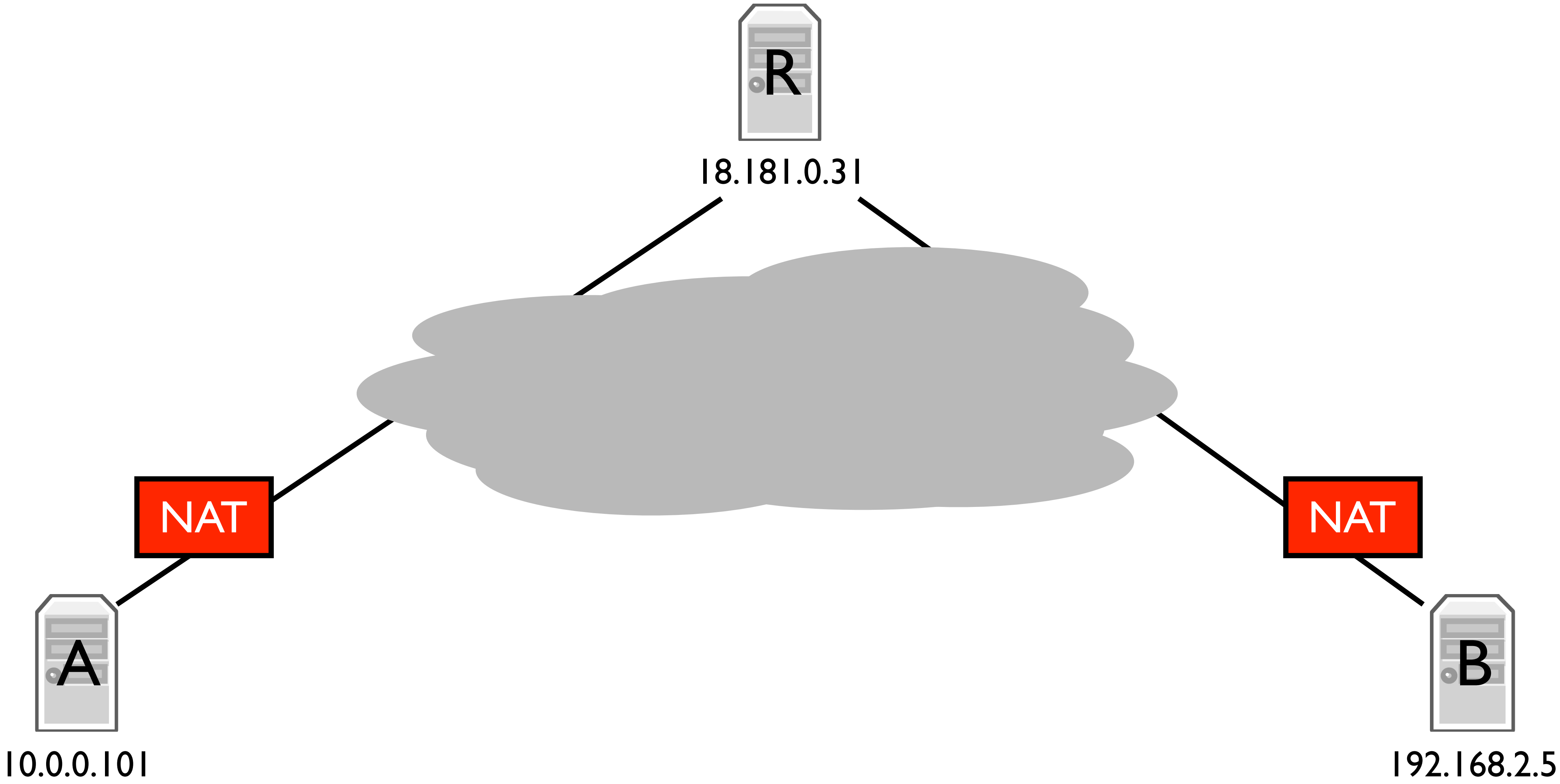
Connection Reversal



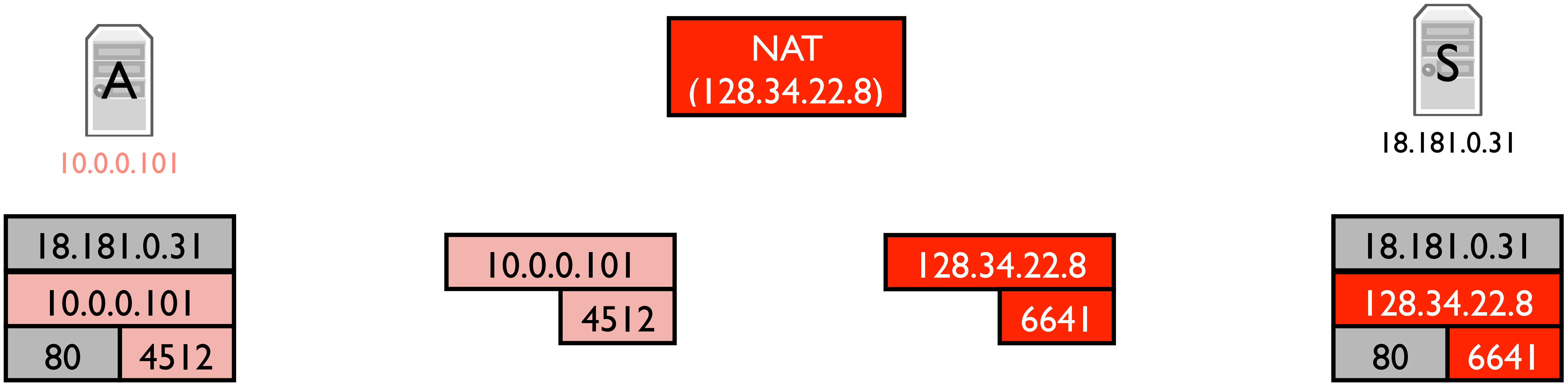
Connection Reversal



Relays



Transport: No New Transport!



NAT Debate

- Tremendously useful
  - Reuse addresses
  - Security (not opening connections can be good!)
- Tremendously painful
  - Large complication to application development
  - Speak Freely (pre-Skype VoIP!)
- Debate interesting but pointless: NATs are here to stay

The New Hourglass