Domain Name System (DNS)

- Internet communication requires IP addresses
- Humans prefer to use computer names
- Automated system available to translate names to addresses
- Known as *Domain Name System (DNS)*
- November 1987

 - RFC 1034: Informational

 - RFC 1035: Implementation details

DNS Functionality

- Given
  - Name of a computer
- Return
  - Computer’s internet (IP) address
- Method
  - Distributed lookup
  - Client contact server(s) as necessary

Domain Name Syntax

- Alphanumeric segments separated by dots
- Examples:
  - www.tcd.ie
  - ntrg.cs.tcd.ie
  - www.research.att.com
- Most significant part on the right

Domain Name Acquisition

- Organization
  - Chooses a desired name
  - Must be unique
  - Registers with central authority
  - Placed under one *top-level domain*
- Names subject to international law
  - Trademarks
  - Copyright

Top-Level Domains (TLDs)

• .com	commercial organization
• .edu	U.S. educational institution
• .gov	U.S. government organization
• .mil	U.S. military group
• .net	major network provider or other
• .org	organization other than above
• .arpa	temporary ARPA domain (still used)
• .int	international organization
• country code	A country (e.g. ie or gr (grrr))

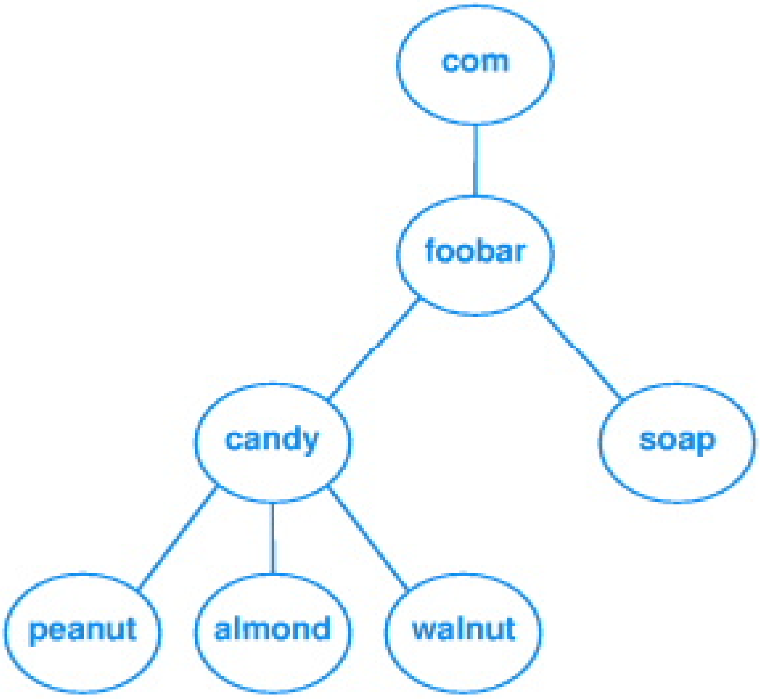
Within Top-Level Domains

- Subdivision possible
- Arbitrary levels possible
- Not standardized
- Controlled locally by organization

Example Name Structure

- First level: `com`
- Second level is company name: `cisco`
- Third level is division within company: `security`
- Fourth level either
  - Company subdivision: `crypto`
  - Individual computer: `smtp`

DNS Illustrated



DNS Key Concepts

- The number of segments in a domain name corresponds to the naming hierarchy
- There is no universal standard for this hierarchy; each organization can choose its own naming convention
- Furthermore, names within an organization do not need to follow a uniform pattern; individual groups within the organization can choose a hierarchical structure that is appropriate for that group

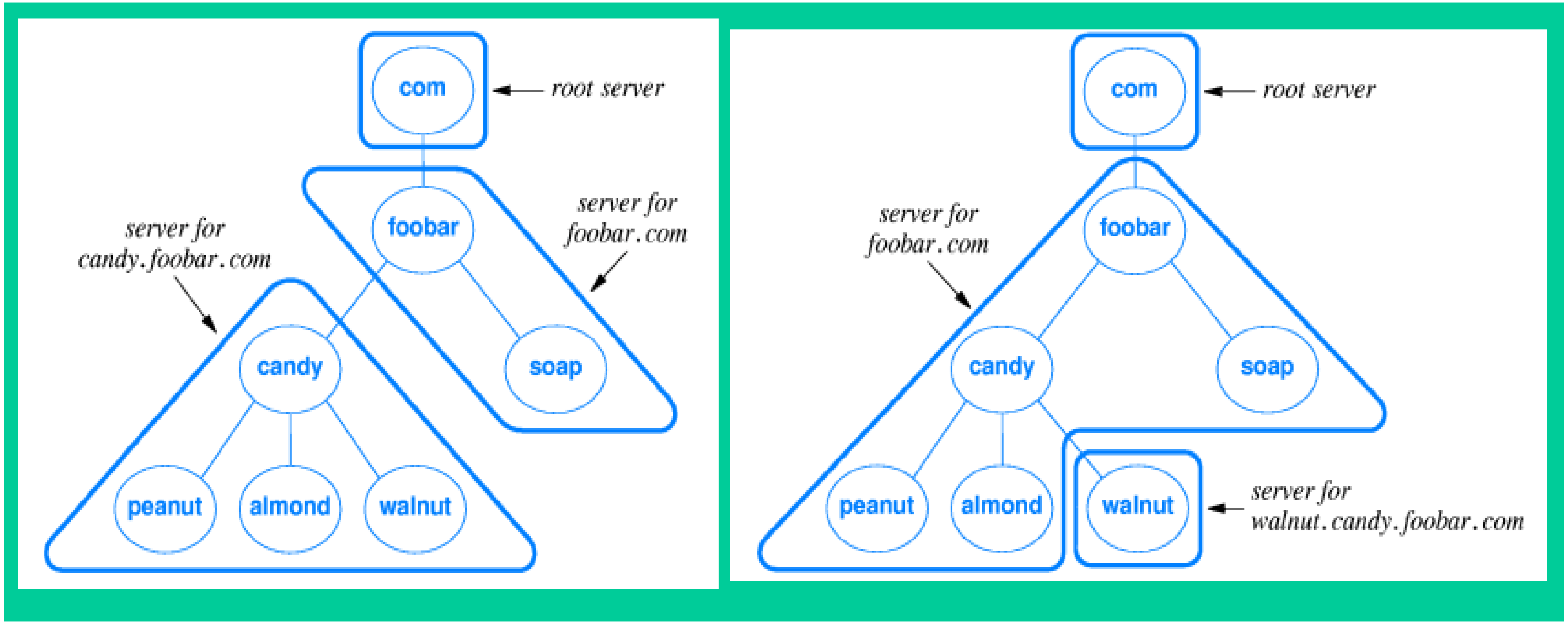
DNS Client/Server Interaction

- Client known as *resolver*
  - Actually a library that applications link against
- Multiple DNS servers used
- Arranged in a hierarchy
- Each server corresponds to an adjacent part of the global naming hierarchy

Inter-Server Links

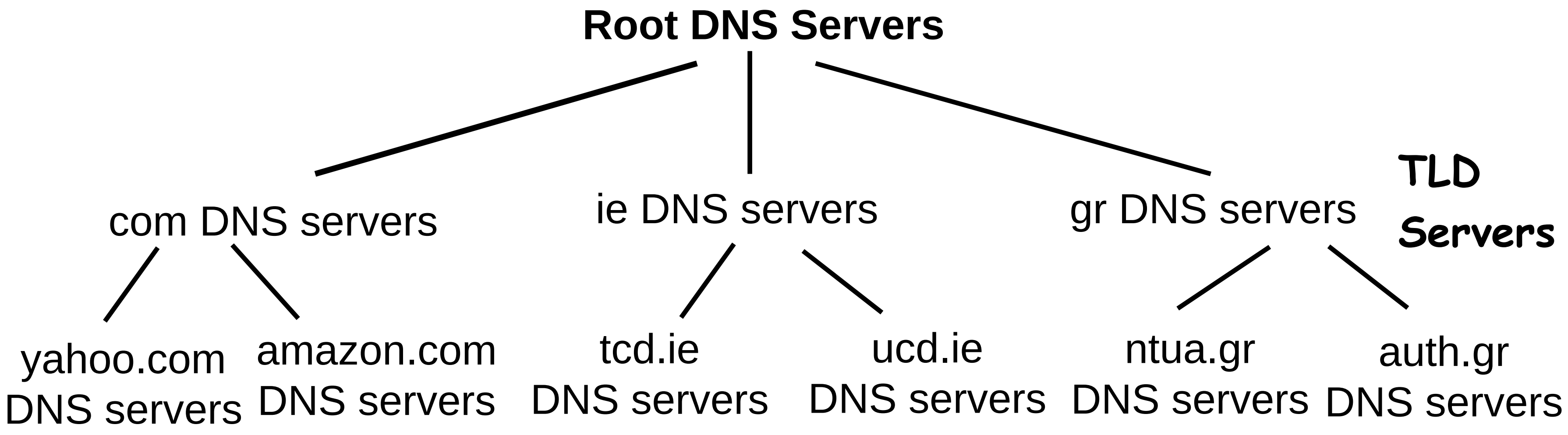
- All domain name servers are linked together to form a unified system
- Each server knows how to reach a root server, and
- How to reach servers that are authorities for names further down the hierarchy

DNS Hierarchy



- Choice made by organization `foobar`

Distributed Hierarchical Database



Root DNS Servers

com DNS servers

ie DNS servers

gr DNS servers

TLD Servers

yahoo.com DNS servers

amazon.com DNS servers

tcd.ie DNS servers

ucd.ie DNS servers

ntua.gr DNS servers

auth.gr DNS servers

- Root servers and TLD servers typically do not contain hostname to IP mappings
- They contain mappings for locating authoritative servers

DNS Root Name Servers

- Contacted by local name server that can not resolve name
- Root name server:
  - Contacts authoritative name server if name mapping not known
  - Gets mapping
  - Returns mapping to local name server

TLD and Authoritative Servers

- Top-level domain (TLD) servers: Responsible for com, org, net, edu, etc., and all top-level country domains ie, gr, …
- 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

- 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
  - Reduces lookup latency for commonly searched hostnames

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 Records

DNS: Distributed DB 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 IP address of authoritative name server for this domain

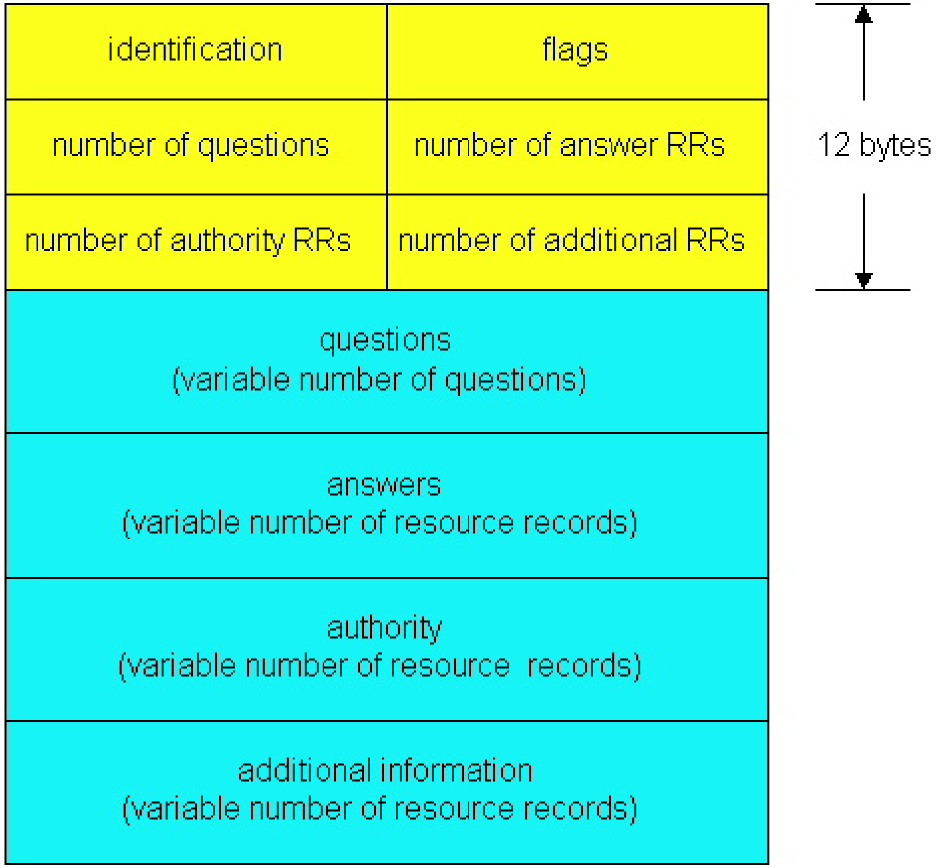
- Type=CNAME
  - **`name`** is alias name for some ``canonical'' (the real) name
    `www.ibm.com` is really
    `servereast.backup2.ibm.com`
  - **`value`** is canonical name
- Type=MX
  - **`value`** is name of mail server associated with **`name`**

DNS Protocol and Messages (1/2)

DNS protocol: *Query* and *reply* messages, both with same *message format*

Message header:

- Identification: 16 bit # for query, reply to query uses same #
- Flags:
  - Query or reply
  - Reply is authoritative



identification

flags

number of questions

number of answer RRs

number of authority RRs

number of additional RRs

questions
(variable number of questions)

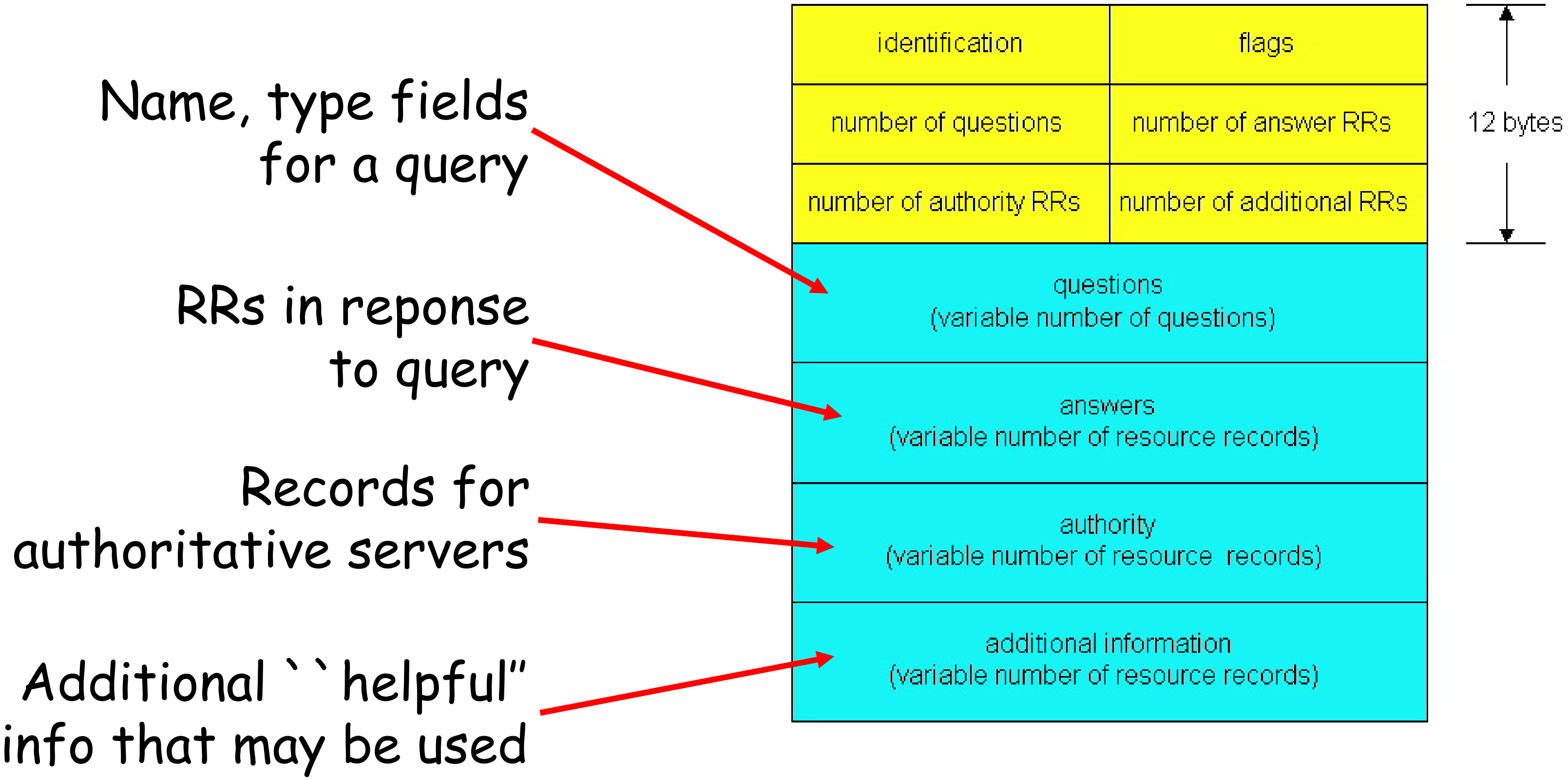
answers
(variable number of resource records)

authority
(variable number of resource records)

additional information
(variable number of resource records)

12 bytes

DNS Protocol and Messages (2/2)



Name, type fields for a query

RRs in reponse to query

Records for authoritative servers

Additional ``helpful” info that may be used

identification	flags
number of questions	number of answer RRs
number of authority RRs	number of additional RRs
questions (variable number of questions)	
answers (variable number of resource records)	
authority (variable number of resource records)	
additional information (variable number of resource records)	

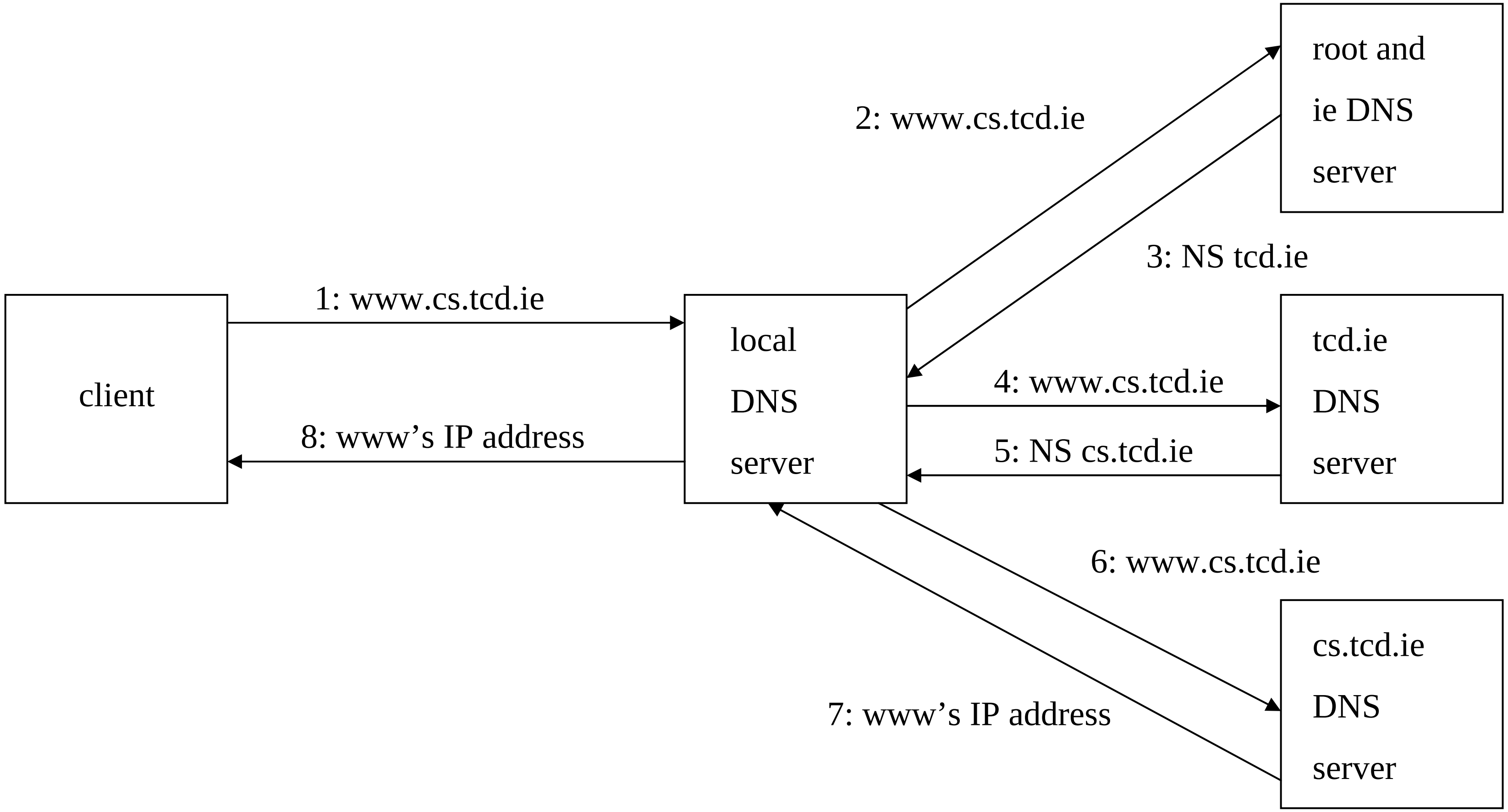
Inserting Records Into DNS

- Example: Just created startup ``Network Fun''
- Register name networkfun.com at a registrar (e.g., Network Solutions, Inc.)
  - 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:

  * (networkfun.com, dns1.networkfun.com, NS)

  * (dns1.networkfun.com, 212.212.212.1, A)
- Put in authoritative server (dns1.networkfun.com) Type A record for www.networkfun.com and Type MX record for networkfun.com
- How do people get the IP address of your web site?

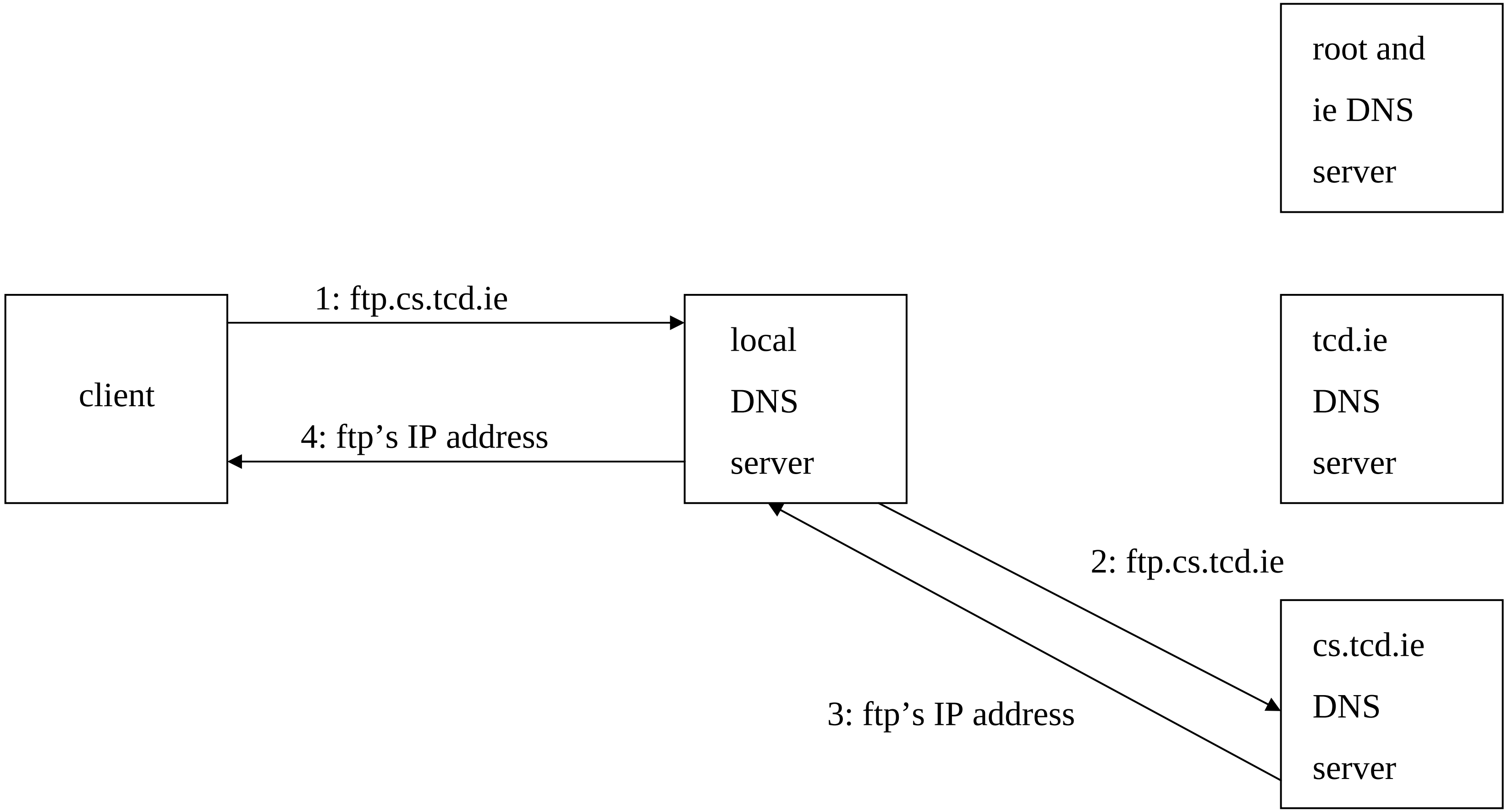
DNS Lookup Example



DNS Caching

- DNS responses are cached:
  - Quick response for repeated translations
  - Other queries may reuse some parts of lookup
    - * NS records for domains
- DNS negative queries are cached
  - Don’t have to repeat past mistakes
  - E.g. misspellings
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed along with every record

Subsequent Lookup Example



Reverse DNS

- Given numeric IP address, find DNS name
- To find 150.10.20.1:

 - Query 1.20.10.150.in-addr.arpa

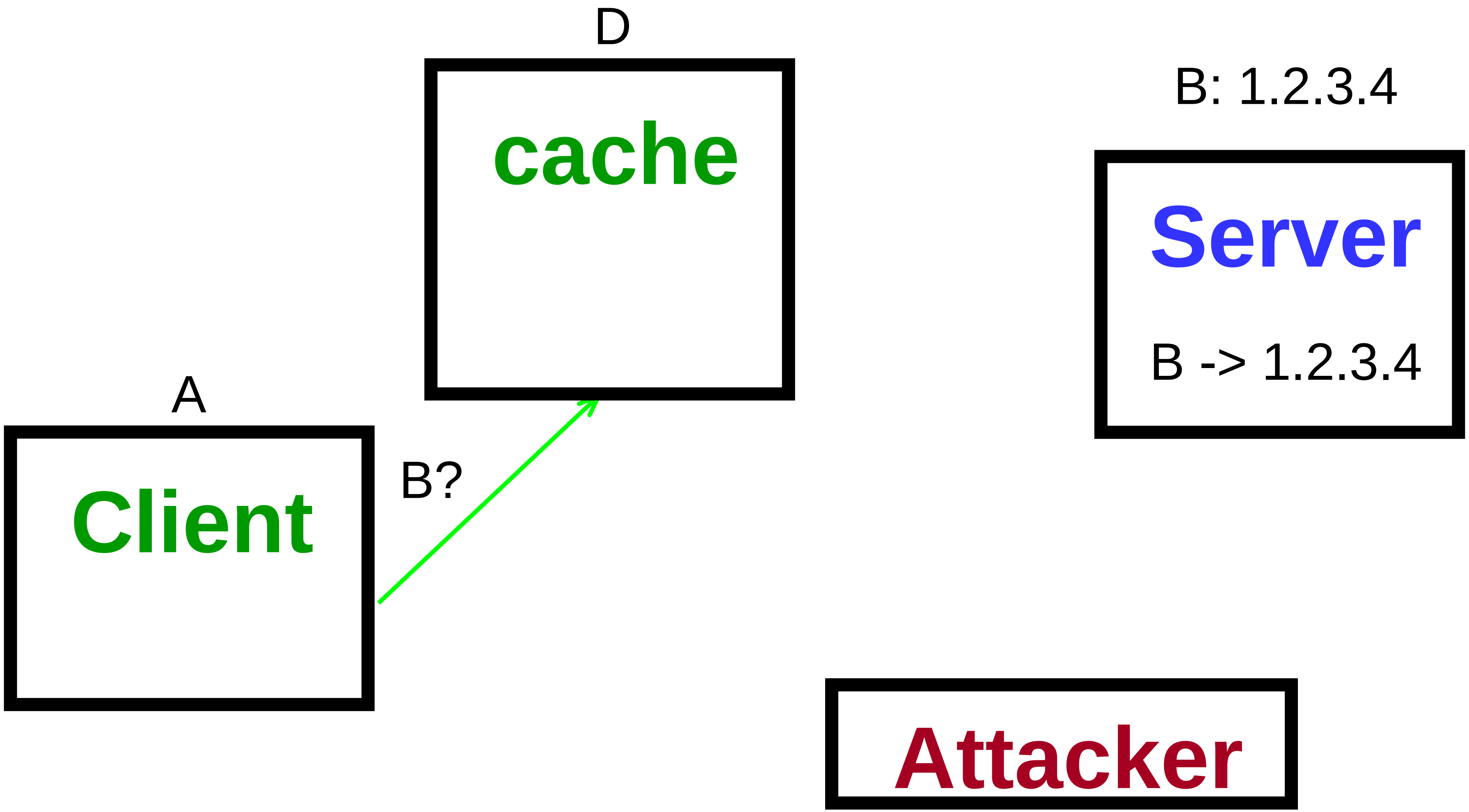
 - Get back server.acme.com

DNS Attacks

- Cache poisoning
- Reverse DNS attack

 - Known as the Bellovin/Mockapetris attack

DNS Lookup: A asks D for B’s IP address



D

cache

B: 1.2.3.4

Server

B -> 1.2.3.4

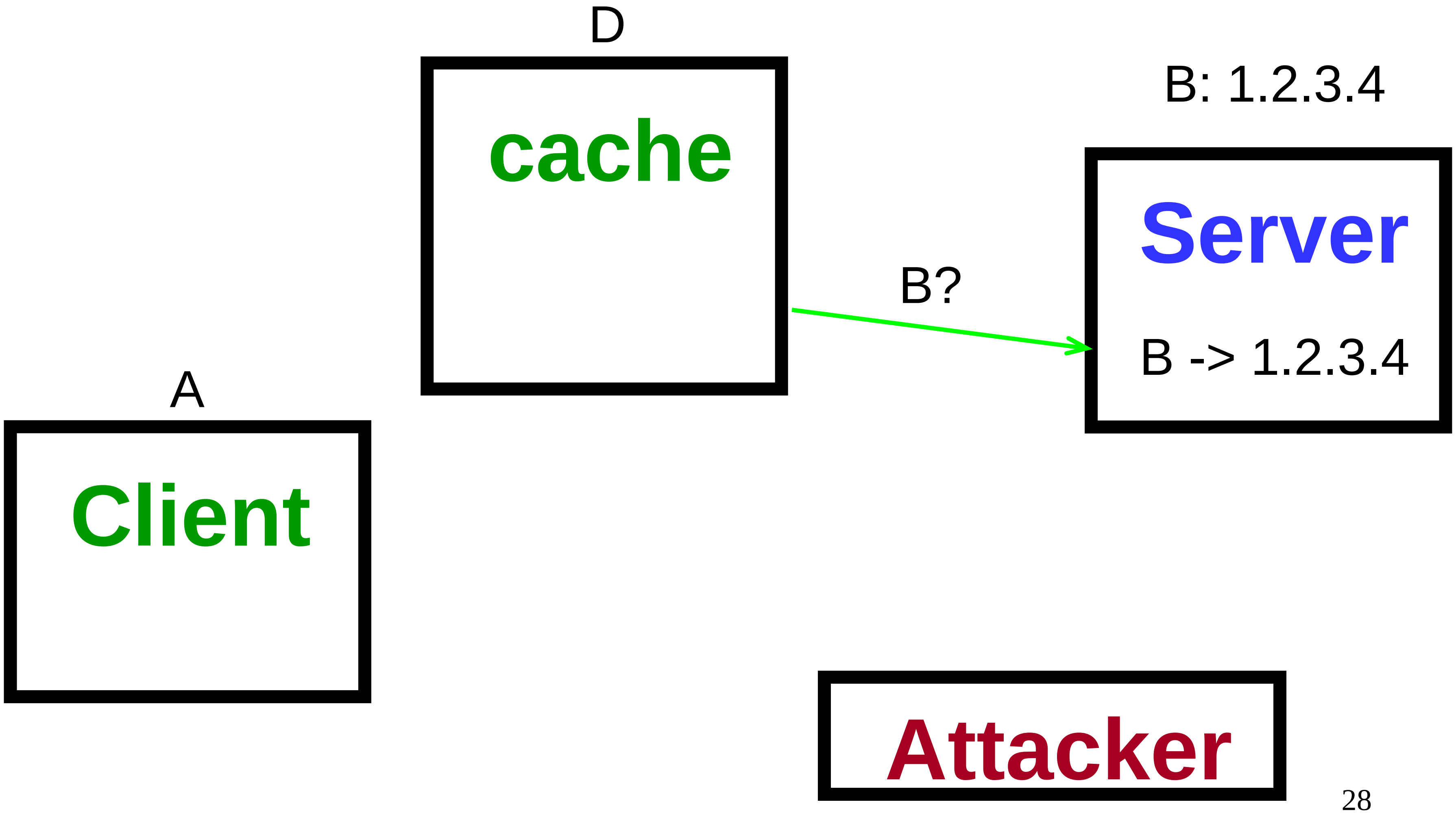
A

Client

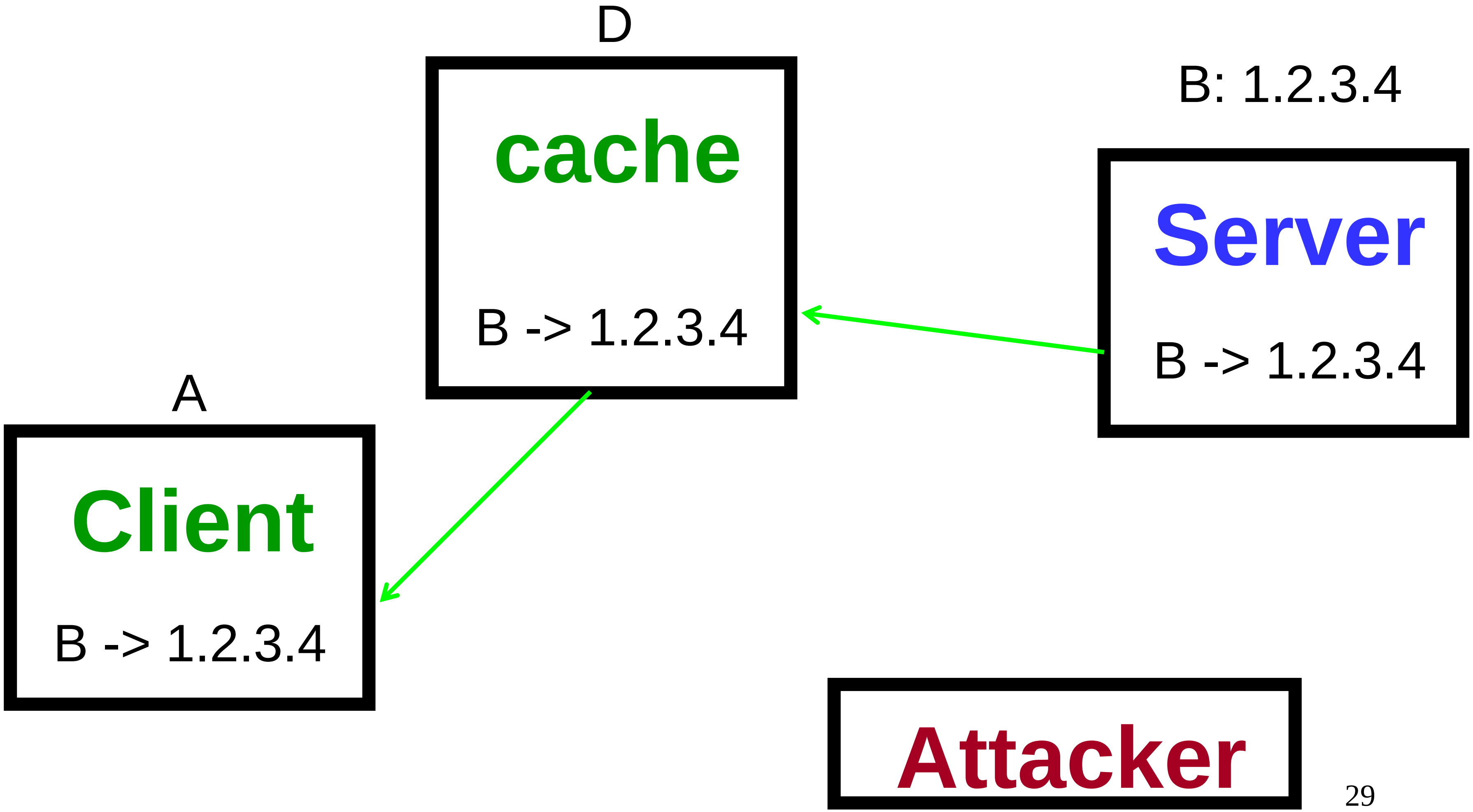
B?

Attacker

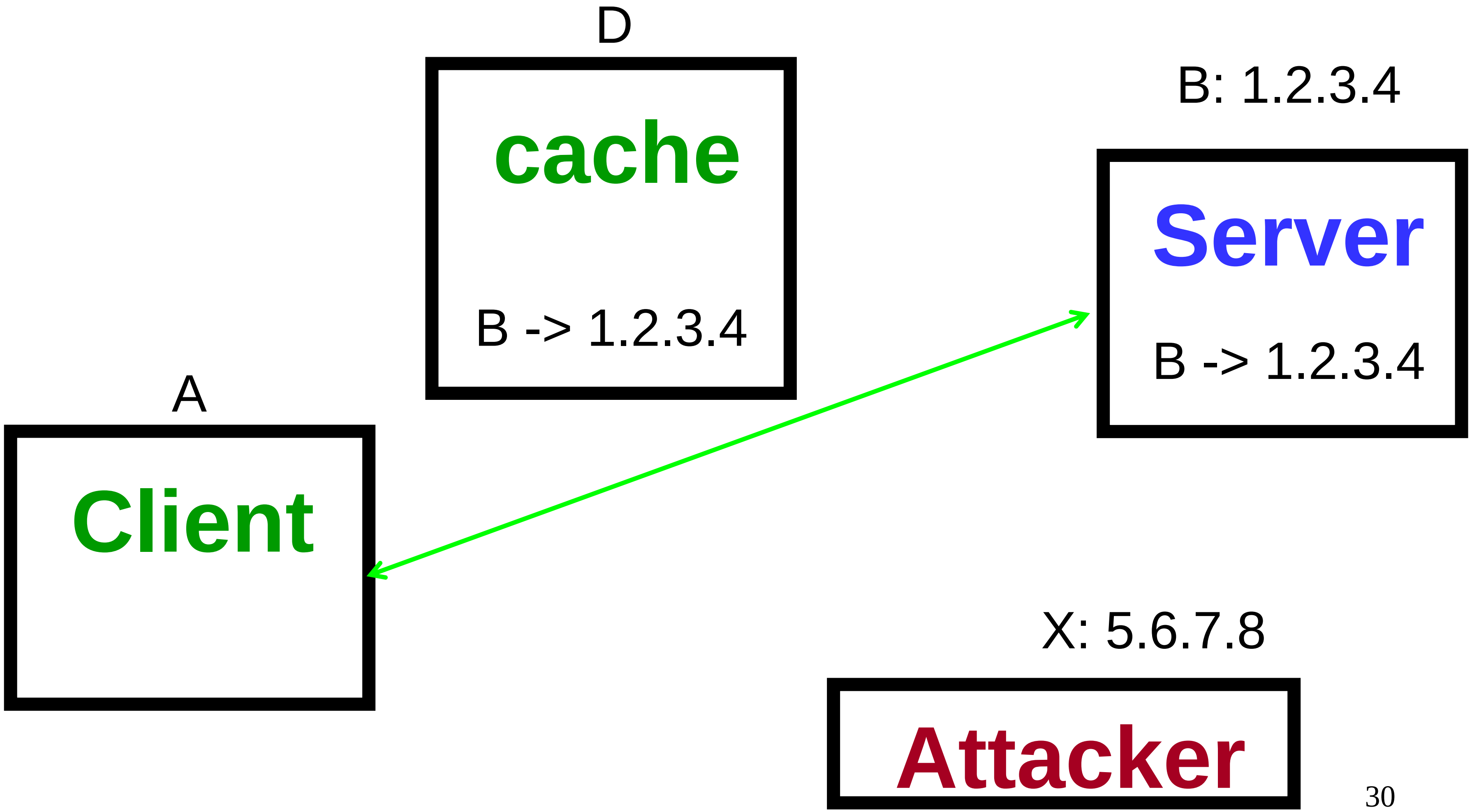
DNS Lookup: D asks B (or someone who knows about B)



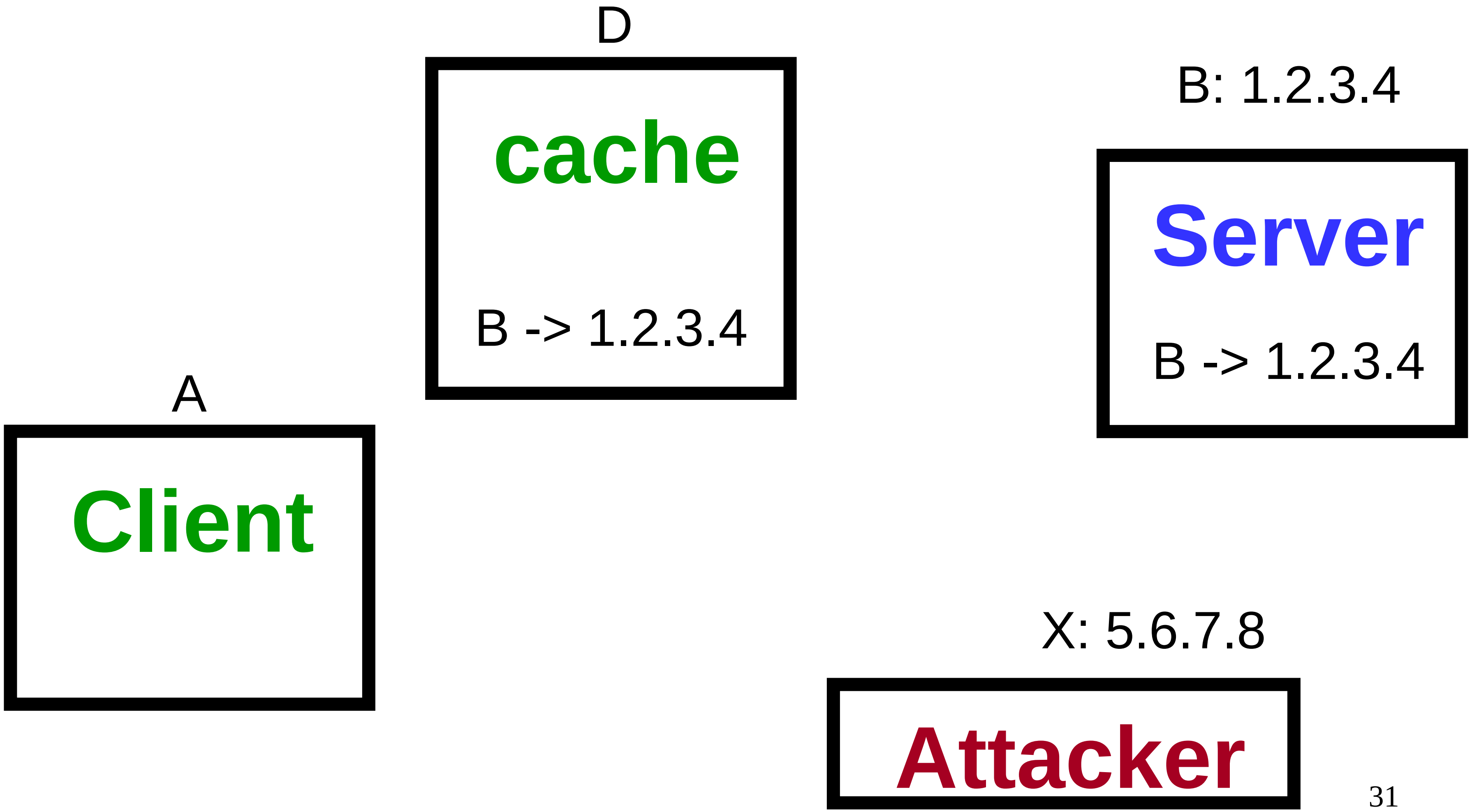
DNS Lookup: B answers, D caches the answer, and tells A



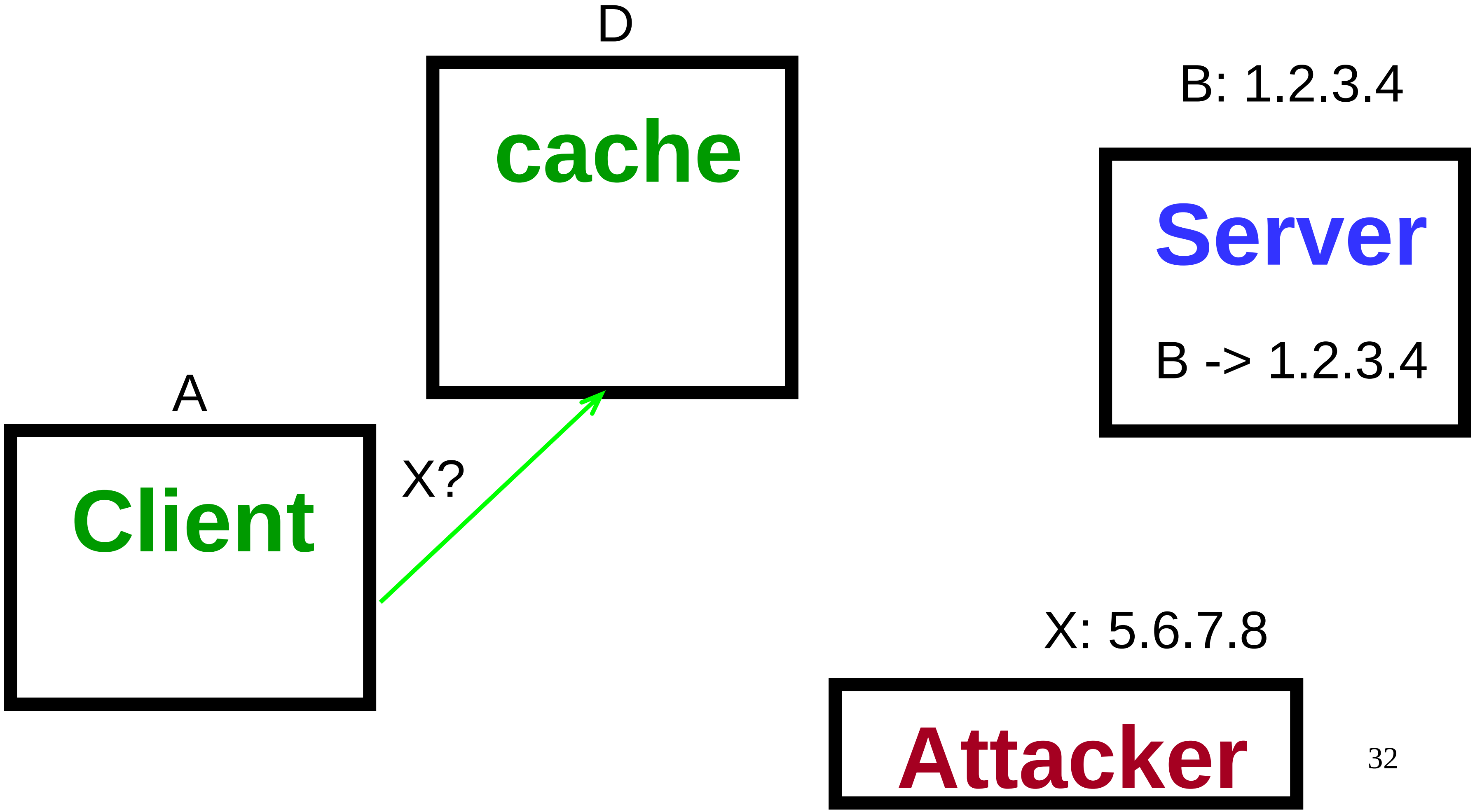
DNS Lookup: A uses the answer



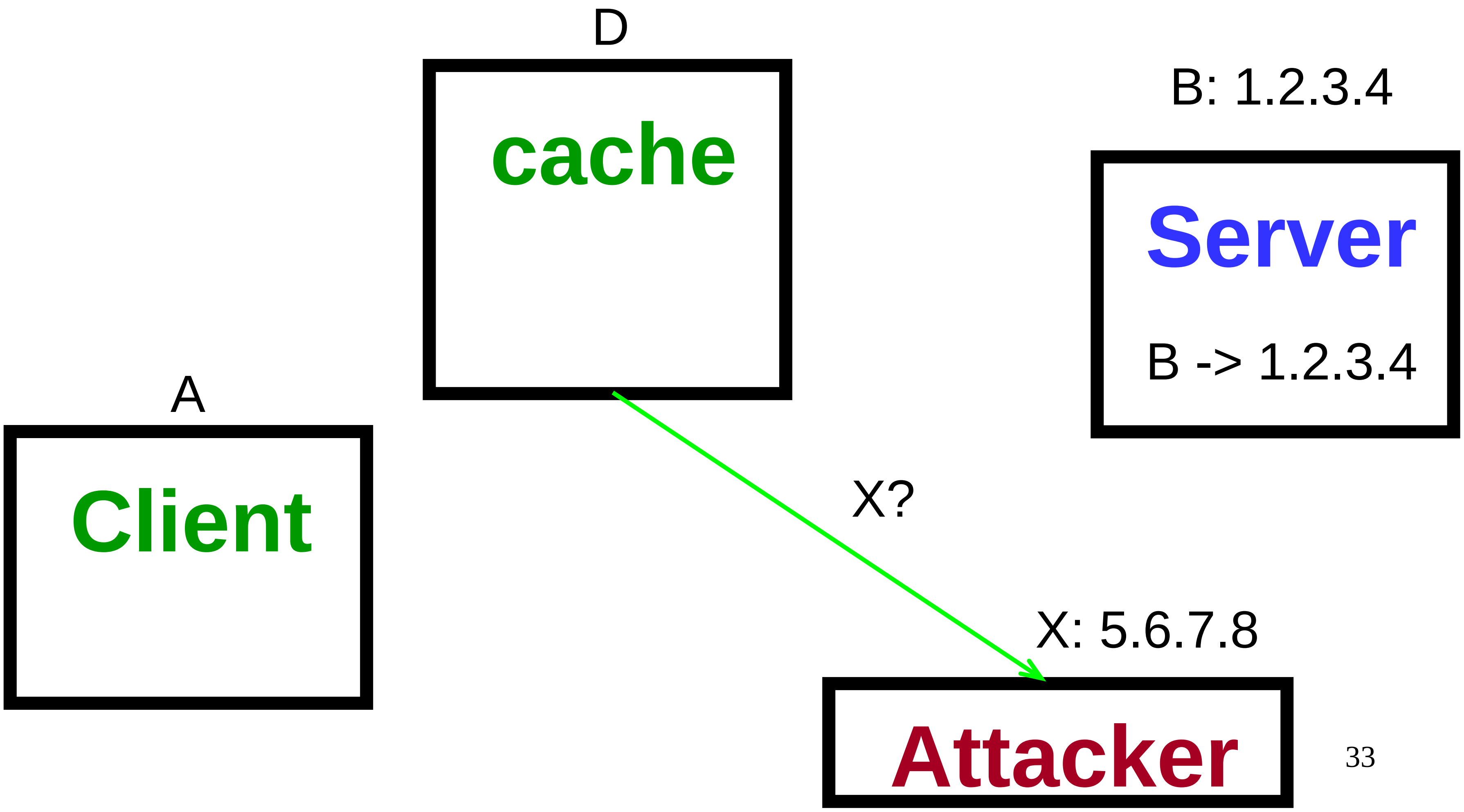
DNS Lookup: D remembers the answer for a given period



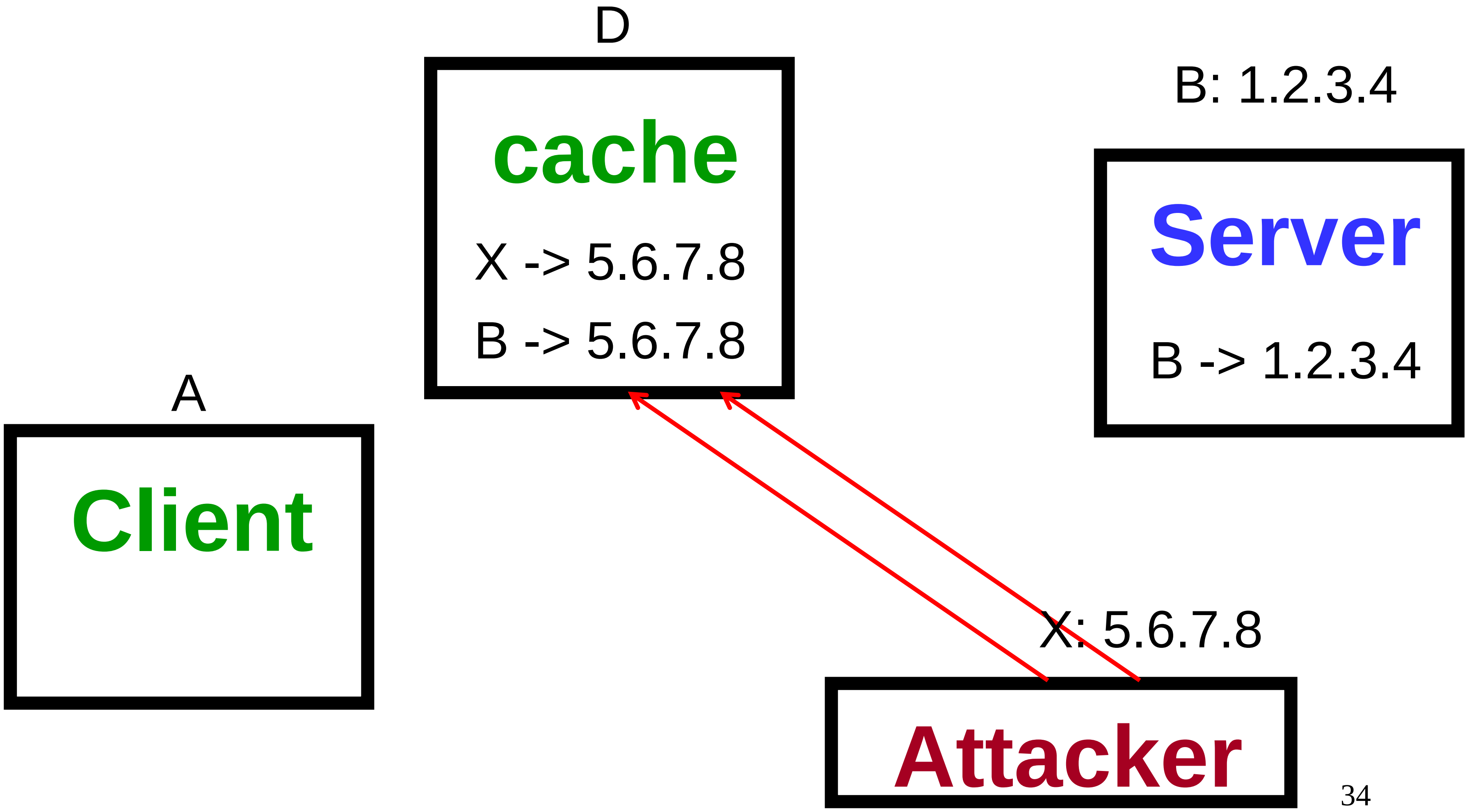
DNS Cache Poisoning Attack: Attacker C arranges for D to ask him a question



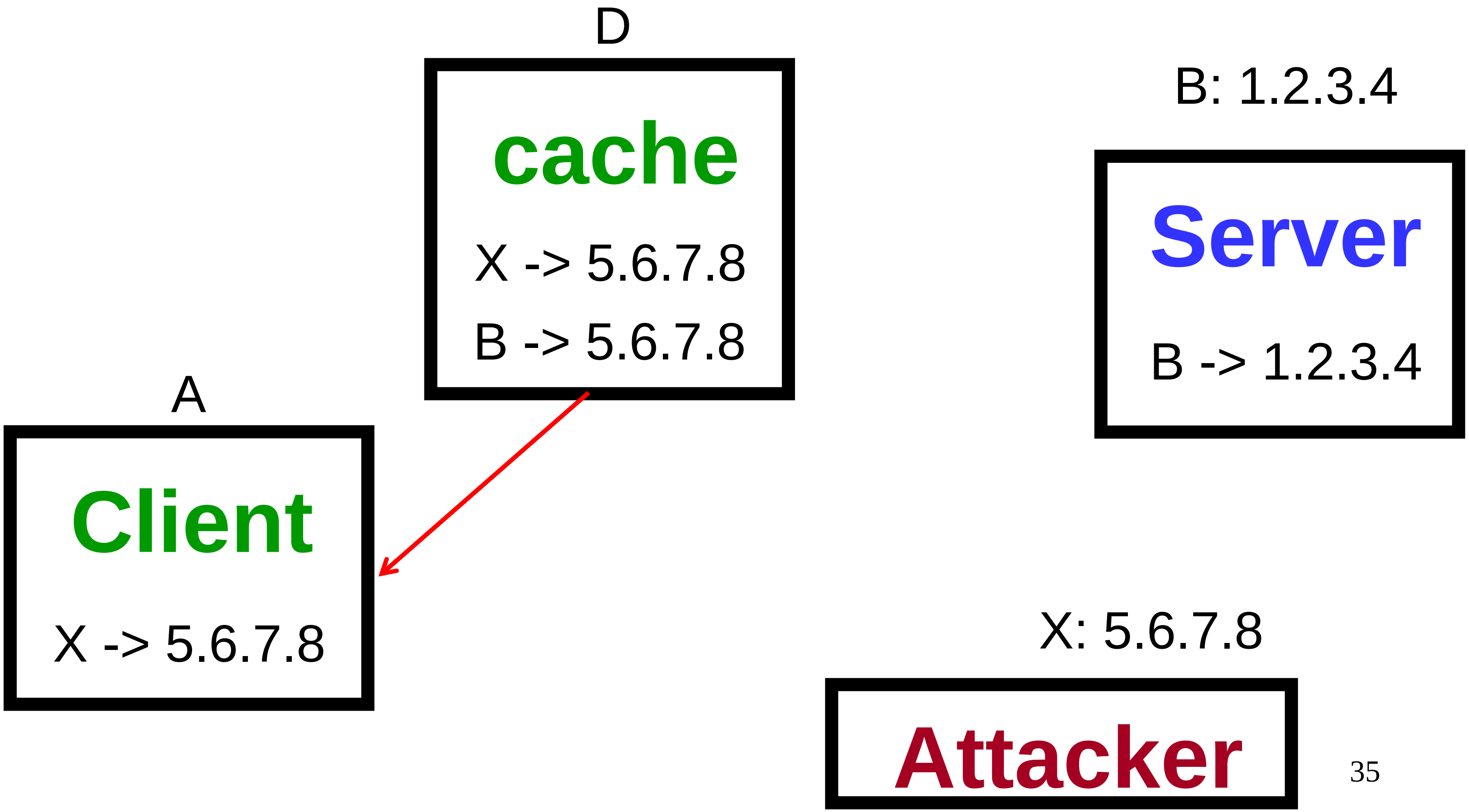
DNS Cache Poisoning Attack: Attacker C arranges for D to ask him a question



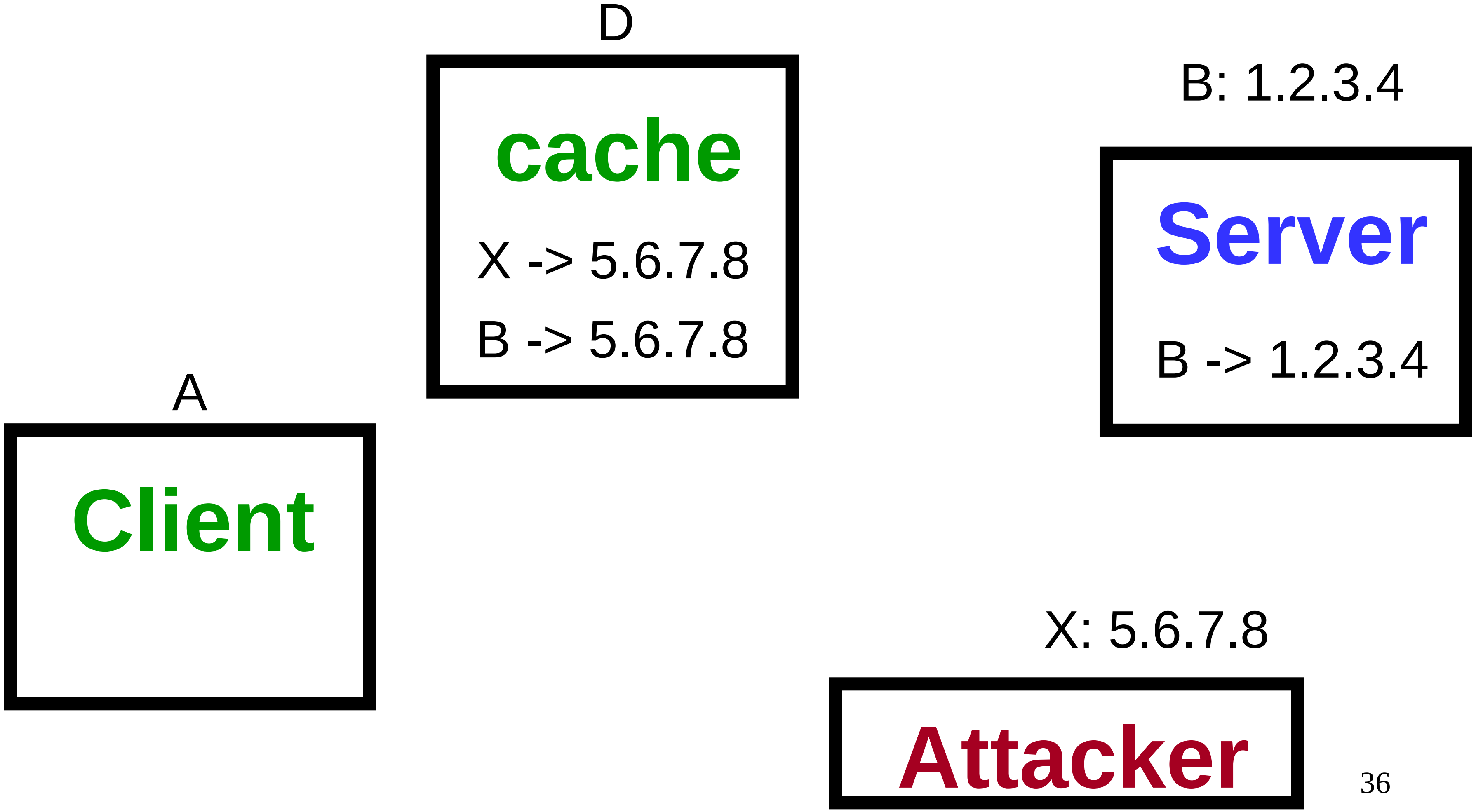
DNS Cache Poisoning Attack: The attacker gives an answer, plus…



DNS Cache Poisoning Attack:
A gets his answer, and uses it



DNS Cache Poisoning Attack: The cache has an extra answer



D

cache

X -> 5.6.7.8

B -> 5.6.7.8

B: 1.2.3.4

Server

B -> 1.2.3.4

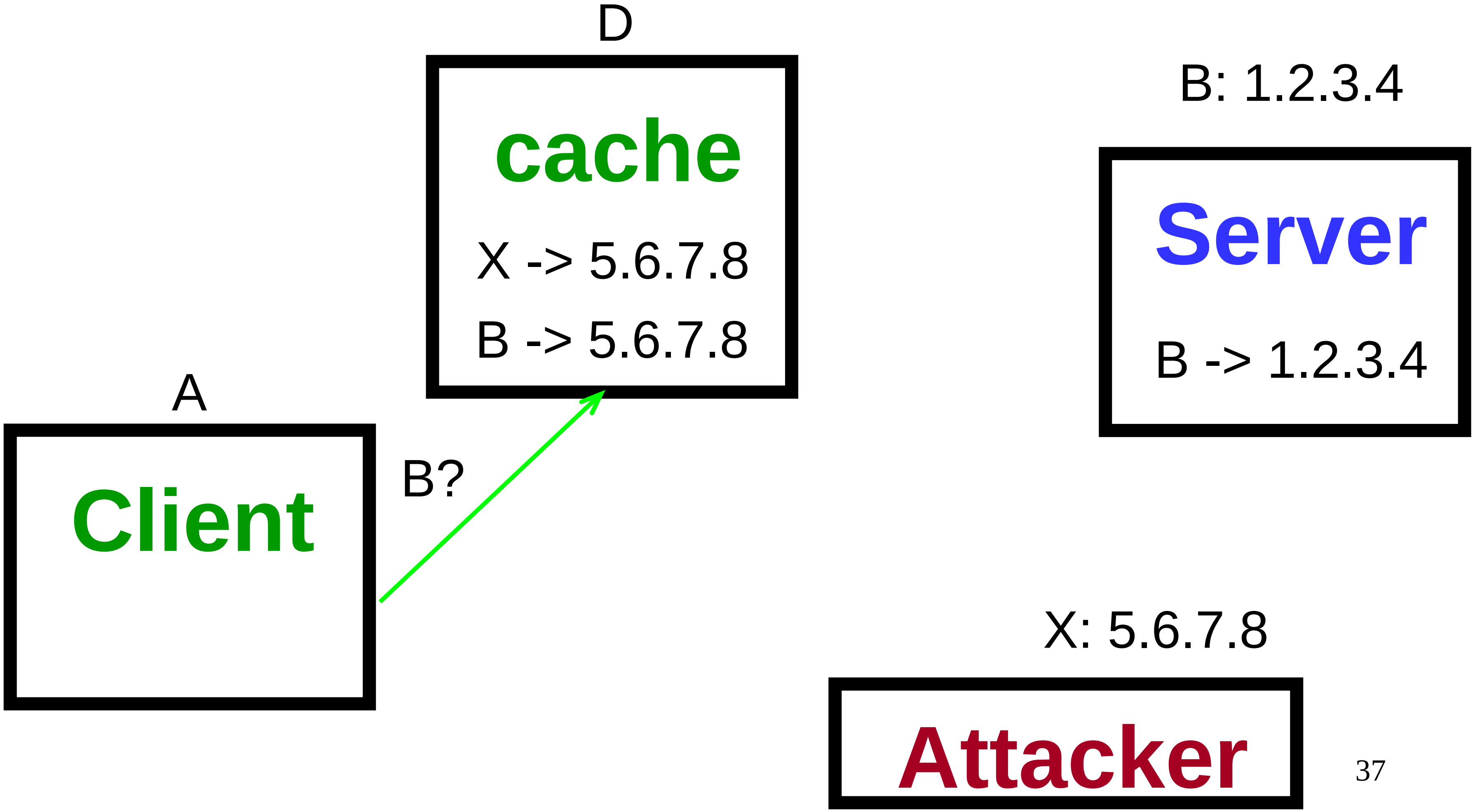
A

Client

X: 5.6.7.8

Attacker

DNS Cache Poisoning Attack: Now A asks for B’s address



D

cache

X -> 5.6.7.8

B -> 5.6.7.8

B: 1.2.3.4

Server

B -> 1.2.3.4

A

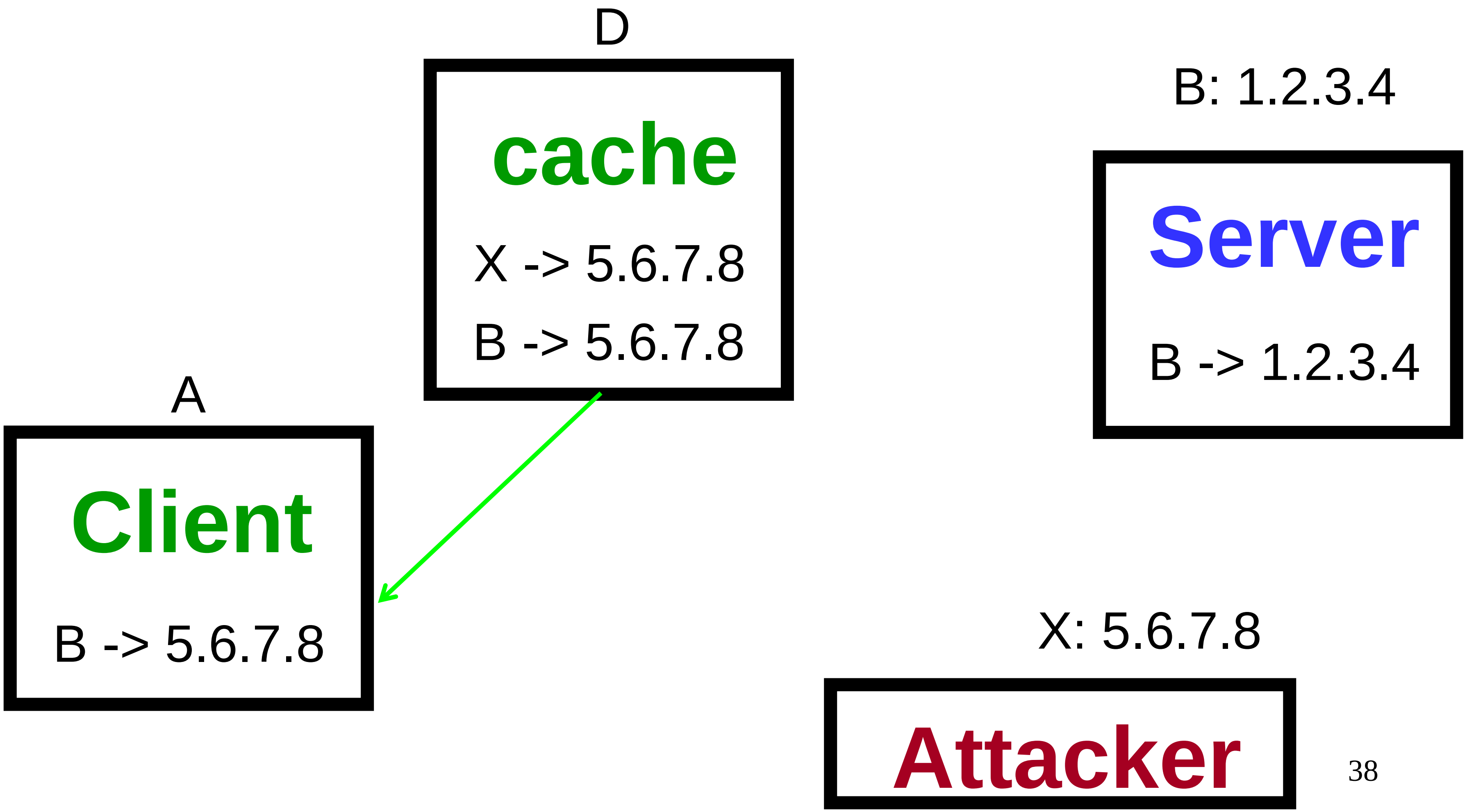
Client

B?

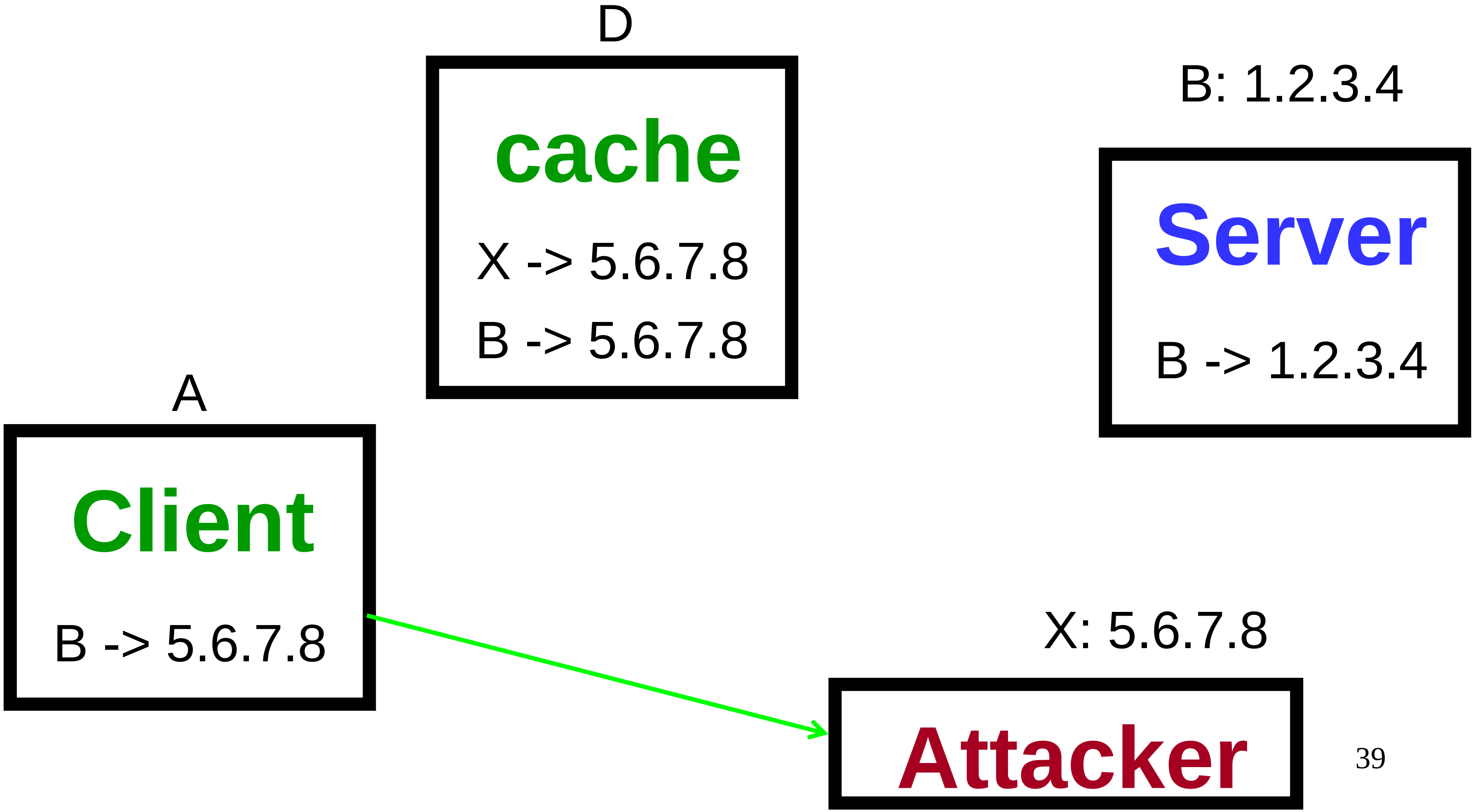
X: 5.6.7.8

Attacker

DNS Cache Poisoning Attack: D ``knows’’ the answer already, and returns it



DNS Cache Poisoning Attack: A uses the answer



DNS Cache Poisoning

- Older versions of bind fall for this
- You can even send an answer without a query, to some implementations!
- DNS responses can be spoofed to
  - What if the query gets two answers: Use the first?!
- DNSsec fixes this

Bellovin/Mockapetris Attack

- Trust relationships use DNS names
  - /etc/hosts.equiv contains ntrg.cs.tcd.ie
- Requests come with numeric IP source address
  - Use reverse DNS to find DNS name
  - Decide access based on /etc/hosts.equiv

Attack

- Gain control of DNS service for domain
- Select target machine in domain
- Find trust relationships

 - SNMP, finger can help find active sessions, etc.
 - Example: Target trusts host1

- Connect:

 - Attempt rlogin from compromised machine
 - Target contacts reverse DNS server with IP addr1
 - Use modified reverse DNS to say addr1 is host1
 - Target allows rlogin

Defense Against This Attack

- Double-check reverse DNS:
  - Modify rlogind, rshd to query DNS server
  - See if DNS name maps to numeric IP address
- Authenticate entries in DNS tables:
  - DNSsec
  - Requires some form of PKI…