CSC7970 – NEXT-GENERATION NETWORKING

DOMAIN NAME SYSTEM (DNS)

Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu

So far

- We talked about
  - Telephony
  - IP
    - Design principles (end-to-end)
    - Why is IP a huge success?
    - How IP networks create an Internet (BGP)
    - Looked at Anycast for tricking the applications
  - So we know how to move packets between two hosts located anywhere
    - But – there are too many hosts and too many services

Domain Name System (DNS)

- Humans are good with names, machines are good with numbers
- Users and applications understand names, the network understands IP addresses –
  - 208.113.161.95 – not that interesting
  - cat-bounce.com
- We need a name to address translation system
  - Domain Name System (DNS)

Before DNS? Flat files

- Stanford was maintaining a directory file (HOSTS.txt)
  - Contained mappings of hostnames to IP addresses
  - Download HOSTS.txt on a regular basis to learn up-to-date mappings
  - Still exists – and still overrides DNS queries
- How to add your computer to the directory?
  - Call Stanford, ask then to add your mapping to the file
- In the 80s, more entities joined the Internet research
  - Maintaining a centralized, monolithic file did not scale

DNS

Paul Mockapetris
IEEE/ACM Fellow
Internet Hall of Fame 2012

is a hierarchical and decentralized **naming system** for hosts, services and other resources

From a systems perspective

- You need various components
  - Network protocols
  - Databases
  - Distributed system concepts

Network Protocols

- DNS translates (“resolves”) domain names to IP addresses

- 3 main components:
  - Hierarchical ”namespace”
  - Domain Name Servers that keeps the name ↔ IP mapping
  - Resolvers (aka clients) that query the servers

Databases

- Design and implementation of the database
  - How to efficiently store entries (value, key pairs)?
  - How to efficiently lookup/add/delete entries in the database?

A Distributed System

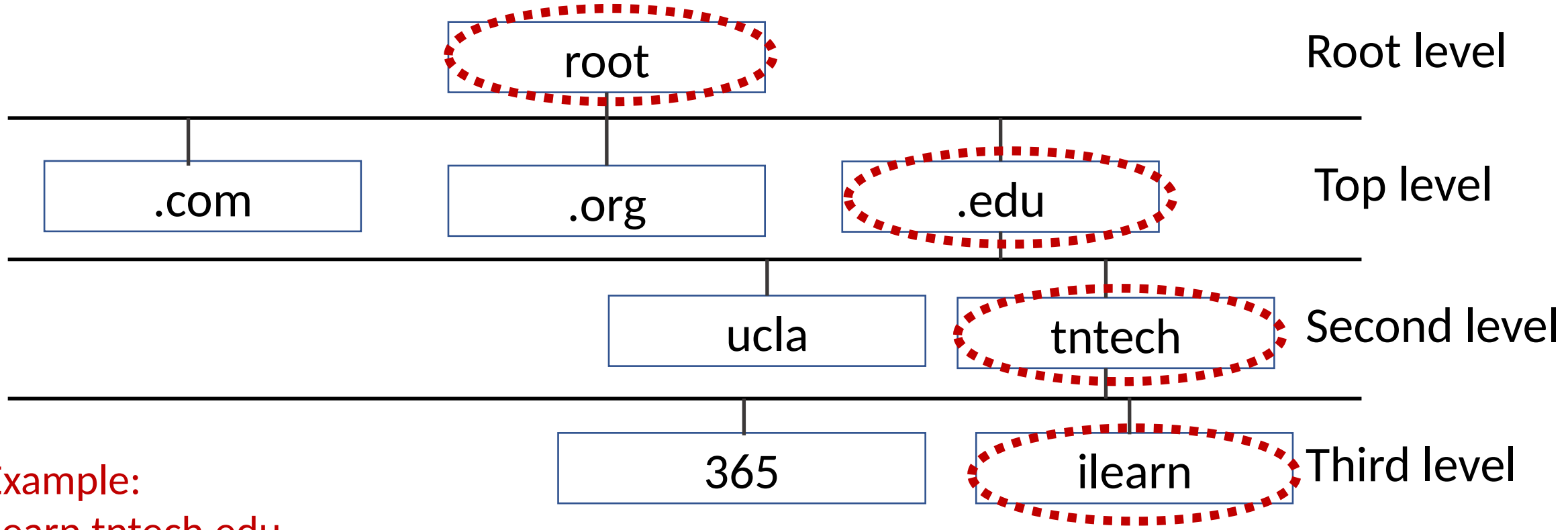
- Globally distributed database
  - How to synchronize different instances of the database?
  - How to deal with failures?
  - How to scale the database deployment all over the world?
  - How to scale the database lookup mechanism?
  - How to do load balancing?

How Does DNS Provide These Functions?

- A system that translates URLs to IP addresses
  - Clients query DNS and DNS returns the IP address of a URL
- Fault Tolerance
  - Distributed design, Anycast
- Scalability
  - Hierarchical design
- Performance/Overhead
  - Caching at various levels
- Security
  - What security?
  - DNSSEC: Security extensions for DNS (added later..)
    - Operators aren't eager to deploy it

1 - The Namespace Component

- The namespace determines the structure of the database
  - Hierarchical tree structure



Example:
ilearn.tntech.edu

Namespace Delegation

- DNS administrators create sub-domains to group hosts based on geography, affiliations, etc.
  - ITS decides the names inside tntech
  - This delegation process creates “zones” of responsibility

- Delegation: good or bad?

Domains and Subdomains

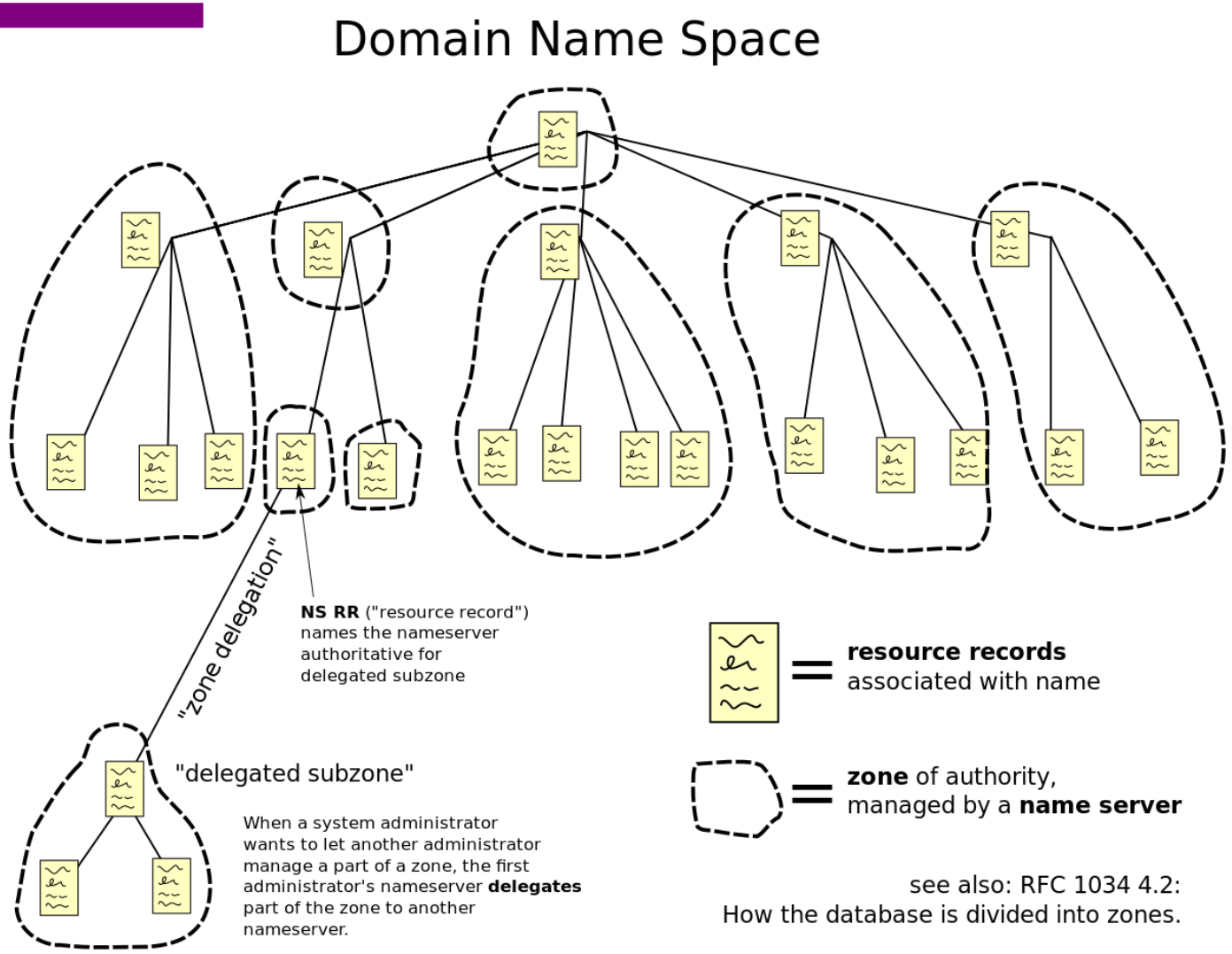
- Domain: sequence of names from a node in the namespace tree to the root
  - Names are separated by dots
  - Example: tntech.edu

- Sub-domain: a domain is a sub-domain of another if its domain name ends in the other's domain name
  - ilearn.tntech.edu is a sub-domain of tntech.edu
  - tntech.edu is a sub-domain of edu
  - edu is a sub-domain of _ ?

2. Domain Name Servers

- Name servers store information about the namespace
- 2 types of servers:
  - Authoritative servers: hold the actual DNS records (mappings of names to addresses)
    - Master: Records are edited there
    - Slave: Records are replicated from master
  - Caching servers: stores data obtained from an authoritative server
    - Records are cached based on lifetime
- Why designate an authoritative server?
- Why we need caching?
- Any problems that caching can create?

Zoning



https://en.wikipedia.org/wiki/Domain_Name_System

A few numbers..

- 13 root DNS servers
- More than 1000 Top Level Domains (TLDs)
  - .com, .org, .edu, etc.
- The most expensive domain name?
  - **CarInsurance.com — $49.7 million[1]**
  - **LasVegas.com – 90 million[2]**

[1] https://www.godaddy.com/garage/the-top-20-most-expensive-domain-names/

[2] https://en.wikipedia.org/wiki/List_of_most_expensive_domain_names

Resolvers

- Perform name resolution by querying DNS servers
  - Which servers to query first?

- If you have things cached, try that first
- If not, start resolving

- Starting point: Root servers
  - Root servers will provide the a list of TLD servers
  - TLD server will a list of second level DNS server
  - and so on

Example of Operation

Local Name Server

“What’s the address of www.tntech.edu?”

Client

Example of Operation

Local Name Server

"What's the address of www.tntech.edu?"

Root Name Server

Client

Example of Operation

Local Name Server

Root Name Server

"No idea. Here is a list of name servers for .edu. Go ask one of them"

Client

Example of Operation

Local Name Server

"What's the address of www.tntech.edu?"

Root Name Server

.edu Name Server

Client

21

Example of Operation

Local Name Server

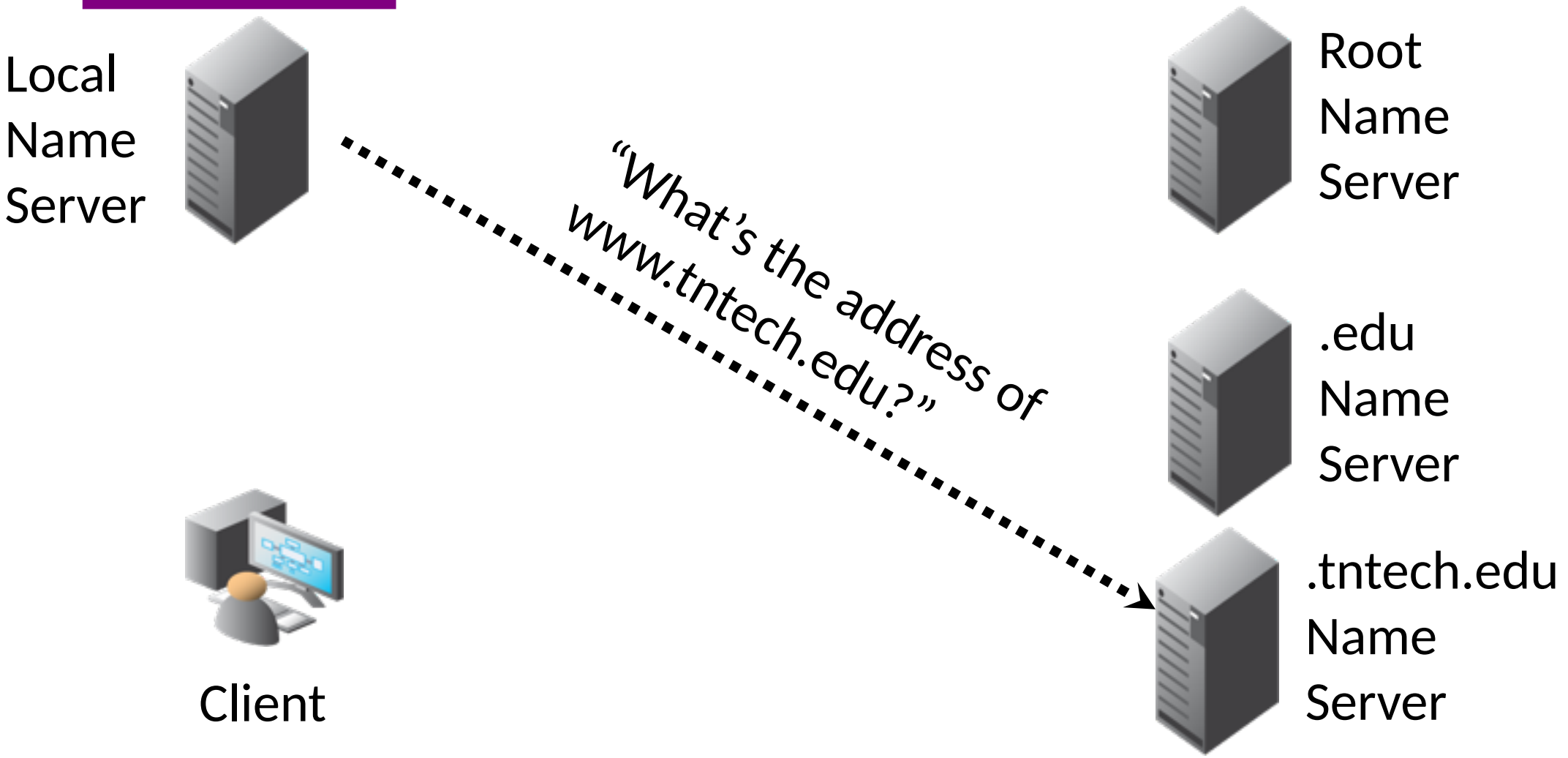
No idea. Here is a list of name servers for .tntech.edu. Go ask one of them"

Root Name Server

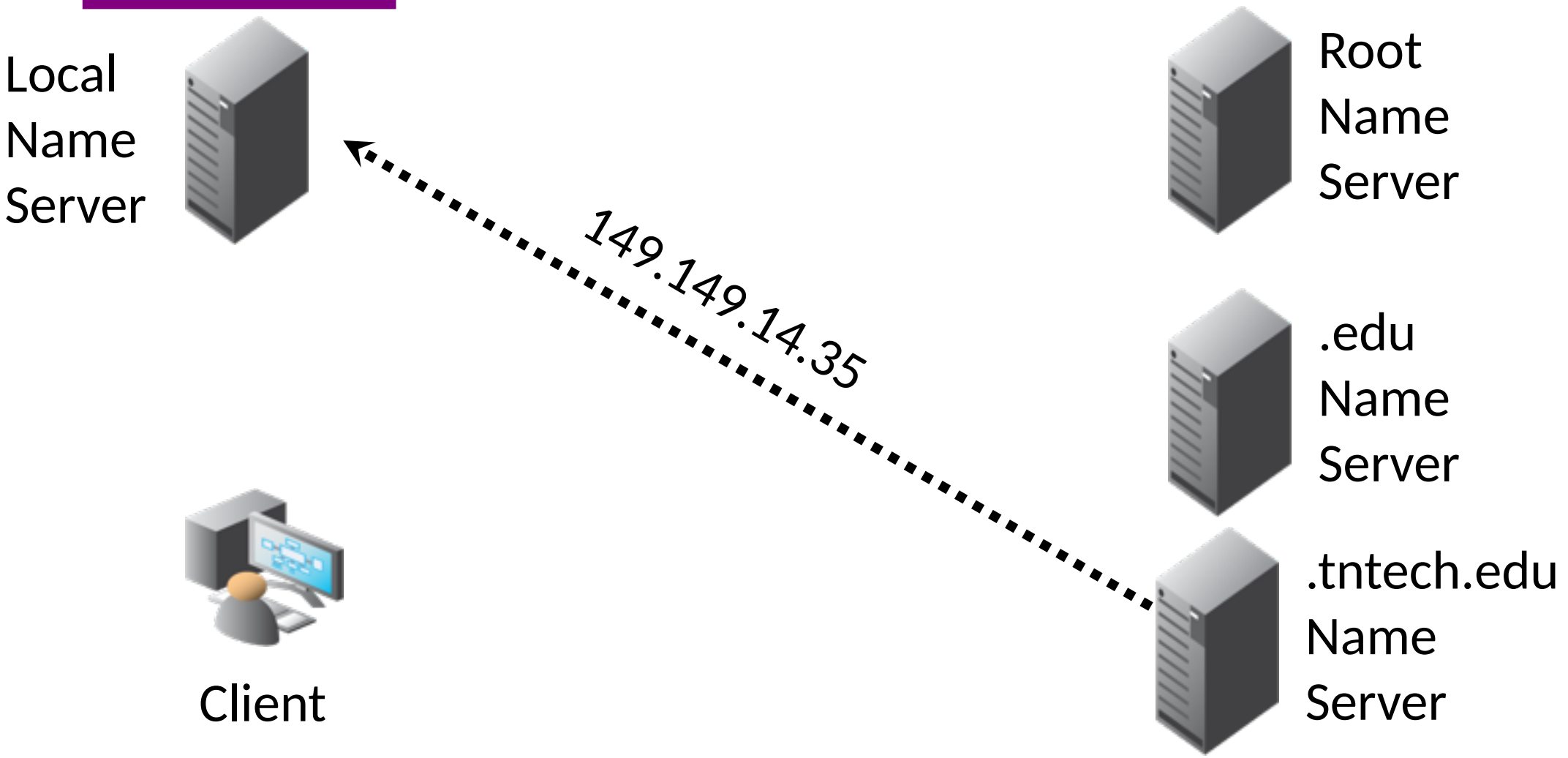
.edu Name Server

Client

Example of Operation



Example of Operation



Example of Operation

Local Name Server

149.149.14.35

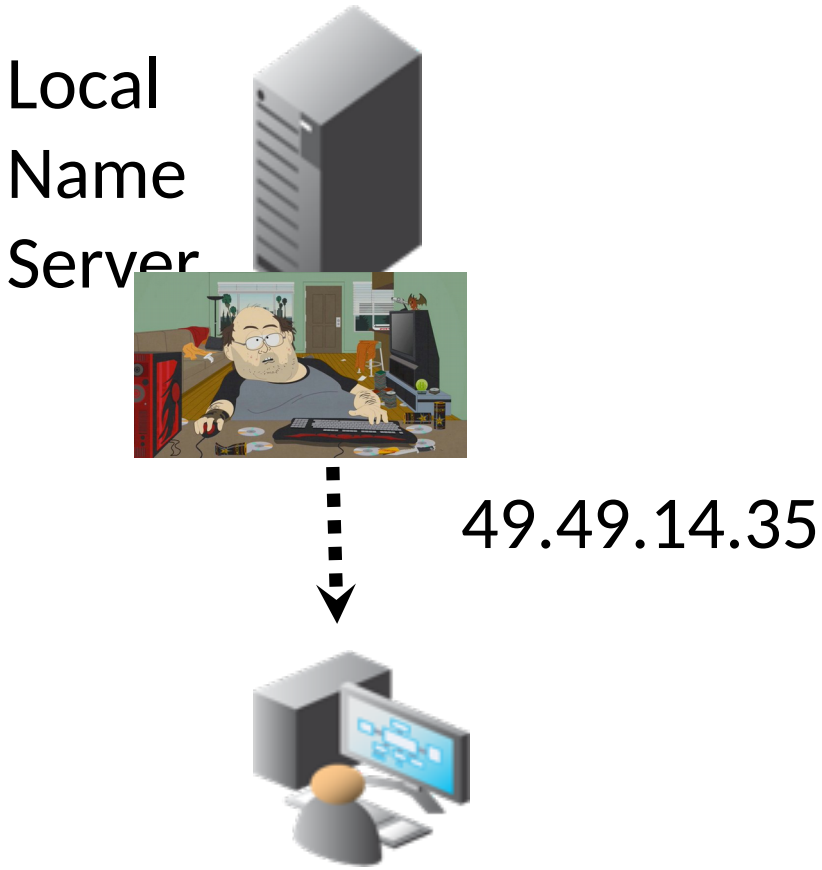
Client

Root Name Server

.edu Name Server

.tntech.edu Name Server

A Simple Attack



Remember – DNS uses unencrypted channels

Other attacks?

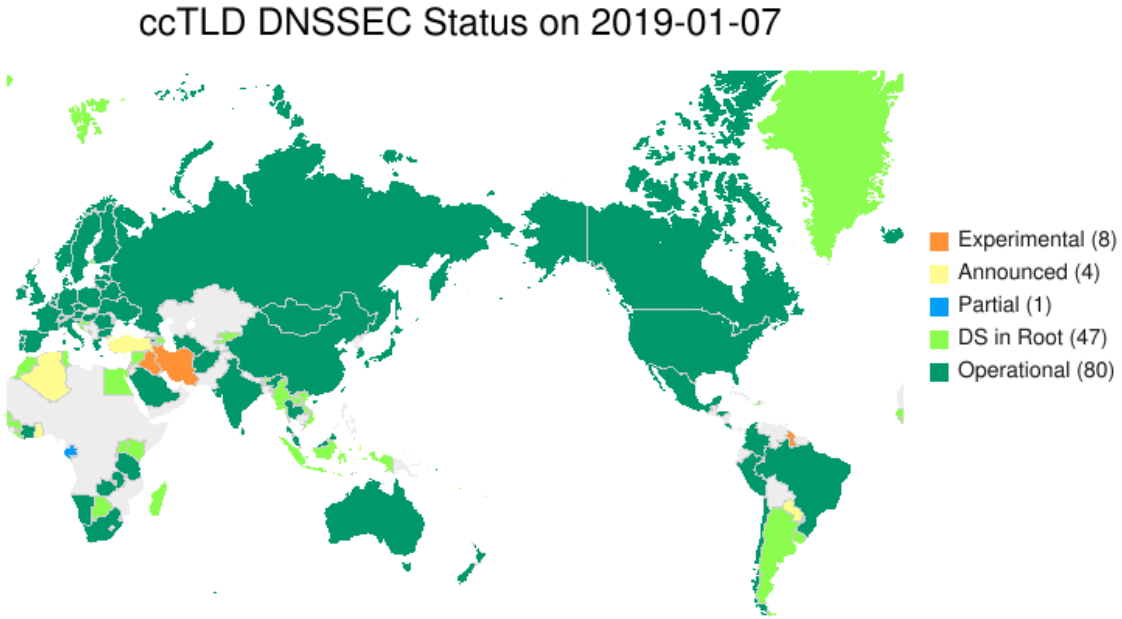
- Cache poisoning
- DNS hijacking
- Man-in-the-middle attacks

DNSSEC: Security Extensions for DNS

DNSSEC: All DNS data is digitally signed

- Origin authentication of data
- Data integrity
- Many different keys

- Complex
  - Operators are not thrilled
  - Introduces new problems while solving others



https://www.internetsociety.org/deploy360/dnssec/maps/

Is depending on DNS bad or good?

- ***Ideal:*** Build network systems with no (minimal) dependencies!

- DNS is ***an amazing distributed system*** from a design and engineering point of view!
  - So many lessons learned!
  - Caching, hierarchical namespace design, name server redundancy

- What can go wrong?
  - Misconfigurations
  - Caching of spoofed DNS records
  - DDoS attacks
  - Censorship

Incidents of Misconfiguration

- Microsoft DNS misconfiguration, 23 January 2001[1]

"The mistaken configuration change limited communication between DNS servers on the Internet and Microsoft's DNS servers. This limited communication caused many of Microsoft's sites to be unreachable (although they were actually still operational) to a large number of customers throughout last night and today."

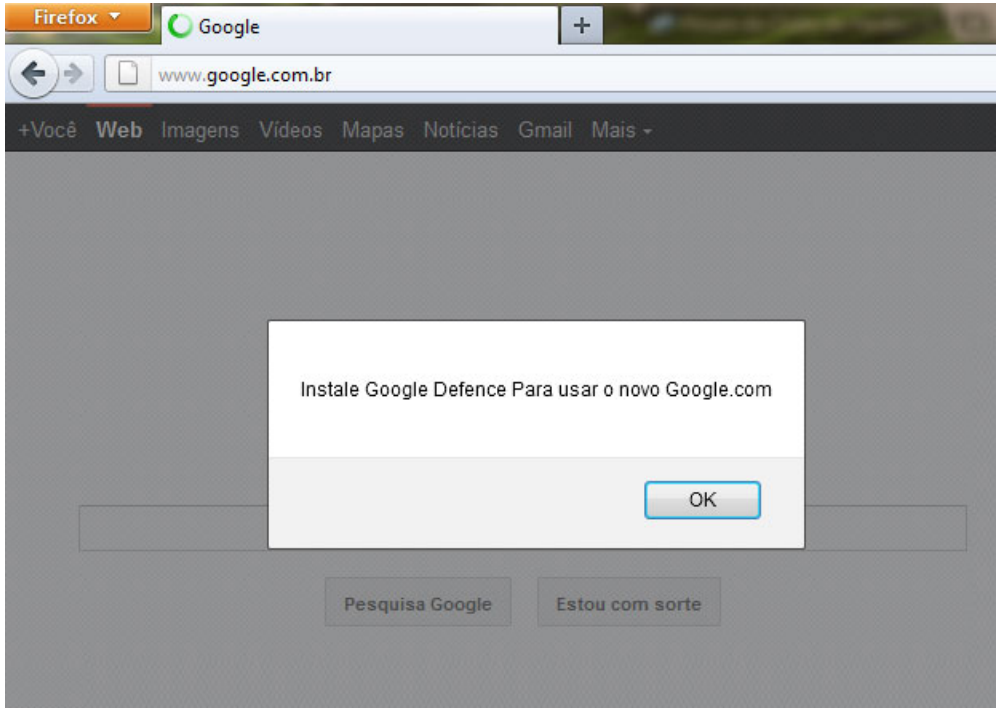
- Configuration errors can make parts of the Internet "disappear"

[1] https://news.microsoft.com/2001/01/24/microsoft-responds-to-dns-issues/

Caching of Spoofed Records

- A spoofed DNS record can redirect users to malicious servers
  - Users think they talk to a legitimate server
  - Server may ask users to download malware and other malicious software

- Incident: DNS spoofing in Brazil (2011)
  - DNS caches of several ISPs got spoofed
  - Users got redirected to a malicious website for google.com.br that looked exactly the same as legitimate
  - Users were asked to download malware
  - Tens of millions of users affected

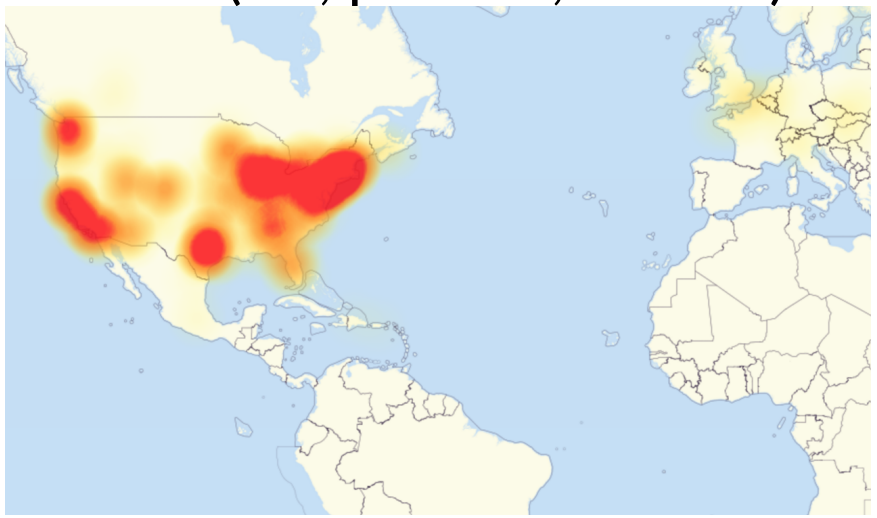


DDoS Attacks

- DDoS attacks can take DNS down
  - Usually performed by groups of compromised devices

- Attack against DYN DNS in 2016
  - Caused by botnet consisting of internet-connected devices (IoT, printers, routers)
  - Bots infected by mirai malware
  - Most websites in NA became inaccessible including Amazon.com, BBC, Starbucks, Reddit.

https://en.wikipedia.org/wiki/2016_Dyn_cyberattack



Censorship through DNS

- Straightforward way to censor specific websites
  - Users perform DNS lookups for a website as usual
  - Censors (typically) control local name servers
  - Censors configure local name servers to return an error to lookup requests for website they do not like

Major Privacy Headache

- Websites typically use HTTPS that are end-to-end
- Costly for your ISP to look into the packet headers (barely scales)
- DNS is the low hanging fruit
  - Your ISP looks at your DNS requests
  - Sells the data (people in Cookeville loves oranges) for targeted ads
- Solution – you can run DNS over https/TLS
- You can run a VPN (now your VPN provider sells the data)
- Other solutions – Tor, DNSCrypt, Quad-1

Summary

- DNS is an incredibly well-designed and engineered distributed system
  - Fault tolerance
  - Scalability
  - Low overhead
  - Ongoing efforts to make it secure and accountable

- It is a dependency that today's Internet cannot live without
  - If DNS does not work, the Internet does not work
  - Security is a problem