IT-ONK

Domain Name System

Christian Fischer Pedersen
Assistant Professor, PhD
cfp@iha.dk

Electrical and Computer Engineering
Aarhus University

April 7, 2013

Outline

PART 1: Domain Name System
PART 2: Berkeley Internet Name Domain (BIND) DNS

Part I

Domain Name System

Outline

Naming in distributed systems

Domain Name System

Outline

Naming in distributed systems

Domain Name System

Distributed system: Loose definitions

Definition (Tanenbaum, 1995 [2])

A distributed system is a collection of independent computers that appear to the users of the system as a single computer.

Definition (Coulouris et al., 2001 [1])

System in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages.

Definition (Wikipedia, 2013)

A distributed system consists of multiple networked computers that interact with each other in order to achieve a common goal.

Identifying entities in a distributed system

What is a name and why is it needed? 

- A **name** in a distributed system is a string of bits or characters that is used to **refer to an entity**

Typical examples of entities

- Hosts
- Printers
- Disks

Common for entities

- Entities can be **operated** on
- E.g. a printer offers an interface for printing, see status, ...

Naming in distributed systems

Three important ways of naming in distributed systems

1. Flat naming
2. Structured naming
3. Attribute-based naming

In this lecture

- We will only look at **structured naming**
- **Domain name system** employs structured naming

Outline

Naming in distributed systems

Domain Name System

Defining DNS

Both are correct

- Domain Name System
- Domain Name Service

Definition (DNS)

A network service that translates names into IP addresses.

The three overall naming schemes in distributed systems

1. Flat naming
2. Structured naming: **DNS employs this**
3. Attribute-based naming

Example of using DNS

Say you want to browse to au.dk

1. For your computer to connect to the au webserver, it must know the IP address of the webserver
2. To get the IP address, your computer asks the DNS
3. The DNS gives back the IP address
4. Your computer can now connect to the webserver using the IP address

In this way, DNS translates a name, e.g. au.dk, into an IP address.

Names

Type 1: Domain names 

- For instance: com, net, org, au, google, yahoo
- For instance: example.net is **two seperate** domain-names

Type 2: Host names

- For instance: cfp-laptop, cfp-desktop, lab-pc, family-pc
- A host name refers to a single computer

Hosts

What are hosts on our network?

- Workstations, printers, routers, web-servers, mail-servers, ...
- Any device on the network that participates in DNS

Common for all hosts

- All hosts have a **host-name** and an **IP address**

IP address

An IPv4 address is a 12 digit number divided in four parts, e.g.

- 192.0.43.10 $\Leftrightarrow$ example.org
- The IP address identifies the host

If two hosts with IP addresses exchange data

- **Very** common to use the TCP/IP protocol

TCP/IP

TCP/IP

- For data exchange between two hosts

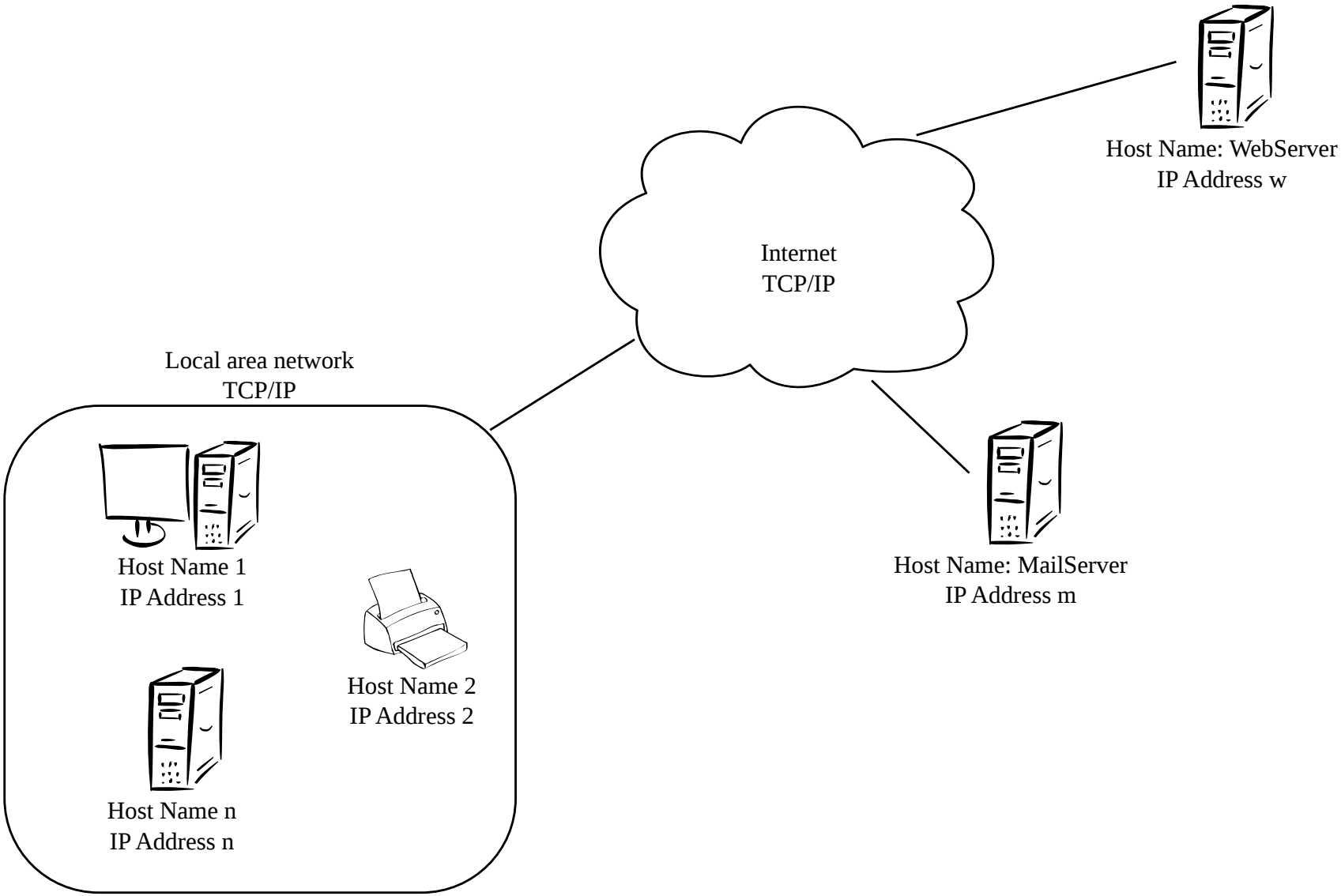
TCP: Transmission Control Protocol

- Establish and maintain error free connection between two hosts

IP: Internet Protocol

- Data packets

DNS and TCP/IP



Inspect your host name and DNS server

When this host, e.g. my pc, connects to example.org

- It asks the DNS for the IP of example.org

Linux

- `hostname`
- `nm-tool | grep DNS`

Windows

- `ipconfig /all`

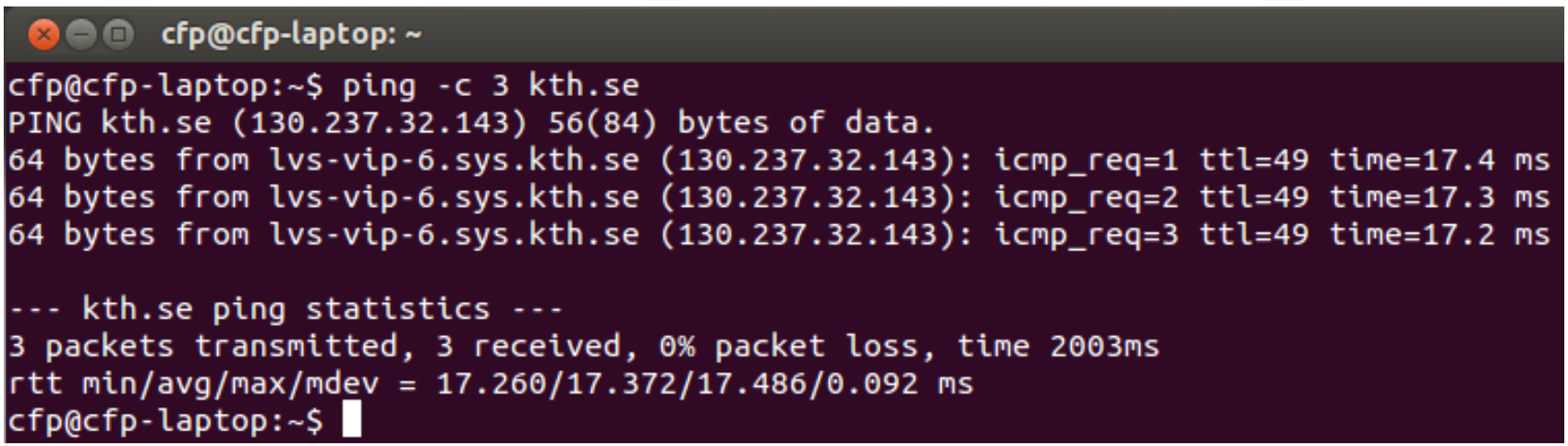
Find an IP of a webserver

Find an IP of a webserver with `ping`, e.g.

- `ping kth.se`

Ping (Windows and Linux)

- Asks the DNS server for IP of webserver
- Sends package to webserver
- Webserver responds to package



Browse with IP or name

Browse by name

- Host asks DNS for IP of webserver
- Host uses IP for connecting to webserver
- Download the webpage

Browse by IP

- Type in the IP address of webserver
- Connects directly to webserver
- Download the webpage

Internet service provider (ISP)

ISPs typically provide to subscribers, e.g.

- Connection to internet
- An IP address
- DNS servers

Timeline of DNS

ARPAnet (1969)

- Start of the internet
- Stanford, UCLA, Uni. of Santa Barber, Uni. of Utah

Host Lookup Table (1971-1972)

- Hosts.txt file by Peggy Karp
- The file must reside on every single host
- Hard to update simultaneously on many hosts

1st Domain Name System (1985)

Host Lookup Table

Windows host file

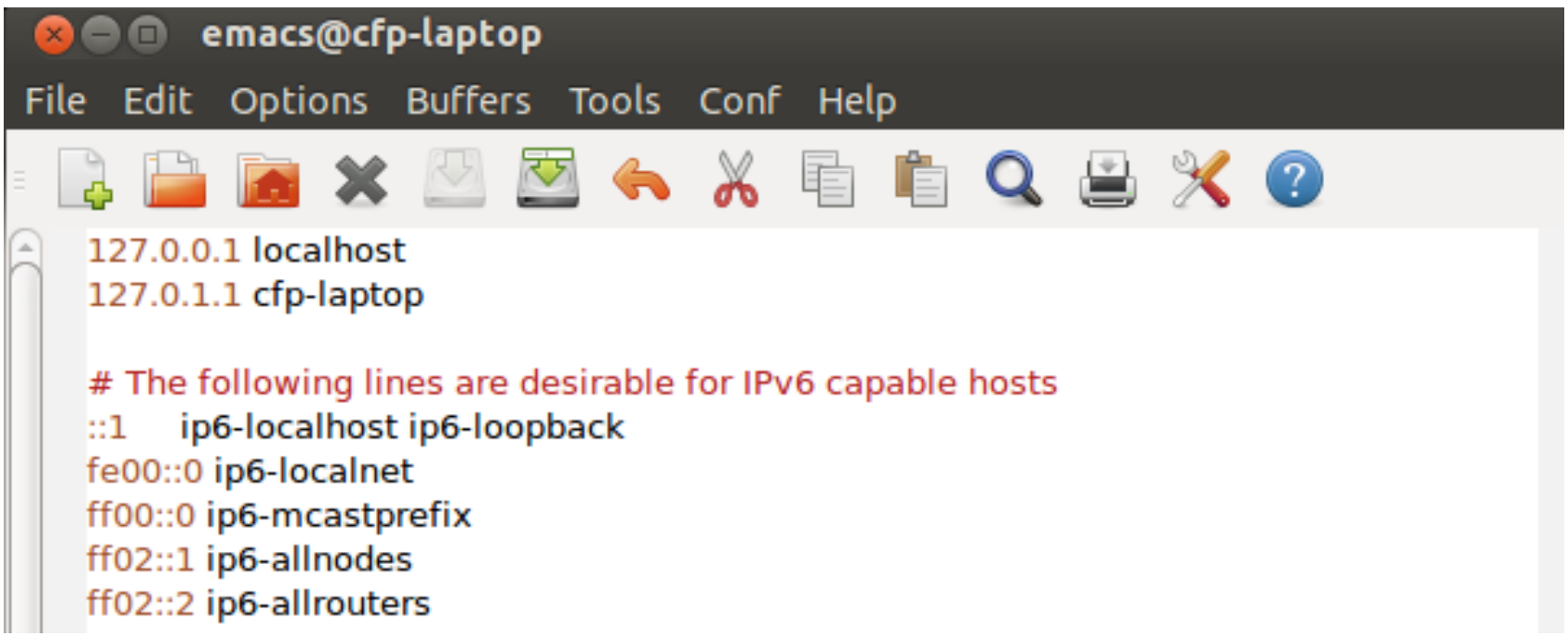
- `c:\windows\system32\drivers\etc\hosts`

Linux host file

- `/ect/hosts`

Example

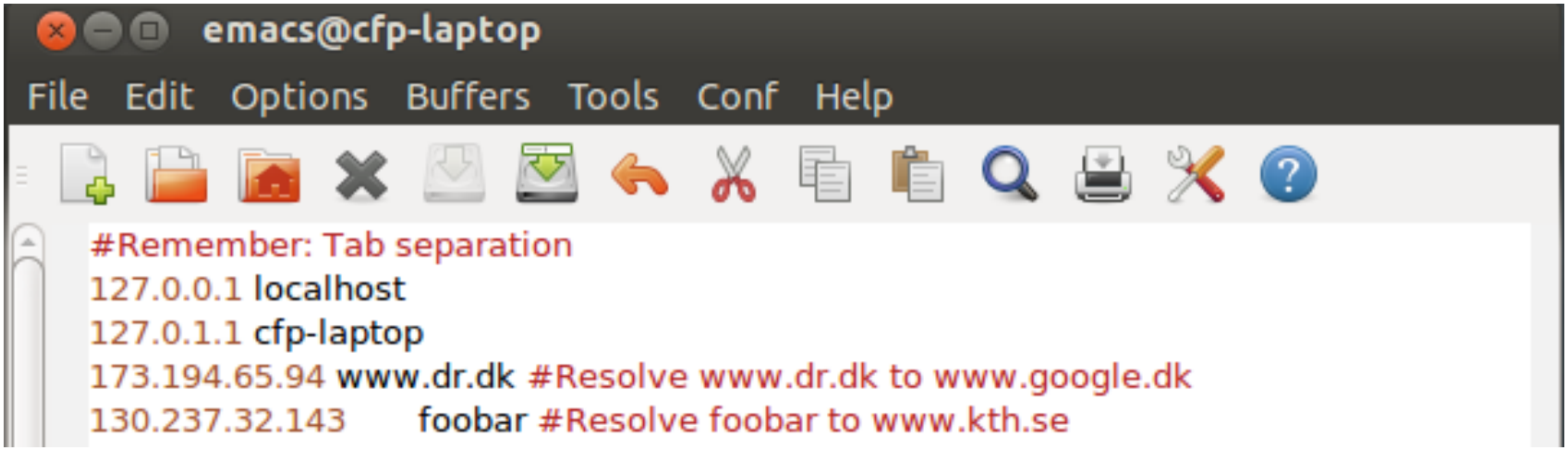
- `sudo emacs /etc/hosts/`



Host Lookup Table: Editing

Example

- `sudo emacs /etc/hosts/`



Thus

1. Resolves via host file
2. Resolves via DNS server

Some viruses

- **Changes** the `hosts` file and **redirects** to shady webserver

How does DNS work?

DNS is a

- Distributed database of names and IP addresses
- Distributed among DNS servers
- The DB is made up of Zone Files each with records

- Like a distributed `hosts` file

When a computer requests name resolution

- The DNS server consults it's Zone File for IP address

DNS levels

Root server

- "."

Top level domain servers (TLD), e.g. 

- .com .net .org .dk

Domain servers

- Yahoo, Google, Lego

Hierarchy

- Root servers knows where top level domain servers are
- Top level domain servers knows where the domain servers are
- Like a **parent-child** or a **sub-super** relationship
- E.g.: cfp-laptop.example.org

Fully qualified domain name (FQDN)

FQDN or absolute domain name

- Specifies **unambiguously** an **exact** location in DNS hierarchy
- Specifies **all** domain levels, **including** the TLD
- Ends with a dot as the DNS root domain is unnamed
- However, there is **no** root server on the global internet

Example

- Host name: `cfp-laptop`
- Parent domain name: `example.org.`
- FQDN: `cfp-laptop.example.org.`

There may exist **many** hosts named `cfp-laptop`

- But only **one** `cfp-laptop.example.org.`

Root nameservers

There are

- 13 authoritative root name servers
- They contain the DNS records for root name lookups
- Each server knows the IP addresses of TLD DNS servers, e.g. .dk, .uk, and .com

Although indicated with a dot, i.e. “.”

- There is **no single** root server on the internet
- Nothing follows the “.” as the DNS root domain is **unnamed**

Labels vs. host names

Connecting to a **single** webserver, i.e.

- **One** webserver and **one** IP address
- A **host name** points to a **single** computer

"Connecting" to a **group** of webservers, e.g.

- www.yahoo.com or www.google.com
- Reasons, e.g. no single point of failure and load-balancing
- Reasons, e.g. politics - webservers scattered around the world
- A **label** points to a **group** of servers

DNS methods to transparently connect to single webserver in group

DNS hierarchy

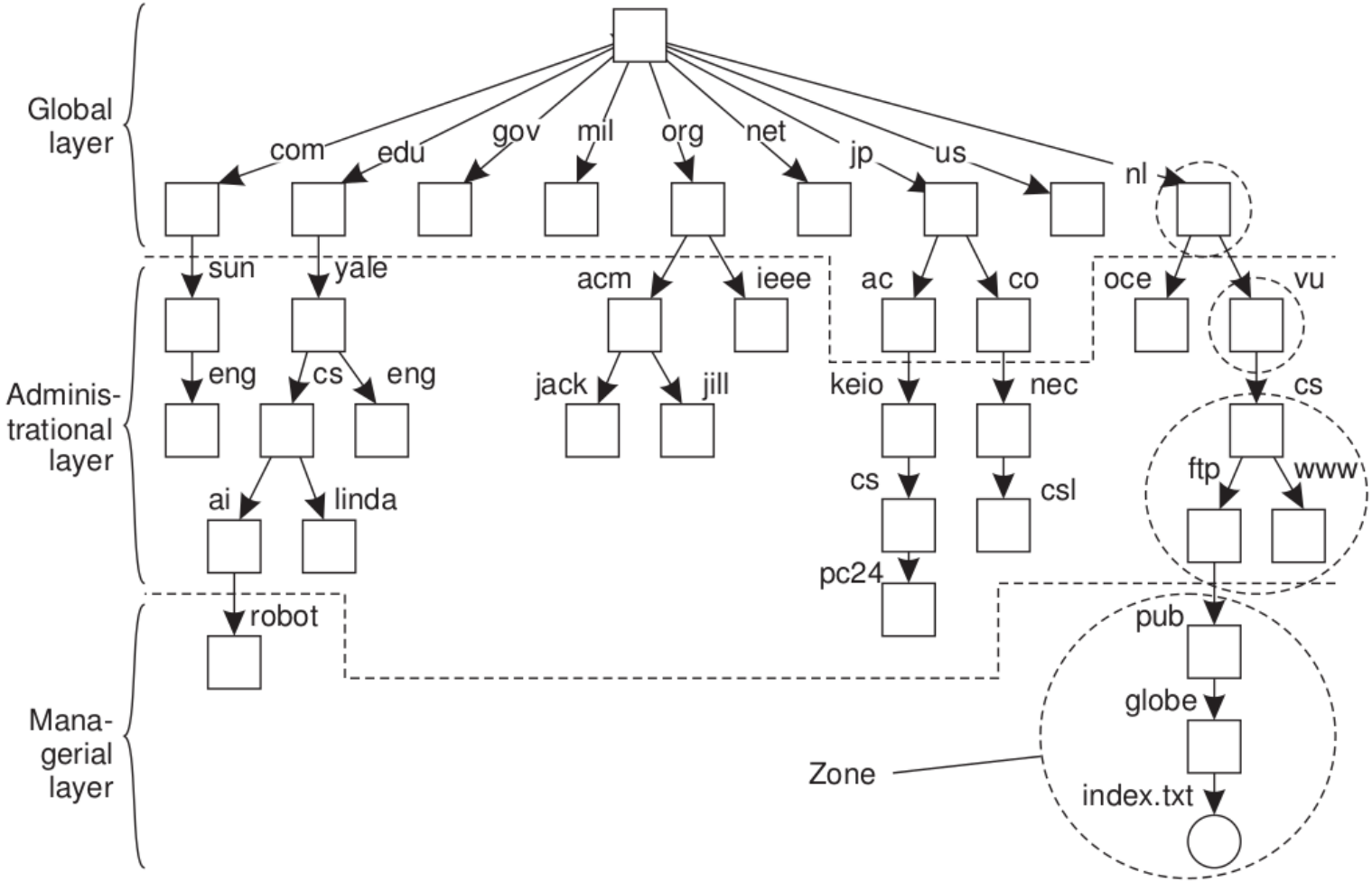


Figure: Courtesy of A. Tanenbaum and M. van Steen

DNS hierarchy tracing example I/II

Let us trace through the DNS hierarchy with: `dig au.dk +trace`

1. `dig` heads to the root name servers
2. then to the servers responsible for all the *.dk domains
3. finally to the name servers responsible for au.dk

DNS hierarchy tracing example II/II

```
.                       81815   IN      NS      f.root-servers.net.
.                       81815   IN      NS      a.root-servers.net.
.                       81815   IN      NS      h.root-servers.net.
.                       81815   IN      NS      c.root-servers.net.
.                       81815   IN      NS      e.root-servers.net.
.                       81815   IN      NS      d.root-servers.net.
.                       81815   IN      NS      l.root-servers.net.
.                       81815   IN      NS      g.root-servers.net.
.                       81815   IN      NS      k.root-servers.net.
.                       81815   IN      NS      j.root-servers.net.
.                       81815   IN      NS      m.root-servers.net.
.                       81815   IN      NS      b.root-servers.net.
.                       81815   IN      NS      i.root-servers.net.
;; Received 512 bytes from 127.0.0.1#53(127.0.0.1) in 198 ms

dk.                     172800  IN      NS      a.nic.dk.
dk.                     172800  IN      NS      l.nic.dk.
dk.                     172800  IN      NS      b.nic.dk.
dk.                     172800  IN      NS      p.nic.dk.
dk.                     172800  IN      NS      s.nic.dk.
dk.                     172800  IN      NS      c.nic.dk.
;; Received 331 bytes from 192.112.36.4#53(192.112.36.4) in 286 ms

au.dk.                  86400   IN      NS      ns.au.dk.
au.dk.                  86400   IN      NS      ns-soa.darenet.dk.
au.dk.                  86400   IN      NS      ns2.au.dk.
;; Received 163 bytes from 193.163.102.222#53(193.163.102.222) in 48 ms

au.dk.                  86400   IN      A       130.225.9.11
au.dk.                  86400   IN      NS      ns.au.dk.
au.dk.                  86400   IN      NS      ns2.au.dk.
au.dk.                  86400   IN      NS      ns-soa.darenet.dk.
```

Iterative name resolution

Assume

- **No** name server replication and **no** client-side caching
- Resolve: <nl,vu,cs,ftp> $\Leftrightarrow$ ftp.cs.vu.nl
- $\#<cs>$: Addr of server handling node referred to by $<cs>$

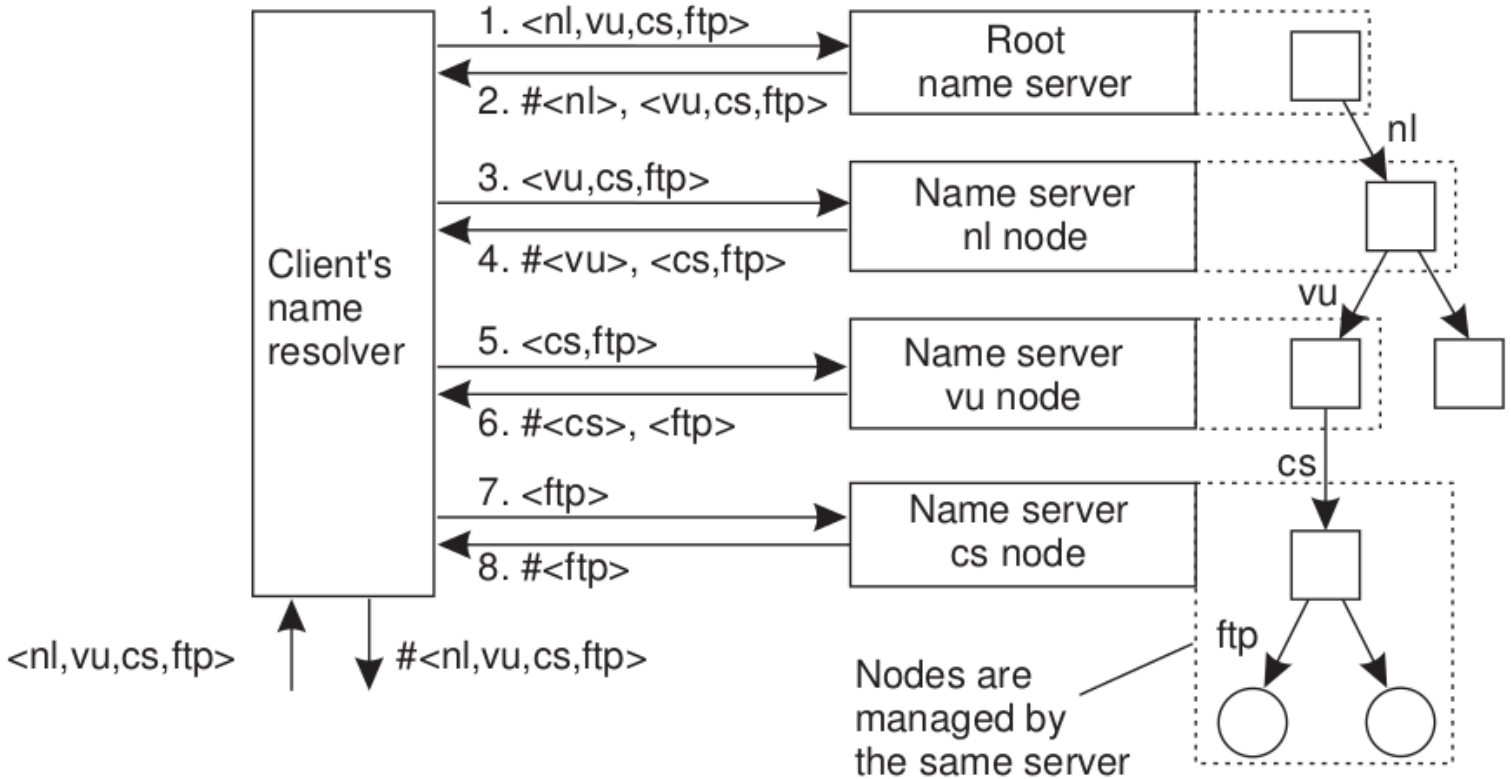


Figure: Courtesy of A. Tanenbaum and M. van Steen

Recursive name resolution

Assume

- **No** name server replication and **no** client-side caching
- Resolve: <nl,vu,cs,ftp> $\Leftrightarrow$ ftp.cs.vu.nl
- $\#<cs>$: Addr of server handling node referred to by $<cs>$

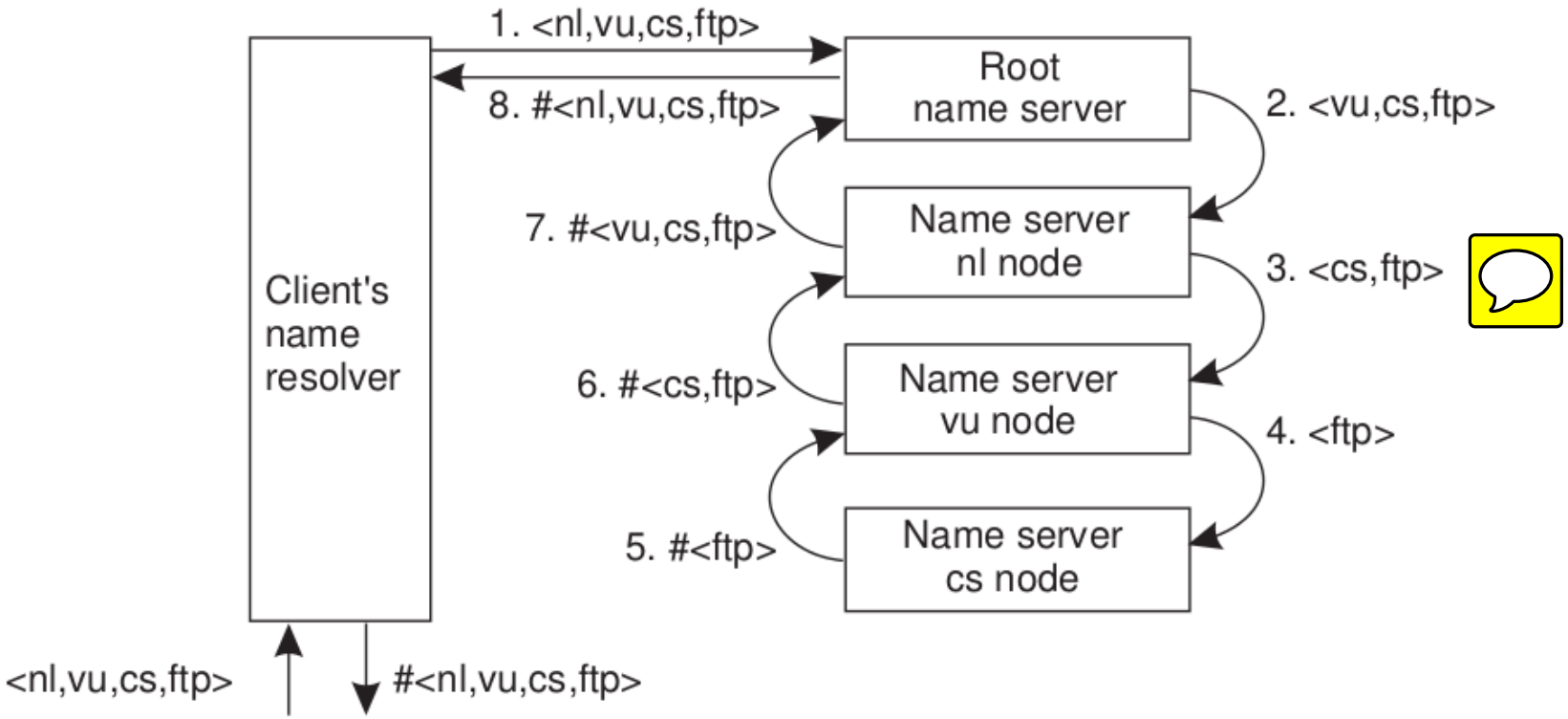


Figure: Courtesy of A. Tanenbaum and M. van Steen

Pros and cons of iterative and recursive resolution

Recursive

- Pro: Caching more effective compared to iterative resolution
- Pro: Client-side communication costs are reduced
- Con: Performance demand on each name server

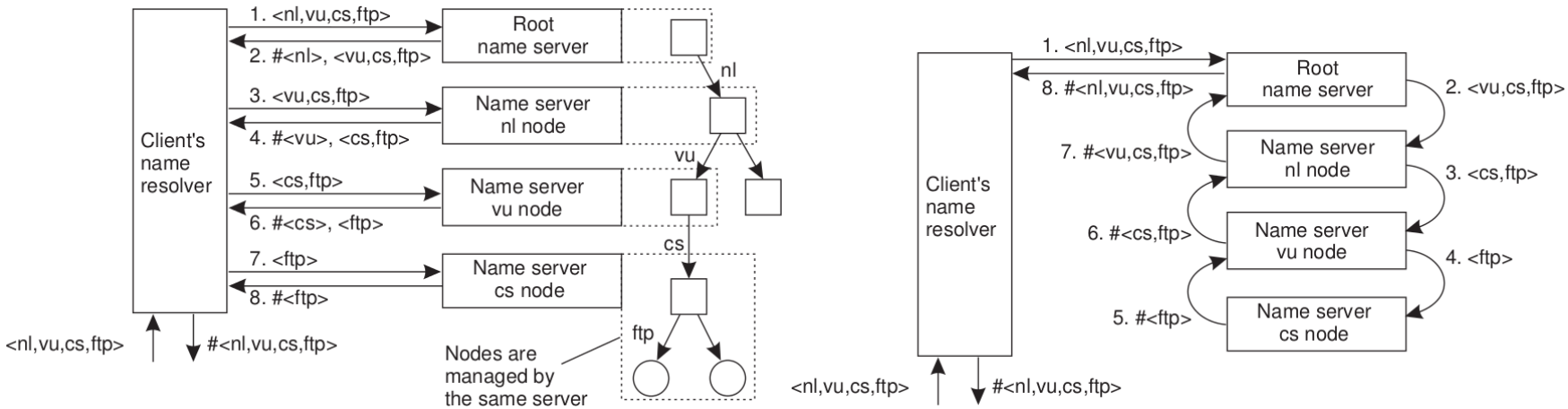


Figure: Courtesy of A. Tanenbaum and M. van Steen

How to get a domain name

Say we want to set up it-onk.dk

- .dk it the country code top level domain (ccTLD) for Denmark
- The .dk ccTLD was created on **July 14, 1987** at ARPA Network Information Center, Stanford Research Institute

Registrar

- DK Hostmaster handles exclusively the .dk ccTLD

Registrator

- Any new .dk domain name has to be applied for by an **approved registrator**. See the list on dk-hostmaster.dk

Applicant

- Apply any approved registrator from your pc, e.g. at home
- Most registrators manage the domain name, e.g. dns, web, ...
- Sometimes the registrar offers to manage domain names too
- From your registrator account you may point to own DNS

Why point to own DNS

Manage your own in-house LAN

- Associate host names internally on own in-house LAN

Filter shady sites by OpenDNS.org

- Maintains a list of shady sites, e.g. malware, virus, ...
- Redirects to secure page if trying to access a black listed one
- Good for, e.g. homes w/ children, schools, etc.

Some DNS records

"A"

- Host name $\Leftrightarrow$ IP address

Canonical name (CNAME)

- Name to name, i.e. www.yahoo.com $\Leftrightarrow$ foobar

Name Server (NS)

- Authoritive DNS server (main DNS server)
- SOA: Start of authority (your first in line DNS server)

Mail exchanges (MX)

- Mail servers

DNS records example: `dig yahoo.com ANY`

```
;; ANSWER SECTION:
yahoo.com.              712     IN      MX      1 mta5.am0.yahoodns.net.
yahoo.com.              712     IN      MX      1 mta6.am0.yahoodns.net.
yahoo.com.              712     IN      MX      1 mta7.am0.yahoodns.net.
yahoo.com.              757     IN      A       206.190.36.45
yahoo.com.              757     IN      A       98.138.253.109
yahoo.com.              757     IN      A       98.139.183.24
yahoo.com.              171324  IN      NS      ns4.yahoo.com.
yahoo.com.              171324  IN      NS      ns1.yahoo.com.
yahoo.com.              171324  IN      NS      ns8.yahoo.com.
yahoo.com.              171324  IN      NS      ns2.yahoo.com.
yahoo.com.              171324  IN      NS      ns3.yahoo.com.
yahoo.com.              171324  IN      NS      ns5.yahoo.com.
yahoo.com.              171324  IN      NS      ns6.yahoo.com.

;; AUTHORITY SECTION:
yahoo.com.              171324  IN      NS      ns8.yahoo.com.
yahoo.com.              171324  IN      NS      ns5.yahoo.com.
yahoo.com.              171324  IN      NS      ns1.yahoo.com.
yahoo.com.              171324  IN      NS      ns3.yahoo.com.
yahoo.com.              171324  IN      NS      ns6.yahoo.com.
yahoo.com.              171324  IN      NS      ns4.yahoo.com.
yahoo.com.              171324  IN      NS      ns2.yahoo.com.

;; ADDITIONAL SECTION:
mta5.am0.yahoodns.net.  61      IN      A       66.196.118.33
mta5.am0.yahoodns.net.  61      IN      A       66.196.118.35
mta5.am0.yahoodns.net.  61      IN      A       66.196.118.36
mta5.am0.yahoodns.net.  61      IN      A       74.6.136.244
mta5.am0.yahoodns.net.  61      IN      A       98.136.216.25
mta5.am0.yahoodns.net.  61      IN      A       98.136.217.202
mta5.am0.yahoodns.net.  61      IN      A       98.138.112.38
mta5.am0.yahoodns.net.  61      IN      A       66.94.238.147
```

Who is responsible

Internet Corporation for Assigned Names and Numbers (ICANN)

- Non profit organization
- Top level management
- Internet Protocol address space allocation
- TLD name system and root server system management

Internet Assigned Numbers Authority (IANA)

- Non profit organization
- Middle management
- Maintains and publishes the "root zone file"
- Allocates Internet Name Space to Regional Internet Registries
- Oversees the root operators and TLD registrars

Root operators

- 12 root operators
- Owns and maintains the root DNS servers

12 root operators manage 13 root servers

- **A** VeriSign Global Registry Services
- **B** Information Sciences Institute
- **C** Cogent Communications
- **D** University of Maryland
- **E** NASA Ames Research Center
- **F** Internet Systems Consortium, Inc.
- **G** U.S. DOD Network Information Center (original ARPnet)
- **H** U.S. Army Research Lab
- **I** Autonomica/NORDUnet
- **J** VeriSign Global Registry Services
- **K** RIPE NCC
- **L** ICANN
- **M** WIDE Project

Root Operators

The root operators

- Do not determine the content of the root zone file
- The file is edited by IANA according to a defined process
- Root name server operators publish file as received from IANA

DNS root name servers

- Publish the contents of one small file to the internet
- The file is called the root zone file
- Dec. 12, 2004: 5335 lines of text in the file. Size 119KB, cf. http://www.isoc.org/briefings/020/zonefile.shtml

The actual name server machines

- Dec. 2004: Located in more than 80 locations in 34 countries
- Locations are confidential

DNS servers

- Variety of DNS implementations on variety of OS
- Safe-guard towards viruses, buggy SW etc.

DNS backbone DDoS attack

Substantial DNS backbone DDoS attack on October 22, 2002

- Distributed denial of service (DDoS) attack
- Tageted the 13 main DNS root servers
- Lasted approximately 1 hour
- 9 out of 13 servers were disabled
- Trying to disable the entire internet
- The internet has never "gone down"

Malfunction on July 1997

- Largest malfunction of DNS servers: 7 out of 13 machines
- Due to technical problem

Part II

Berkeley Internet Name Domain (BIND) DNS

Outline

Introduction and installation

Caching nameserver and forwarder

(Primary master)

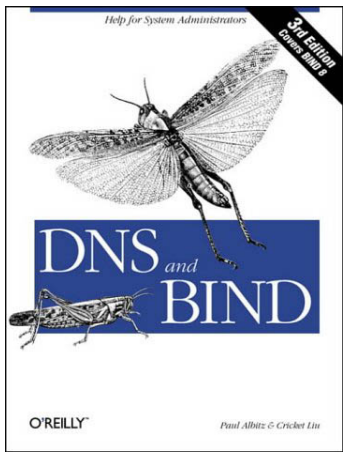
(Secondary master)

References

Berkeley Internet Name Domain (BIND)

- The most widely used DNS server SW on the internet
- De facto standard on Unix-like operating systems
- Open source implementation of the DNS protocols
- Originally by four graduate students in early 1980s
  - Computer Systems Research Group at Uni. of Calif., Berkeley
- GNU/Linux, Net/Free/Open BSD, Mac OS X, Windows
- Supported by the Internet Software Consortium, www.isc.org

ISC Internet Systems Consortium



BIND for Linux

You can get BIND for

- GNU/Linux, Net/Free/Open BSD, Mac OS X, and Windows

We will **only** look at BIND from a

- **Linux** perspective

Install BIND

- Install: `sudo apt-get install bind[n]`.
  - Currently: $\texttt{n} = 9$
- Check version: `named -v`
  - Currently: `BIND 9.8.1-P1`
- Latest version from ics.org
  - Currently: $\texttt{n} = 10$

Check installation

Make sure you have installed the `dnsutils` package

- sudo apt-get install dnsutils

Test setup with `dnsutils`' DNS lookup utility `dig`

- `dig` the loopback interface: Listening on port 53?
- `dig -x 127.0.0.1`

If successful you get get lines similar to

- `Query time:  1 msec`
- `SERVER: 192.168.1.10#53(192.168.1.10)`

Common configuration of BIND

Caching nameserver and forwarder

- Caching: BIND finds answers to name queries and caches these. This speeds up the DNS process.
- Forwarding: If your DNS server cannot process a request, the request is forwarded to another DNS server.

Primary master DNS server

- BIND reads the data for a zone from a file on it's host and is authoritative for that zone.

Secondary master DNS server

- BIND gets the zone data from another nameserver authoritative for the zone.

Configuration files

Configuration files are stored in

- `/etc/bind/`

Main configuration files are

- `/etc/bind/named.conf.local`
- `/etc/bind/named.conf.options`
- `/etc/bind/named.conf`

Outline

Introduction and installation

Caching nameserver and forwarder

(Primary master)

(Secondary master)

References

Caching nameserver

BIND's **default** configuration

- Setup as **caching server**, i.e. store answers to name queries
- Objective is to **reduce** time of future queries

Test whether DNS lookup time is reduced by caching

- `dig` an outside domain to check the **query time**
- Note query time at end of command output
- 1st: `dig ubuntu.com` ⇒ `Query time: 59 msec`
- 2nd: `dig ubuntu.com` ⇒ `Query time: 1 msec`

Forwarder

Configure the **forwarder**

- If your DNS server cannot process a request, the request is forwarded to another DNS server
- Add the IP address(es) of public DNS servers, e.g. your ISP's, Google's or OpenDNS'

Example (`/etc/bind/named.conf.options`)

```
//Forward to OpenDNS' public DNS servers
  forwarders {208.67.222.222;
              208.67.220.220;
              };
```

Use local server as DNS server

- BIND comes with default `localhost` (127.0.0.1) settings
- See: `dig localhost`
- We want to use our local server as the DNS server

Example (`/etc/resolv.conf`)

```
nameserver 127.0.0.1
nameserver xxx.xxx.xxx.xxx
search example.com
```

How to edit `/etc/resolv.conf`. Two options

1. Delete all the contents and add `nameserver 127.0.0.1`
2. Add `nameserver 127.0.0.1` in `/etc/resolvconf/resolv.conf.d/head` (Ubuntu $\geq$ 12.04)

Enabling the new configurations

Update `/etc/resolv.conf`

- `sudo resolvconf -u`

Restart the BIND DNS server

- `sudo /etc/init.d/bind9 restart`

Maybe needed: Clear the DNS cache

- `sudo /etc/init.d/nscd restart` or
  `sudo /etc/init.d/networking restart`

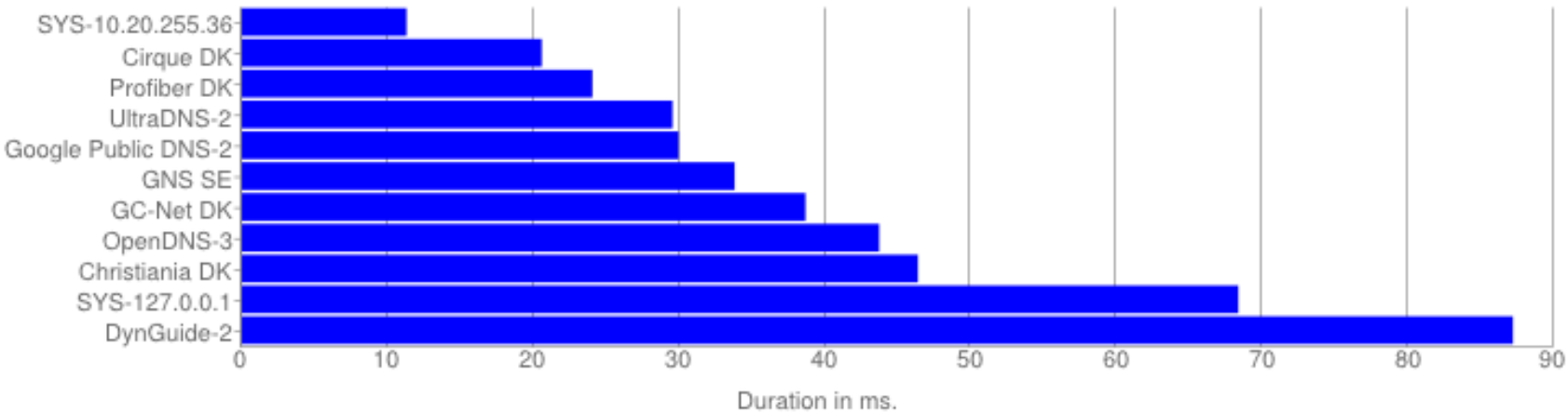
Test OpenDNS connection

Test whether using OpenDNS

- Go to OpenDns test site: **welcome.opendns.com**

Test whether good/fast choice w/ Google Name Bench

- **Mean Response Duration**



What happens

When opening a web page

- First the request is sent to your local DNS server
- If the domain name of the page is not found then BIND forwards that request to OpenDNS's server
- OpenDNS is generally faster than many conventional DNS servers, i.e. you may decrease your query time
- Newly fetched domain name is cached in local BIND server
- When you visit the page for the second time, your local server can resolve the request and the page loads instantly, again providing decrease in query time

Outline

Introduction and installation

Caching nameserver and forwarder

(Primary master)

(Secondary master)

References

Outline

Introduction and installation

Caching nameserver and forwarder

(Primary master)

(Secondary master)

References

Outline

Introduction and installation

Caching nameserver and forwarder

(Primary master)

(Secondary master)

References

Links: Domain Name System

- ICANN, Internet Corporation for Assigned Names and Numbers
- IANA, Internet Assigned Numbers Authority
- Internet Society
- Root servers

Links: BIND

- Internet Systems Consortium
- Administrator's Reference Manual
- BIND configuration
- Testing BIND with `dig`
- BIND for the small LAN
- Setting up DNS for the LAN on Ubuntu 12.04 server
- BIND9 Server How To
- Setup a DNS server with bind
- How to run a local caching name server with BIND

Links: OpenDNS

- OpenDNS
- OpenDns test site
- Name Bench

Links: Ubuntu

- DNS in Ubuntu 12.04
- Networking tips and tricks
- How do I add a DNS server via resolv.conf?
- Network Configuration: Name resolution

References I

[1] Coulouris, G., J. Dollimore, T. Kindberg, and G. Blair (2001). *Distributed systems: Concepts and design*. Pearson.

[2] Tanenbaum, A. (1995). *Distributed operating systems*. Prentice Hall.