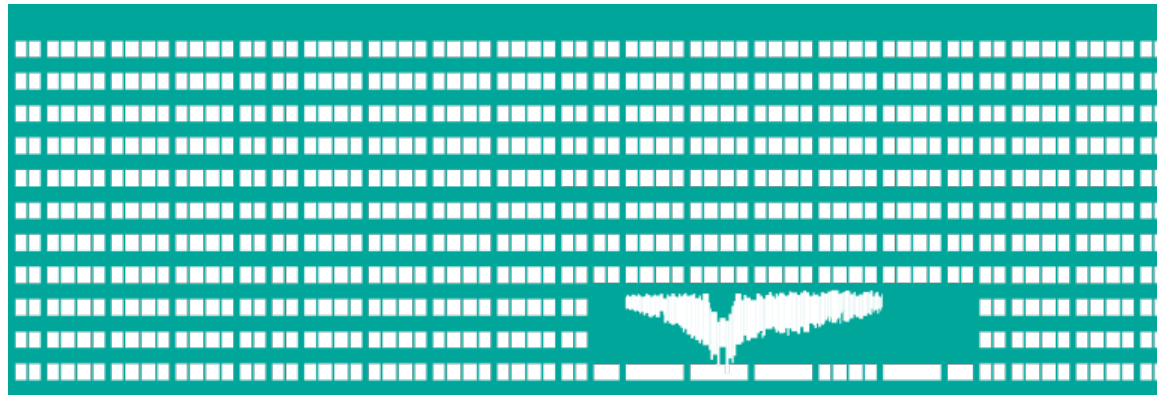


# Domain Name System (DNS)



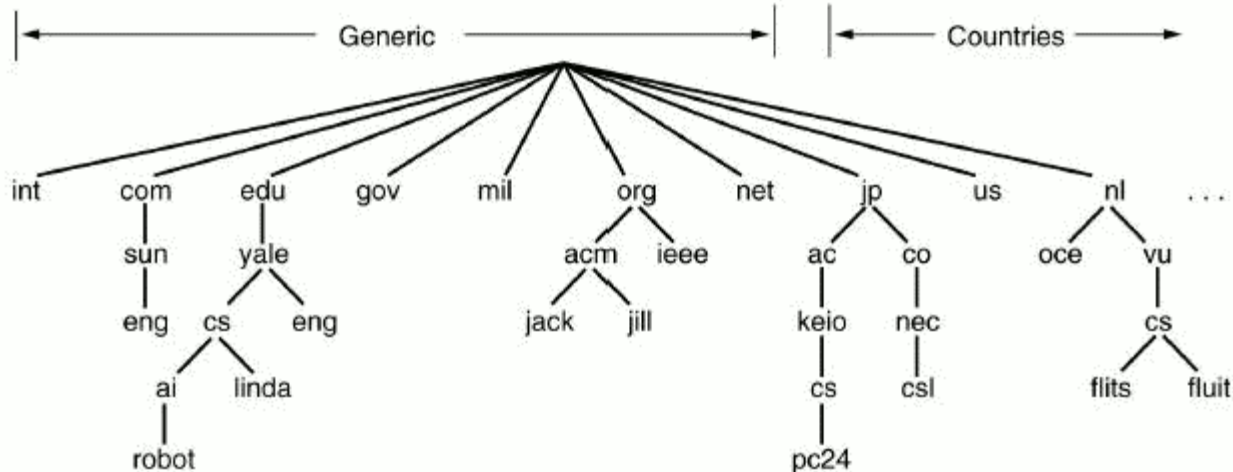
**Computer Networks  
Seminar 11**

# DNS Introduction (Domain Name System)

- Naming system used in Internet
- Translate domain names to IP addresses and back
- Communication works on UDP (port 53), large requests/responses are sent over TCP (port 53)
- DNS server processes requests, which gets from Resolver, and responses to them
- Resolver is system component, which communicates with DNS server

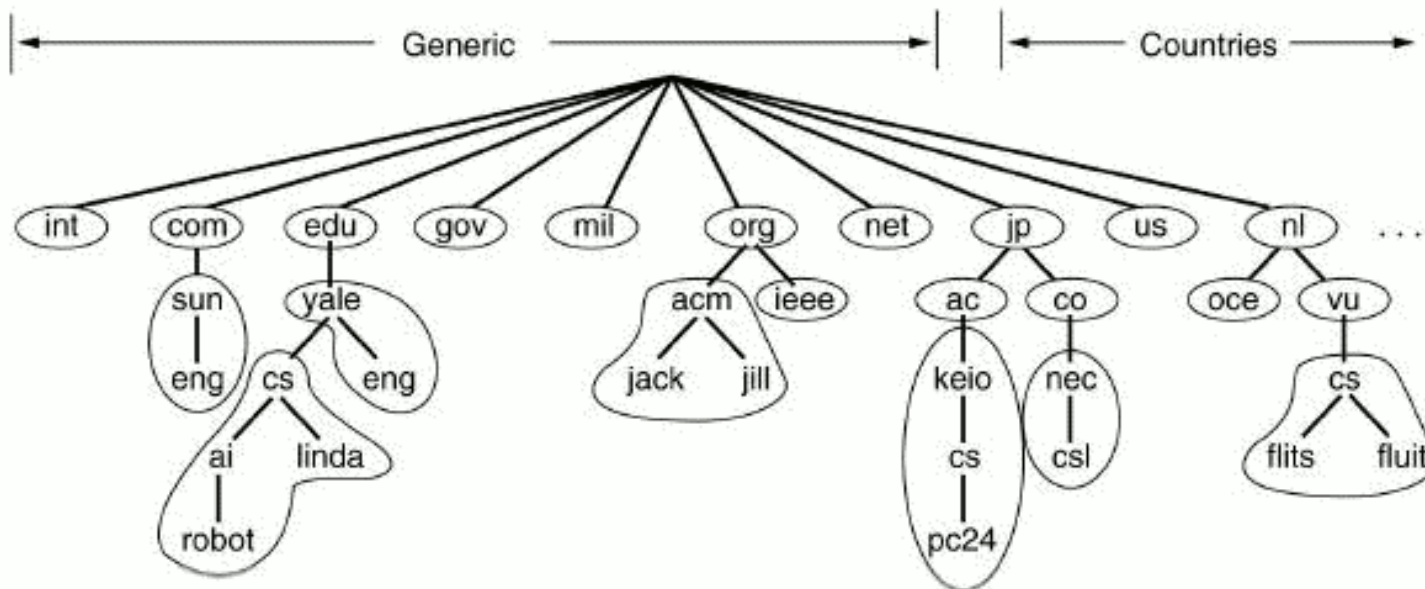
# Domains

- Domains:
  - generic: .edu, .com, ...
  - country codes : .cz, .it, .uk, ...
- Domain names are hierarchically organized (root of the tree is domain “.”)
- Maximum domain name length is 256 char. (1 subdomain max. 63 characters)



# Zone

- Zone is a part of the tree stored on single DNS server
- DNS is authoritative for domains which are contained in the zone the server controls



# DNS record types

- **SOA** - *Start of authority* - specifies authoritative information about a DNS zone, including the email of the domain administrator and several timers relating to refreshing the zone.
- **NS** - *Name server* - delegates a DNS zone to use the given authoritative name servers
- **MX** - *Mail exchange* - maps a domain name to a list of mail exchange servers for that domain
- **A/AAAA** - *Address* - returns IPv4/IPv6 address to given domain name
- **CNAME** - *Canonical name* - alias of one name to another
- **PTR** - *Pointer* - for storing reverse records

# Getting the information from DNS server - nslookup program

- Alternatives of **dig** program used in OS Windows
- Commands:
  - **set type=<record\_type>**
    - (NS,A, ... or ANY, which displays all the records)
- Example:
  - **C:\> nslookup**
    - > **server DNS\_server**
    - > **set type=A**
    - > **home1.vsb.cz**
- Another utility – **host**

# Getting the information from DNS server - command dig

- Looks for and displays the information from DNS server (Linux)
- Parameters of program dig:
  - @<server> - name or IP address of DNS server
  - -t <record\_type> - specifies record type
  - -p <port> - if we are using nonstandard port
- Example: **dig -t A home1.vsb.cz** (or dig home1.vsb.cz A)
- DNS server response:
  - **QUESTION SECTION** - request on DNS server
  - **ANSWER SECTION** - response to the request
  - **AUTHORITY SECTION** - tells which DNS server is authority
  - **ADDITIONAL SECTION** - additional information, usually contains IP addresses of authoritative DNS servers

# Example of command DIG requests

- Gets IP address of yahoo.com
  - **dig** yahoo.com -t A or dig yahoo.com A
- Gets a list of mail servers for yahoo.com
  - **dig** yahoo.com MX +noall +answer
- List of authoritative DNS servers for yahoo.com
  - **dig** yahoo.com NS +noall +answer
- Displays all we tried with previous commands
  - **dig** yahoo.com ANY +noall +answer
- Gets PTR record
  - **dig** 49.149.196.158.in-addr.arpa. ANY +noall +answer
  - **dig -x** 158.196.149.9



# Configuration of DNS server

## BIND

- **Bind** is DNS server implementation for OS OS Linux, Windows and FreeBSD. Configuration is split into few files.
- /etc/bind/**named.conf** - main configuration file. There are zones defined for which the server is authoritative (in named.conf.\*).
  - options { //in **named.conf.options** file nowadays
    - directory "/var/cache/bind";  
//where bind looks for configuration files
    - recursion yes;** //allows recursive lookup
    - // uncomment forwarders and set them to 158.196.0.53**
    - // set dnssec-validation to no** if it causes problems
    - ...
  - };
  - zone "." { // link to the zone file with root servers
    - type hint; //means that contains only list of root servers
    - file "/etc/bind/db.root";
  - };
- /etc/bind/db.\* - records definition for particular zone (e. g. db.testEB4x)

# testEB4 zone definition

- In bind distribution we implicitly find some zones pre-configured (**localhost**, **127.in-addr.arpa**, **0.in-addr.arpa**)
- Definition of next zone in file **named.conf.\*** could look like following:
  - zone "testEB4x.cs.vsb.cz" {  
type master; //this name server will be  
//primary and authoritative  
//for this domain.  
file "/etc/bind/db.testEB4x";  
//File with definitions of the records  
};

# Configuration of zone testEB4 file db.testEB4

- **\$ORIGIN cs.vsb.cz.**
  - The value ORIGIN is implicitly added to the names, which doesn't end with dot.
- **\$TTL 604800**
  - How long the record will be kept in cache
- **SOA** record must **always** be written **one time** at the beginning of zone file:

```
testEB4x IN SOA ns.testEB4x admintestEB4.vsb.cz.  
                (2018092414 ;  
                604800 ;  
                ... )
```
- **ns.testEB4x**
  - Name of domain primary DNS server (ns.testEB4.cs.vsb.cz.)
- **admintestEB4x.vsb.cz.**
  - E-mail of domain administrator (use “.” instead of @)

# Configuration of zone testEB4x file db.testEB4x

- **SOA** record should be followed by **NS** record specifying DNS server for given domain (\$ORIGIN testEB4.cs.vsb.cz.)

	<b>NS</b>	<b>a.ns</b>
<b>a.ns</b>	<b>A</b>	<b>158.196.135.88</b>

- Assigning IP address to pc1.testEB4.cs.vsb.cz.

<b>pc1</b>	<b>A</b>	<b>158.196.135.66</b>
	<b>TXT</b>	<b>"computer No.1"</b>

- Alias definition for pc1

<b>www</b>	<b>CNAME</b>	<b>pc1</b>
------------	--------------	------------

- Subdomain NS definition

<b>subdom</b>	<b>NS</b>	<b>ns.subdom</b>
<b>ns.subdom</b>	<b>A</b>	<b>158.196.135.66</b>

# DNS server configuration for reverse lookup

- It serves for IP address to domain name mapping
- We put zone definition for reverse lookup into file **named.conf**
- Domain name of the record for reverse lookup of address A.B.C.D is  
D.C.B.A.in-addr.arpa.
- ```
zone "135.196.158.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.135.196.158.in-addr.arpa";  
};
```

# Zone configuration

## 135.196.158.in-addr.arpa.

- Based on same rules used for configuring normal zone. There must exist records **SOA** a **NS** (name server responsible for domain)
- Instead of record **A** we use record **PTR**, which maps IP addresses to domain names
- **\$ORIGIN 135.196.158.in-addr.arpa.**  
**66 PTR pc1.testEB4x.cs.vsb.cz.**

# resolv.conf, hosts, host.conf

- Files containing **resolver** configuration (Linux), in **/etc**
  - **resolv.conf** – configuration of DNS on client's side
    - commands:
      - search <domain> – implicit added domain
      - nameserver <DNS server IP address>
  - **hosts** – manually (statically) configured addresses (also in Windows)
    - <IP address> <name> [<name2> ...]
  - **host.conf** – order of record sources (static and DNS addresses) in resolving (no effect in new OS)
    - order hosts, bind – first file hosts, after DNS

# Changing DNS setting in GUI

Details Identity **IPv4** IPv6 Security

**IPv4 Method**


Automatic (DHCP)  Link-Local Only  
 Manual  Disable  
 Shared to other computers

**DNS** Automatic

127.0.0.1|

Separate IP addresses with commas

**Routes** Automatic

| Address | Netmask | Gateway | Metric |                                                                                       |
|---------|---------|---------|--------|---------------------------------------------------------------------------------------|
|         |         |         |        |  |

Use this connection only for resources on its network

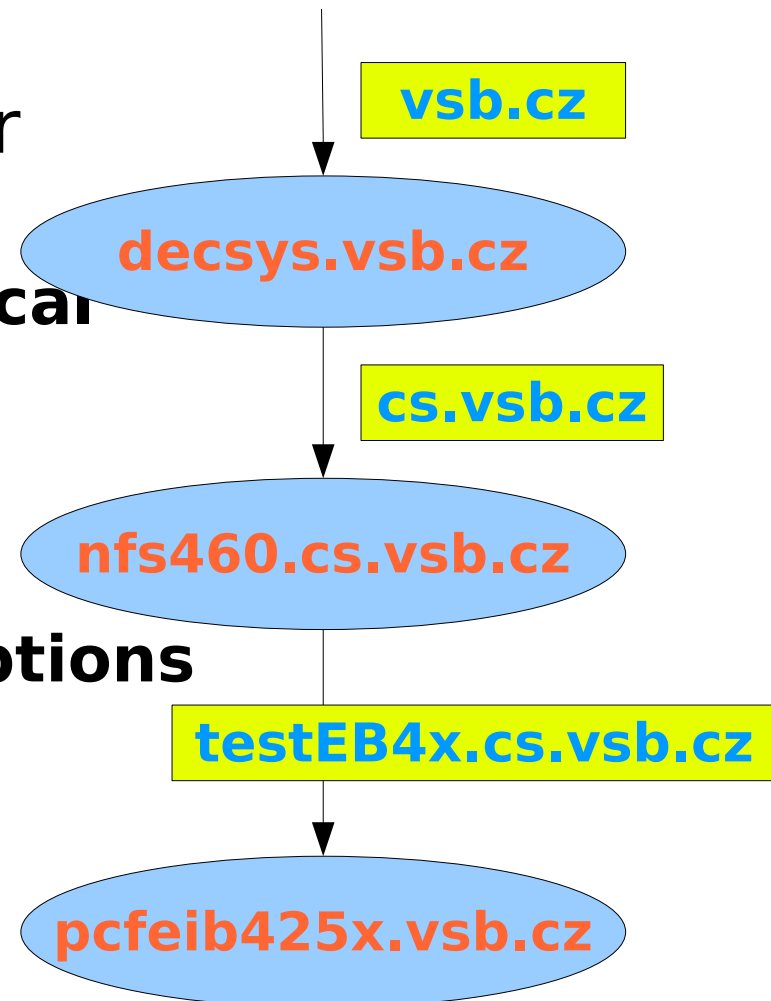


# Tools for checking BIND DNS server configuration

- Syntactic check of configuration files
  - **named-checkconf**  
*/etc/bind/named.conf*
    - Or for individual *named.conf.\** config files
  - **named-checkzone** *zone zone\_db\_file*
    - Out-of-zone data or missing A records for domains where part of DN is repeated twice mean there is an error as well.
- Running named in foreground (sudo)
  - **named -g**
  - If not possible, try: **service bind9 stop**

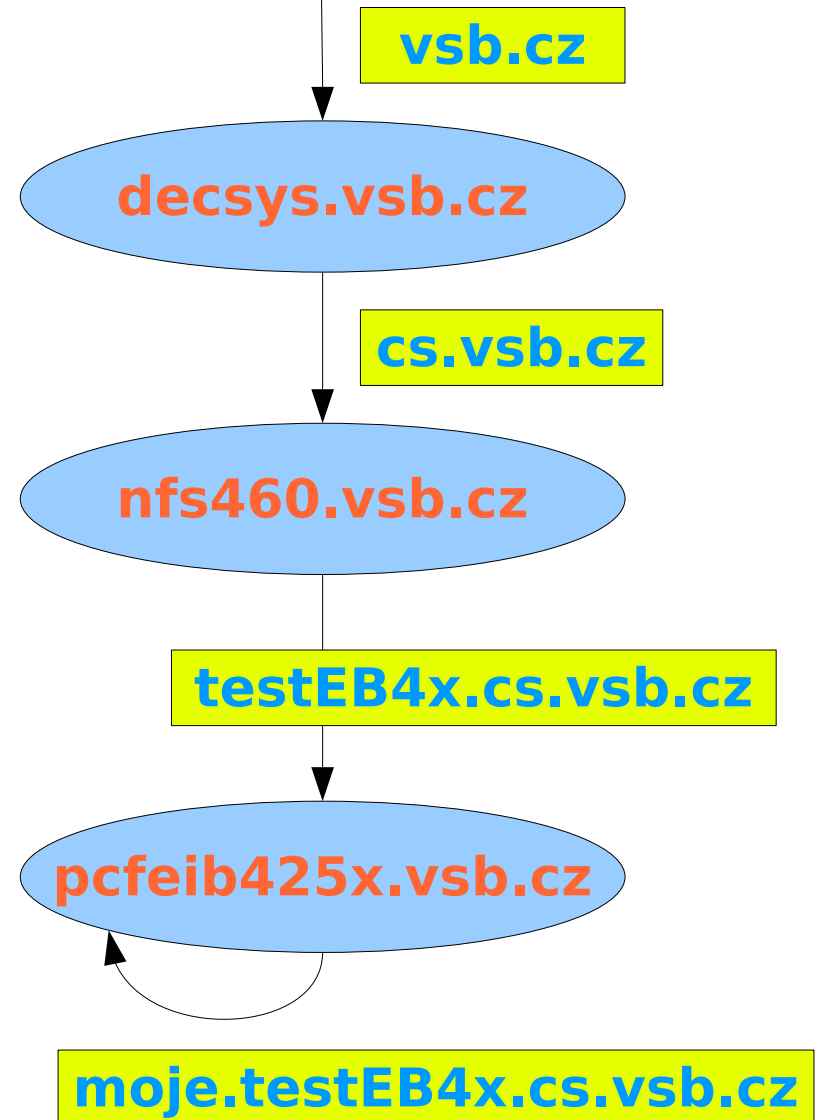
# First Task - basic DNS records

- Bind DNS server configuration
  - Add your DNS server for **testEB4x.cs.vsb.cz**
    - In **named.conf.local**
  - Add **SOA, A** and **TXT** records
  - Enable Reverse lookup
    - In **named.conf.options**
  - Set client resolver (**/etc/resolv.conf**)
  - Test the whole solution



# Second task - subdomain

- Add another level (subdomain) into DNS tree.
- Add an **SOA** record and all necessary **NS** and **A** records, for the subdomain add also **MX** and **TXT** records.



# Third task - reverse records

- Configure the DNS server to be authoritative for the Z.Y.X.0/24 subnet's **X.Y.Z.in-addr.arpa** zone, which will be assigned to you.
- Insert the **PTR** records for some addresses to this zone and test the reverse translation.
  - When checking the reverse translation, make sure the client is communicating directly with your DNS server.