

Bab V - DNS pada IPV6

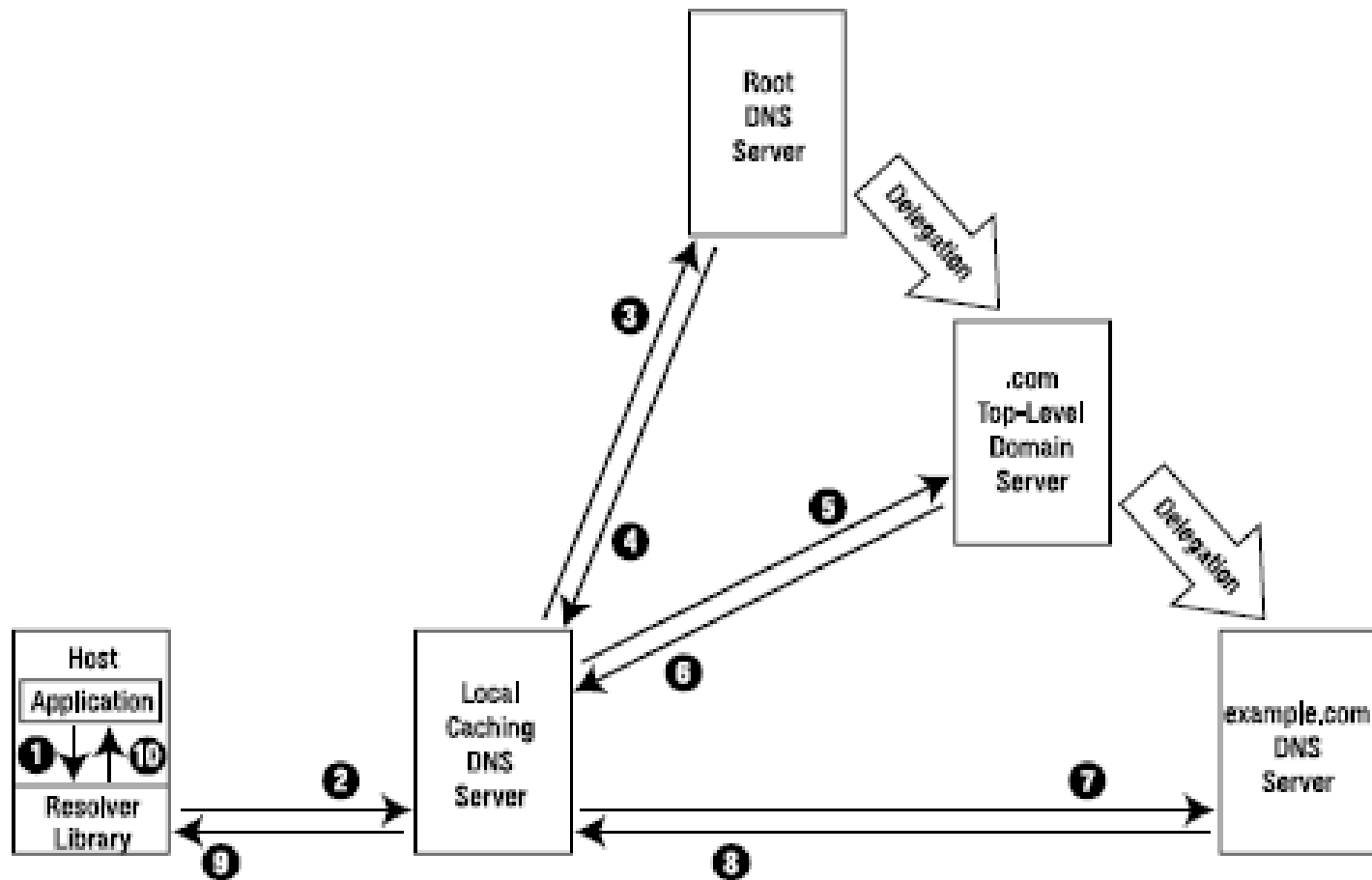
Iljitsch van Beijnum

A decorative graphic consisting of several horizontal lines of varying lengths and colors (teal, light blue, white) extending from the right side of the slide towards the center.

DNS

- Memang pada dasarnya untuk memahami IP address utamanya pada IPv6 sangatlah rumit
- Dan DNS berfungsi agar ketika kita berhadapan dengan IP lebih user-friendly
- Karena DNS melakukan translate dari nama DNS(yang kita tentukan) menuju IP address
- Dalam pengimplementasian DNS pada IPv6 sangatlah kompleks
- Pada bab ini kita akan menjelaskan tentang DNS dengan fungsi utamanya sebagai Domain name pada IP

DNS



Menampilkan IPv6 pada DNS

- Setiap DNS akan membuat sebuah name server, TLD server dan juga tujuan dari name server
- Untuk mengetahuinya maka perlu lah upgrade untuk ISP server agar dapat diimplementasikan IPv6 untuk dapat ditambahkan DNS didalamnya.
- Pada RFC 1886 mendiskripsikan data untuk melakukan publish informasi untuk IPv6

RFC 1886 :AAAA dan ip6.int

- IPv6 akan menampilkan AAAA yang kemudian dilakukan mapping dengan hexadecimal dan selanjutnya IPv6 akan diubah dengan ip6.int
- Ketika RFC dipublikasikan .int level domain adalah TLD infrastruktur dan saat ini terdapat beberapa domain yang didefinisikan untuk ARPANET

RFC 2874 :A6, DNAME Bitlabels dan ip6.arpa

- **A6 Name untuk mapping alamat**

Pada RFC 2874 ditampilkan untuk penomoran ulang dengan mengikuti telepon . Pada penelitian IBM untuk melakukan browse akan merequest nomor dan providing alamat yang selanjutnya dilakukan matching dengan telepon

```

www          IN  A6  64  ::0000:0000:0000:0390  subnet-a
subnet-a     IN  A6  48  0000:0000:0000:c001::  prefix-isp1
subnet-a     IN  A6  48  0000:0000:0000:c001::  prefix-isp2
prefix-isp1  IN  A6   0  2001:0db8:1bff::
prefix-isp2  IN  A6   0  3ffe:9500:003c::
  
```

Bitlabel dan alamat DNAME untuk nama Mapping

- Dengan menggunakan Dname record maka subdomain akan menampilkan data contohnya example.com dengan begitu terdapat bitlabel dengan mekanisme yang kurang perfect akan tetapi dapat menampilkan data dalam bentuk sub domain

```
research.example.com          IN DNAME  r-and-d.example.com.  
www.plastics.r-and-d.example.com.  IN A      192.0.2.1  
www.biotech.r-and-d.example.com.   IN A      192.0.2.2
```

RFC 1886 vs. RFC 2874

- Ketika mengimplementasikan RFC 1886 maka harus menampilkan IPv6 pada DNS
- Selanjutnya merubah library menjadi AAAA
- Setelah itu akan merecogize data dengan membentuk suatu nama DNS yang lebih familiar
- Walaupun DNAME merecord perubahan dengan caching nameserver maka selanjutnya data akan merecord dan memprosesnya dengan ASCII label yang normal

Kondisi Saat Ini

- Pada pengimplementasiannya, meskipun A6 memiliki record yang diperkuat, tetap akan memiliki perbedaan dalam implementasi di lapangan
- Besarnya grup implementasi melihat dari grup label pada ip6.arpa biasanya pada percobaan pertama akan mengalami kegagalan
- Jadi dapat disimpulkan untuk implementasi dari ip6.int memiliki spectrum yang sama dengan DNS pada dig.host dan nslookup pada BIND distribution

Instalasi dan Konfigurasi BIND

- Pada BIND akan aktif dengan internet system consortium, selanjutnya dengan menggabungkan IPv6 transprot anda akan mendapatkan 9.2x untuk bilabel dan A6 support
- Selanjutnya anda dapat menginstall tipe yang berbeda untuk data BIND pada linux atau FreeBSD
- Pada MacOS atau UNIX kita dapat melakukan Compile dengan menggunakan BIND kompiler

Starting BIND pada Boot Time

- Pada Red Hat Linux, terdapat file pada start up script tetapi tidak diaktifkan secara default
- Model ini dapat diselesaikan dengan cara boot pada `chkconfig named` pada command untuk membuat simbol link dengan file yang berada pada `/etc/init.d`
- Untuk FreeBSD file yang dapat diubah untuk mengaktifkan terdapat pada `/etc/rc.conf`

Konfigurasi BIND

- Untuk seluruh BIND extensive konfigurasi dapat didiskripsikan pada BIND administrator
- Selanjutnya untuk access pada TCP dan UDP dengan menggunakan port 53 dengan mengirim HUP signal dengan data yang didapat dari hasil konfigurasi pada file `/etc/rndc.conf`
- Nama server diambil pada file `/var/named` yang merupakan part dari setting BIND
- Direktori dari data tersebut akan ditulis ulang jika setting dari data ini berhasil

Konfigurasi BIND

```
options {
    directory "/var/named";
    allow-recursion { 192.0.2.0/24; 2001:db8:1bff::/48; };
    listen-on { 192.0.2.106; }
    listen-on-v6 { any; };

    # forward first;
    # forwarders { 192.0.2.53; };
    /* C-style comment */
    // C++-style comment
};

zone "." {
    type hint;
    file "named.root";
};
```

```
zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "localhost.rev";
};

zone "example.com" {
    type slave;
    file "example.com";
    masters { 192.0.2.53; };
};

zone "0.0.0.0.f.f.b.1.8.b.d.0.1.0.0.2.ip6.arpa."
{
    type master;
    file "db.2001:db8:1bff:0";
};
```

- Pada line command terdapat perubahan data yang kompatible pada pemrograman C dan C++, data ini seperti zone file yang mana inisial command akan menginstruksikan server untuk melakukan forward data dengan alamat 192.0.2.53 dan selanjutnya server tidak akan melakukan reply data dengan spesifikasi yang sama
- Berikut adalah contoh 4 spesifikasi data
 1. “dot ” zone dengan model point adalah named.root file
 2. Untuk 0.0.127.in-addr.arpa zone dengan membalik zone untuk localhost dan merubahnya menjadi primary zone
 3. Example.com zone yang mengauthorisasikan alamat dari nama server 192.0.2.53 pada example.com file
 4. Nibble-style ipv6.arpa zone untuk 2001:db8:1bff::/48

Pemilihan alamat pada nama server

- Spesifikasi dari nama server biasanya bekerja jika kita menambahkan data pada file DNS untuk perubahan nama file sesuai keinginan kita
 1. Alamat untuk nama server berada pada `/etc/resolv.conf`
 2. Alamat utama dan kedua pada DNS server akan di list pada TLD zone
 3. Alamat pada DNS utama berada pada `named.conf` pada server kedua
 4. Biasanya alamat pada DNS kedua untuk lokal akan di list pada server jika data diterima pada transfer form

Penambahan Informasi IPv6 pada Zone File

- Ketika menambah AAAA untuk populer service pada DNS zone maka anda akan dapat handle IPv6 traffic
- Dan selanjutnya service akan digunakan pada internet dengan jumlah IPv6 yang relative pada IPv4 traffic
- Bagaimanapun jika anda mengupgrade internal network, maka sangat mungkin seluruh host pada internal data dengan IPv6 akan memiliki AAAA record

Checking zone dan konfigurasi file dan reload named

```
# named-checkzone example.com /var/named/example.com
zone example.com/IN: loaded serial 2005020900
OK
# named-checkconf
# rndc reload
```

Membalikan pemetaan (Mapping)

- Untuk membalikan pemetaan dan juga besar dari pemetaan pada reguler zone, dengan merubah PTR record, selain SOA dan NS record
- Sangat dibutuhkan nibble format dan cara termudah untuk memasukkan alamat IPv6 pada nibble format adalah dengan menggunakan host command
- Host akan mengirim ulang nibble format dan melakukan copy paste menuju zone file

```

; 20050209   IvB   created

$TTL 86400

@   IN  SOA  ns1.example.com. root.example.com. ( 2005020900 28800 7200 604800 86400 )

       IN  NS      ns1.example.com.
       IN  NS      ns2.beispiel.de.

$ORIGIN 3.5.0.0.1.3.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
; ns1      IN  AAAA  2001:db8:31:53::53
3.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN  PTR  ns1.example.com.

$ORIGIN 1.0.0.0.1.3.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
; www      IN  AAAA  2001:db8:31:1:201:2ff:fe29:2640
0.4.6.2.9.2.e.f.f.f.2.0.1.0.2.0 IN  PTR  www.example.com.

; smtp     IN  AAAA  2001:db8:31:1:20a:95ff:fe2d:987a
a.7.8.9.d.c.e.f.f.f.5.9.a.0.2.0 IN  PTR  smtp.example.com.

$ORIGIN 1.3.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
0.0.2.c    IN  NS      ns.research.example.com.

```

RFC 1886 dan 2974 untuk pembalikan mapping

- Beberapa cara untuk mengatasi library yang asalah adalah dengan melihat apakah bitlabel ip6.arpa ada atau tidak, untuk memberi solusi dapat dengan set up pembalikan mapping untuk data tersebut
- Seperti contoh berupa data palsu dari pembalikan mapping :

```
$TTL 86400
```

```
@ IN SOA ns1.example.com. root.example.com. ( 2005020900 28800 7200 604800 86400 )
```

```
IN NS ns1.example.com.
```

```
*.\[x2/3].ip6.arpa. IN PTR bit.label.ip6.arpa.
```

```
\[x20010db800310001020a95ffffecd987a/128].ip6.arpa. IN CNAME a.7.8.9.d.c.e.f.f.f.5. 9.a.0.2.0.1.0.0.0.1.3.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
```

Update DNS secara Dinamis

- RFC2136 dikenalkan dengan menggunakan konsep update DNS secara otomatis
- Mekanisme ini memungkinkan client untuk bertanya tentang otorisasi dari server dan menambah informasi pada zone atau menghapus informasi yang telah ada dari zone
- Mekanisme ini memungkinkan host untuk menerima alamat baru untuk DHCP atau stateless autokonfigurasi untuk update dirinya sendiri dengan nama yang dibutuhkan