

Using DNS as an Access Protocol for Mapping Host Identifiers to Locators

Oleg Ponomarev
Helsinki Institute for
Information Technology (HIIT)

Madrid, 13.12.2007

HELSINKI
INSTITUTE FOR
INFORMATION
TECHNOLOGY

Introduction (1/2)

HIP = Host Identity Protocol (RFC 4423)

HIT = Host Identity Tag

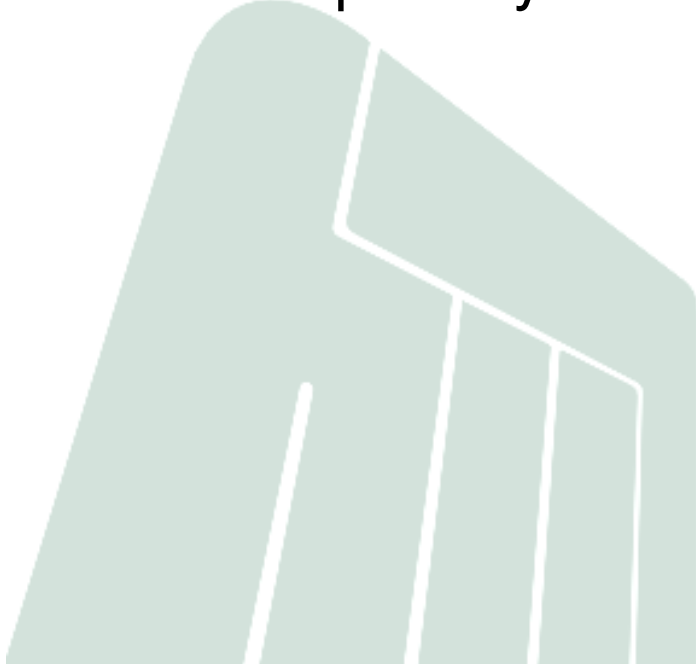
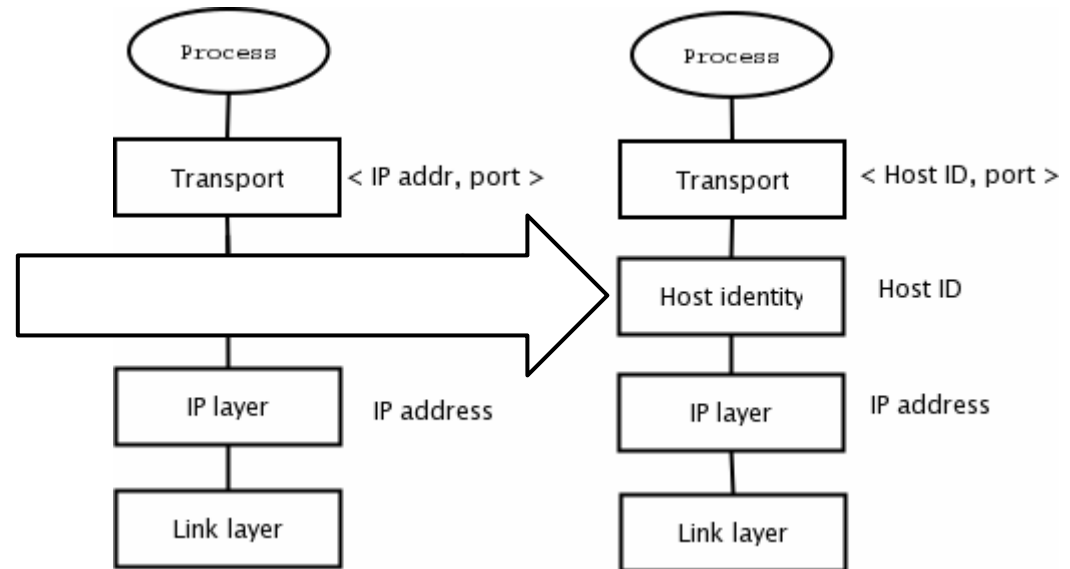
(hash of self-generated public key, such as
2001:15:6099:97fa:1b0c:4322:fb26:7ea1)

IP = Internet Protocol

(IP address ex: *193.167.187.1*,
2001:998:10:0:215:60ff:fe9f:60c4)

Introduction (2/2)

New layer between the internetworking and transport layers



HIT -> IP Address (1/2)

Current implementation stores data in OpenDHT, but we may use DNS:

1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.

2.hit-to-ip.infracore.net.

IN A 193.167.187.1

IN AAAA 2001:470:1f00:ffff::1bb3

IN AAAA 2001:998:10:0:215:60ff:fe9f:60c4

- My HIT was 2001:15:6099:97fa:1b0c:4322:fb26:7ea1
- Location might be more flexible, e.g. an IP address and a UDP port

OpenDHT vs DNS (1/3)

```
time ./put.py colors red    time ./put.py colors green    time ./get.py colors
real  1m8.558s              real  0m1.223s                real  0m1.049s
user  0m0.156s              user  0m0.150s                user  0m0.156s
sys   0m0.022s              sys   0m0.023s                sys   0m0.020s
```

```
time ./put.py animals tiger    time ./get.py animals tiger
real  0m0.546s              real  0m0.352s
user  0m0.096s              user  0m0.100s
sys   0m0.016s              sys   0m0.012s
```

```
time nsupdate< update.txt
real  0m0.105s
user  0m0.000s
sys   0m0.004s
```

```
time dig 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracore.net.
real  0m0.008s
user  0m0.000s
sys   0m0.000s
```

OpenDHT vs DNS (2/3)

```
[planetlab1.diku.dk]$ time ./put.py towns Aachen  
real 0m1.314s  
user 0m0.185s  
sys 0m0.027s
```

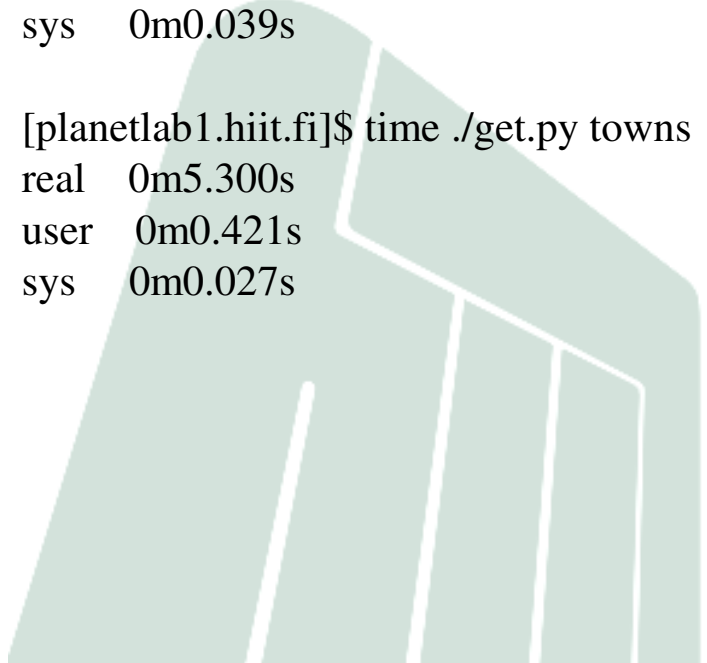
```
[planetlab1.hiit.fi ~]$ time ./get.py towns  
real 0m2.903s  
user 0m0.431s  
sys 0m0.019s
```

```
[planetlab1.diku.dk]$ time ./get.py towns  
real 0m1.126s  
user 0m0.178s  
sys 0m0.039s
```

```
[planetlab1.diku.dk ~]$ time ./get.py towns  
real 0m2.329s  
user 0m0.186s  
sys 0m0.035s
```

```
[planetlab1.hiit.fi]$ time ./get.py towns  
real 0m5.300s  
user 0m0.421s  
sys 0m0.027s
```

```
[planetlab1.hiit.fi ~]$ time ./get.py zzzzzzzzzzzz  
real 0m3.727s  
user 0m0.418s  
sys 0m0.028s
```



OpenDHT vs DNS (3/3)

```
15:14:51.138879 IP 137.226.59.118.46496 > 137.226.12.31.domain: 61489+ AAAA? opendht.nyuld.net. (35)
15:14:51.139144 IP 137.226.12.31.domain > 137.226.59.118.46496: 61489 1/1/0 CNAME[[domain]
15:14:51.139254 IP 137.226.59.118.46496 > 137.226.12.31.domain: 7881+ A? opendht.nyuld.net. (35)
15:14:51.139469 IP 137.226.12.31.domain > 137.226.59.118.46496: 7881 2/0/0 CNAME[[domain]
15:14:51.139648 IP 137.226.59.118.33646 > 130.104.72.201.5851: S 2902443105:2902443105(0) win 5840 <mss
1460,sackOK,timestamp 110486255 0,nop,wscale 6>
15:14:51.160524 IP 130.104.72.201.5851 > 137.226.59.118.33646: S 1423455886:1423455886(0) ack 2902443106 win 5792 <mss
1460,sackOK,timestamp 3564656007 110486255>
15:14:51.160576 IP 137.226.59.118.33646 > 130.104.72.201.5851: . ack 1 win 5840 <nop,nop,timestamp 110486260 3564656007>
15:14:51.160651 IP 137.226.59.118.33646 > 130.104.72.201.5851: P 1:151(150) ack 1 win 5840 <nop,nop,timestamp 110486260
3564656007>
15:14:51.189501 IP 130.104.72.201.5851 > 137.226.59.118.33646: . ack 151 win 5792 <nop,nop,timestamp 3564656034
110486260>
15:14:51.189557 IP 137.226.59.118.33646 > 130.104.72.201.5851: P 151:481(330) ack 1 win 5840 <nop,nop,timestamp 110486267
3564656034>
15:14:51.222324 IP 130.104.72.201.5851 > 137.226.59.118.33646: . ack 481 win 6432 <nop,nop,timestamp 3564656062
110486267>
15:14:51.364380 IP 130.104.72.201.5851 > 137.226.59.118.33646: P 1:400(399) ack 481 win 6432 <nop,nop,timestamp 3564656208
110486267>
15:14:51.364433 IP 137.226.59.118.33646 > 130.104.72.201.5851: . ack 400 win 6432 <nop,nop,timestamp 110486311
3564656208>
15:14:51.364459 IP 130.104.72.201.5851 > 137.226.59.118.33646: F 400:400(0) ack 481 win 6432 <nop,nop,timestamp 3564656208
110486267>
15:14:51.366094 IP 137.226.59.118.33646 > 130.104.72.201.5851: F 481:481(0) ack 401 win 6432 <nop,nop,timestamp 110486312
3564656208>
15:14:51.392833 IP 130.104.72.201.5851 > 137.226.59.118.33646: . ack 482 win 6432 <nop,nop,timestamp 3564656238
110486312>
```

↑ 16 packets, 2132 bytes ↓ 2 packets, 542 bytes

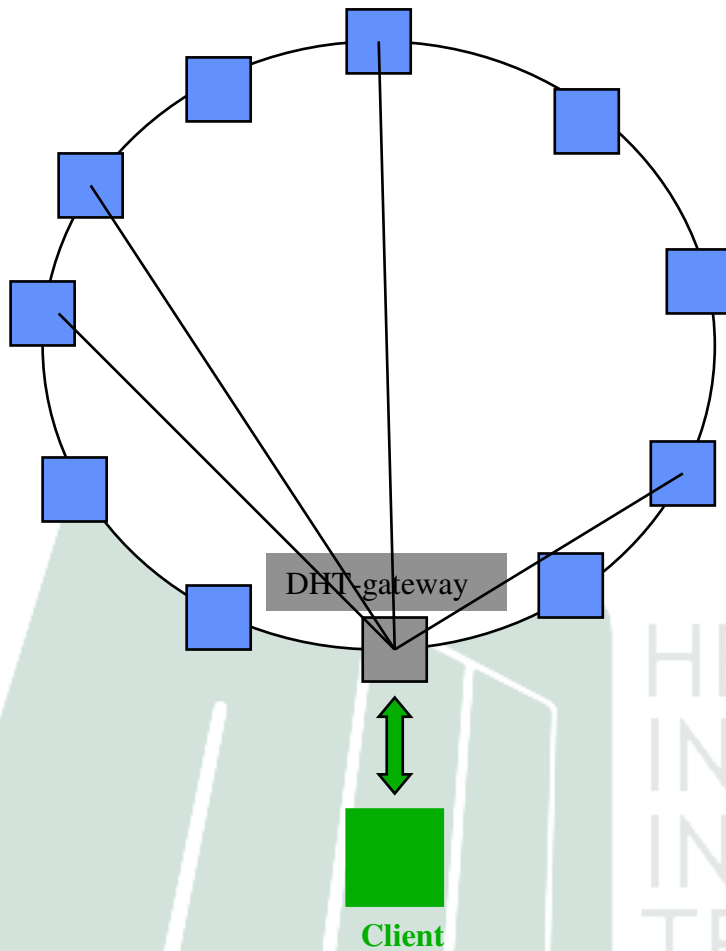
```
16:46:00.396623 IP 137.226.59.118.46613 > 137.226.12.31.domain: 36570+ A? qqqqqqq7.hit-to-ip.infracore.net. (49)
16:46:00.396749 IP 137.226.12.31.domain > 137.226.59.118.46613: 36570 1/0/0 (65)
```

Why DNS (1/2)?

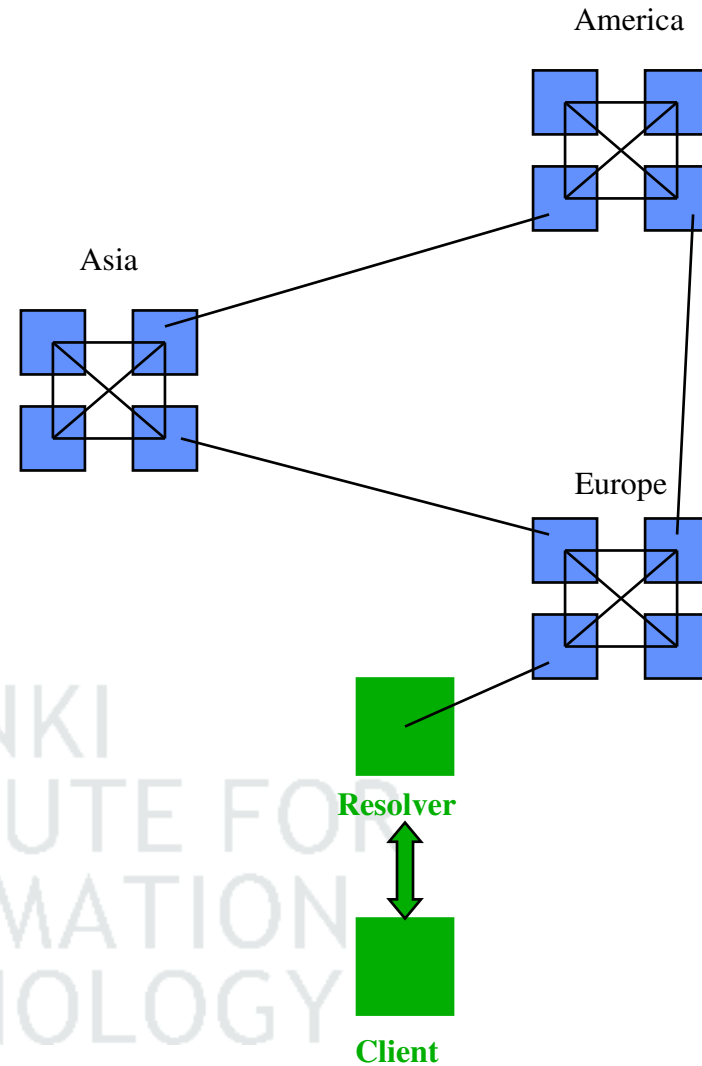
- Domain Name System is the most widely deployed distributed database. Let us embed HIT/IP mapping to this system
- Almost every client can access a recursive resolver in the same network. We may use existing dns servers instead of dht-gateways
- Simple UDP packets instead of XML-RPC requests
- And DNS is already used for OpenDHT boot-strapping

Why DNS (2/2)?

OpenDHT



DNS



HELSINKI
INSTITUTE FOR
INFORMATION
TECHNOLOGY

NSUPDATE (1/2)

- nsupdate submits Dynamic DNS update requests to a name server, as defined in RFC 2136
- Patch to HIPL hipd:
When the daemon starts or notices interface reconfiguration, it calls nsupdate for every HIT.

```
update delete 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net IN  
A
```

```
update delete 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net IN  
AAAA
```

```
update add 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net 60 IN  
AAAA 2001:470:1f00:ffff::1bb3
```

```
update add 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net 60 IN  
AAAA 2001:998:10:0:215:60ff:fe9f:60c4
```

```
update add 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net 60 IN  
A 193.167.187.1
```

```
send
```

NSUPDATE (2/2)

Using pre-defined keys we may update records, if the server is properly configured. For example:

```
server 2001:15:6099:97fa:1b0c:4322:fb26:7ea1
key hit-to-ip.infracorp.net Ousu<...>EQ==
update ...
update ...
send
```

```
key "hit-to-ip.infracorp.net." {
    algorithm hmac-md5;
    secret "Ousu<...>EQ";
};
zone "hit-to-ip.infracorp.net" IN {
    <...>
    update-policy { grant hit-to-ip.infracorp.hit subdomain hit-to-ip.infracorp.net. A AAAA; };
};
```

NAMED (1/3)

- Why don't we use Host Identifiers as keys?
- Patched ISC BIND: The nameserver allows to update the records corresponding to the HIT.
- In other words, the owner of 2001:15:6099:97fa:1b0c:4322:fb26:7ea1 is allowed to modify 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net



NAMED (2/3)

```
zone "hit-to-ip.infracorp.net" IN {  
    type master;  
    file "hit-to-ip.infracorp.net";  
    allow-query { any; };  
    allow-transfer { any; };  
    update-policy { grant *.hit self-reverse  
hit-to-ip.infracorp.net. A AAAA; };  
};
```

The query from HIT is equal to a query using key
1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit

NAMED (3/3)

Mar 13 17:04:19 stargazer named[16914]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36843:
updating zone 'hit-to-ip.infracorp.net/IN': deleting rrset at '1.a.e.7.6.2.b.f.2.2.3.4.c.0.
b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net' A

Mar 13 17:04:19 stargazer named[16914]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36843:
updating zone 'hit-to-ip.infracorp.net/IN': deleting rrset at '1.a.e.7.6.2.b.f.2.2.3.4.c.0.
b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net' AAAA

Mar 13 17:04:19 stargazer named[16914]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36843:
updating zone 'hit-to-ip.infracorp.net/IN': adding an RR at '1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.
1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net' AAAA

Mar 13 17:04:19 stargazer named[16914]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36843:
updating zone 'hit-to-ip.infracorp.net/IN': adding an RR at '1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.
1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net' AAAA

Mar 13 17:04:19 stargazer named[16914]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36843:
updating zone 'hit-to-ip.infracorp.net/IN': adding an RR at '1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.
1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net' A

HIT -> hostname (1/3)

HIP-aware hosts already try to resolve hostnames for HITs. They get NXDOMAIN from root-servers creating unnecessary load

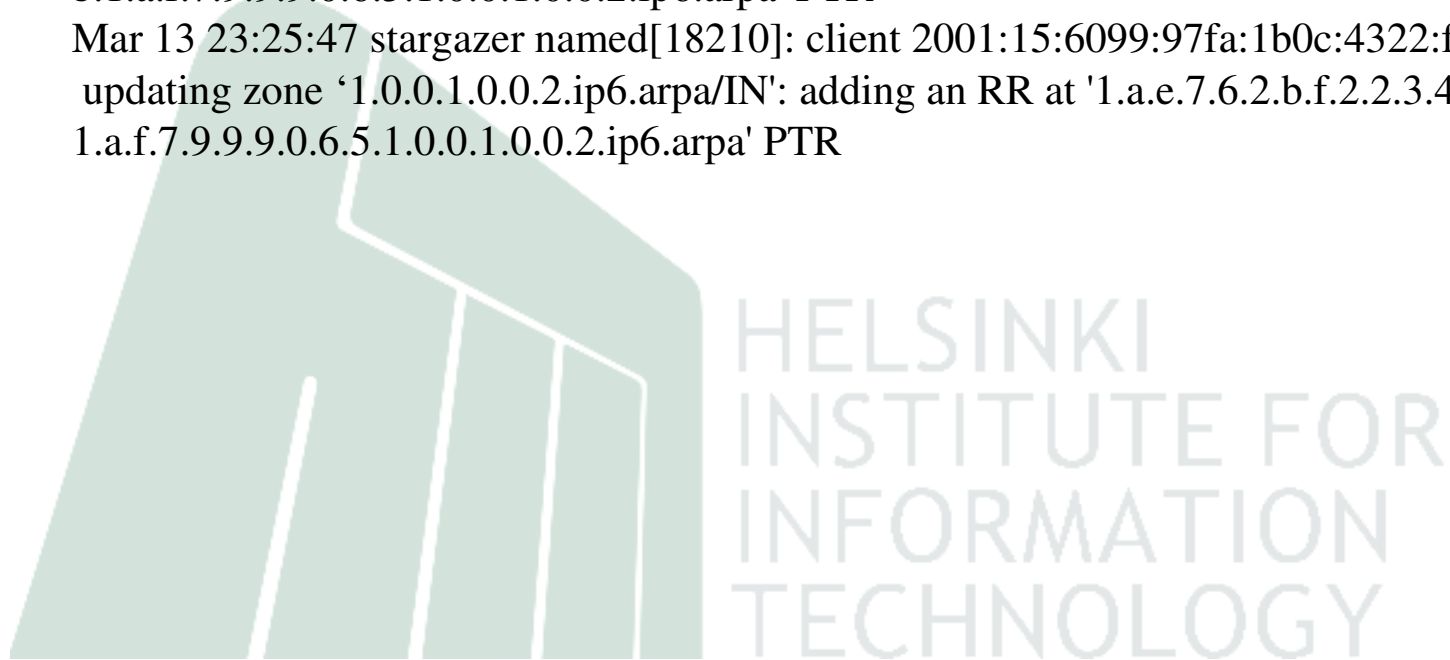
```
zone "1.0.0.1.0.0.2.ip6.arpa" IN {  
    type master;  
    file "1.0.0.1.0.0.2.ip6.arpa";  
    allow-query { any; };  
    allow-transfer { any; };  
    update-policy { grant *.hit self-reverse ip6.arpa. PTR; };  
};
```

```
1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa.  
    IN PTR stargazer-hit.pc.infracorp.net.
```

HIT -> hostname (2/3)

```
server 2001:15:6099:97fa:1b0c:4322:fb26:7ea1
update delete 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa. IN PTR
update add 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa. 60 IN PTR
stargazer-hit.pc.infrahip.net.
send
```

```
Mar 13 23:25:47 stargazer named[18210]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36872:
  updating zone '1.0.0.1.0.0.2.ip6.arpa/IN': deleting rrset at '1.a.e.7.6.2.b.f.2.2.3.4.c.0.
  b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa' PTR
Mar 13 23:25:47 stargazer named[18210]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36872:
  updating zone '1.0.0.1.0.0.2.ip6.arpa/IN': adding an RR at '1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.
  1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa' PTR
```



HIT -> hostname (3/3)

```
server 2001:15:6099:97fa:1b0c:4322:fb26:7ea1
```

```
update delete 2.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa. IN PTR
```

```
update add 2.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa. 60 IN PTR
```

```
mallory.infracore.net.
```

```
send
```

```
Mar 14 00:18:56 stargazer named[20291]: client 2001:15:6099:97fa:1b0c:4322:fb26:7ea1#36878:
```

```
updating zone '1.0.0.1.0.0.2.ip6.arpa/IN': update failed: rejected by secure update (REFUSED)
```



Practical Implementation (1/3)

- TWO levels:

This can be distributed and does not change so often

; TTL 1d

1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.arpa

IN CNAME 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net.

This is provided by user's company/ISP/public service

; TTL 15

1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-ip.infracorp.net.

IN A 193.167.187.1

IN AAAA 2001:470:1f00:ffff::1bb3

IN AAAA 2001:998:10:0:215:60ff:fe9f:60c4

Practical Implementation (2/3)

; TTL 1d

1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.ip6.arpa.

IN CNAME 1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-hostname.infracorp.net.

; TTL 15

1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.1.0.0.1.0.0.2.hit-to-hostname.infracorp.net.

IN PTR stargazer-hit.pc.infracorp.net.



Some estimations

100 (key) + 28 (value) bits for each HIT

$6 * 10^9$ HITs will require ~90Gb

(or probably ~50Gb when compressed)

One update per hour, 100 bytes for request

~1300 Mbps in total, ~1.700.000 updates / second

If 32 bytes per update (better protocol for distribution)

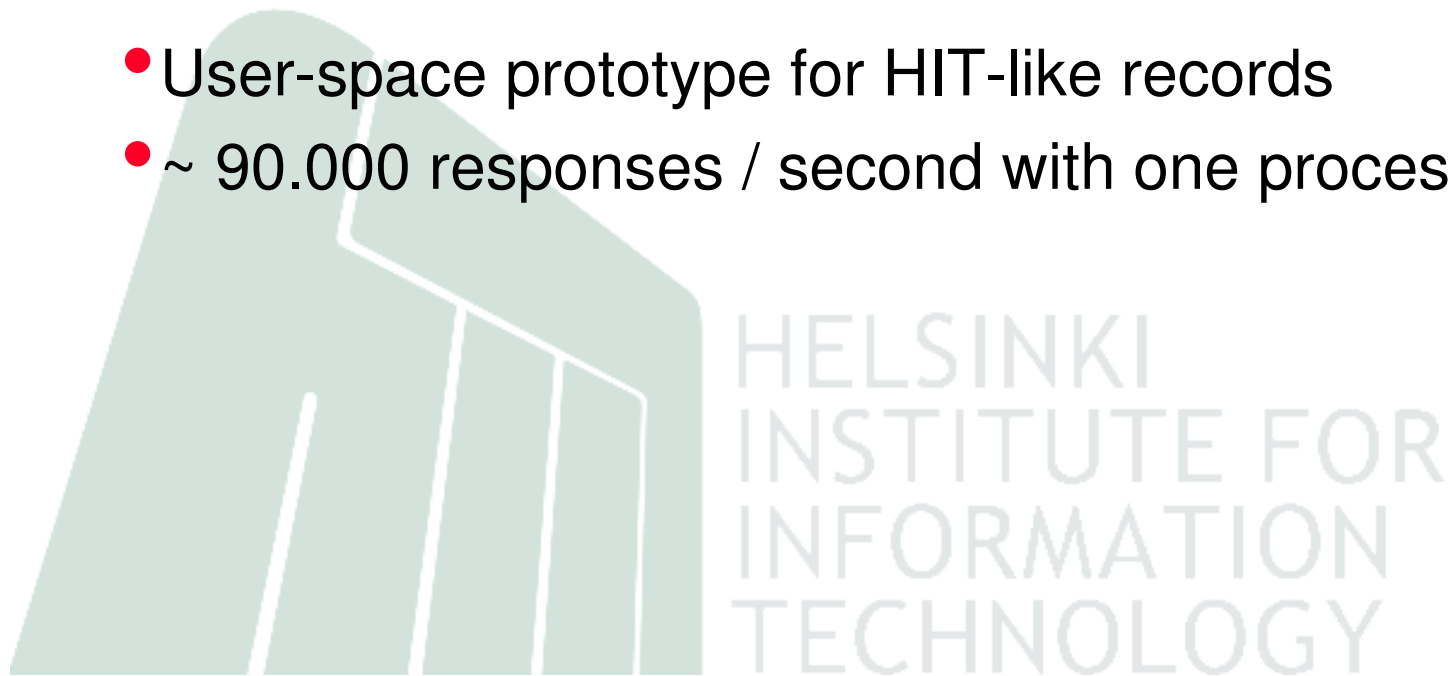
~400 Mbps

Practical Implementation (3/3)

- Operate similar to distributed root-servers, use IP Anycast (RFC 1546)
- Clients' query is automatically routed to the nearest server
- At least two different implementations (with compatible exchange protocol) running on different hardware
- Cluster of 10 computers (with 8 Gb RAM) serving one IP-address – about 10 kEUR in 2007?

BIND performance

- DNS queries to BIND9 on felwood (HP G5 2 x 2 Xeon 2Gz, 4 working threads) sent from multiple clients over ethernet
- ~ 25.000 responses / second
- User-space prototype for HIT-like records
- ~ 90.000 responses / second with one process



Questions, ideas?

THE END

<mailto:oleg.ponomarev@hiit.fi>

HELSINKI
INSTITUTE FOR
INFORMATION
TECHNOLOGY