

# Using DNS for Mapping Host Identifiers to Locators

Oleg Ponomarev

28 July 2009

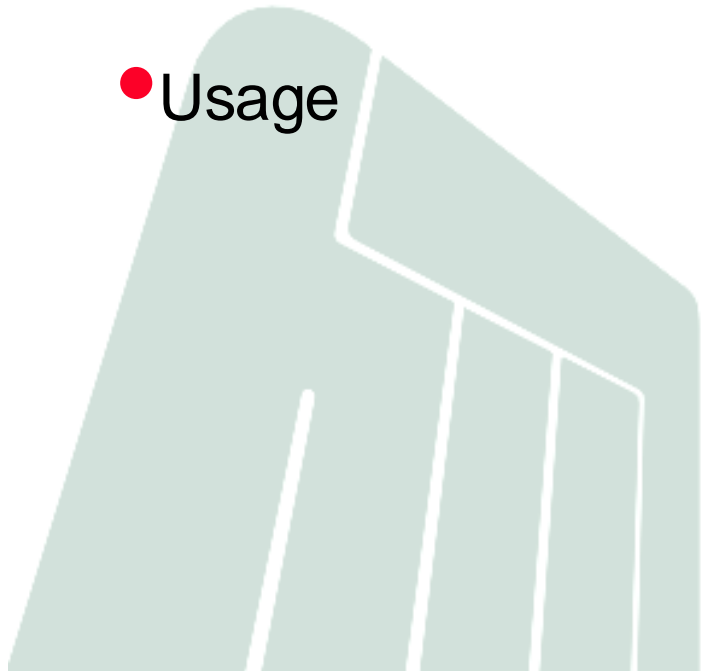
IETF75, Stockholm

The logo consists of a stylized, light green graphic on the left, resembling a series of parallel lines or a building facade, and the text 'HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY' on the right, stacked in four lines.

HELSINKI  
INSTITUTE FOR  
INFORMATION  
TECHNOLOGY

# OUTLINE

- Current situation
- Storage conventions
- Usage



HELSINKI  
INSTITUTE FOR  
INFORMATION  
TECHNOLOGY

# NOW

DNS Server



EXAMPLE.COM. HIP?  
EXAMPLE.COM. HIP 2001...5678  
EXAMPLE.COM. AAAA?  
EXAMPLE.COM. AAAA Ø  
EXAMPLE.COM. A?  
EXAMPLE.COM. A 192.0.2.1

HIP Software

OpenHIP



HIT	IP
2001...5678	192.0.2.1

EXAMPLE.COM.  
AAAA 2001...5678

EXAMPLE.COM. AAAA?

sendto(2001...5678)

Legacy Application



# OK, BUT

- What if the application does not use DNS?
- What if the application stores IP addresses internally?
- HIT to IP global mapping database?
- OpenDHT: 16 packets, 2132 bytes – **SLOW!**

# draft-ponomarev-hip-hit2ip-03

- ~~2006:504:1234:5678:9A0BCD:EF08:7234:5678~~.1.0.  
1.0.0.1.0.0.2.HIT-TO-IP.EXAMPLE.

**A/AAAA** IP ADDRESS(ES)

**PTR** HOSTNAME(S)

**CNAME** LINK TO ANOTHER DOMAIN

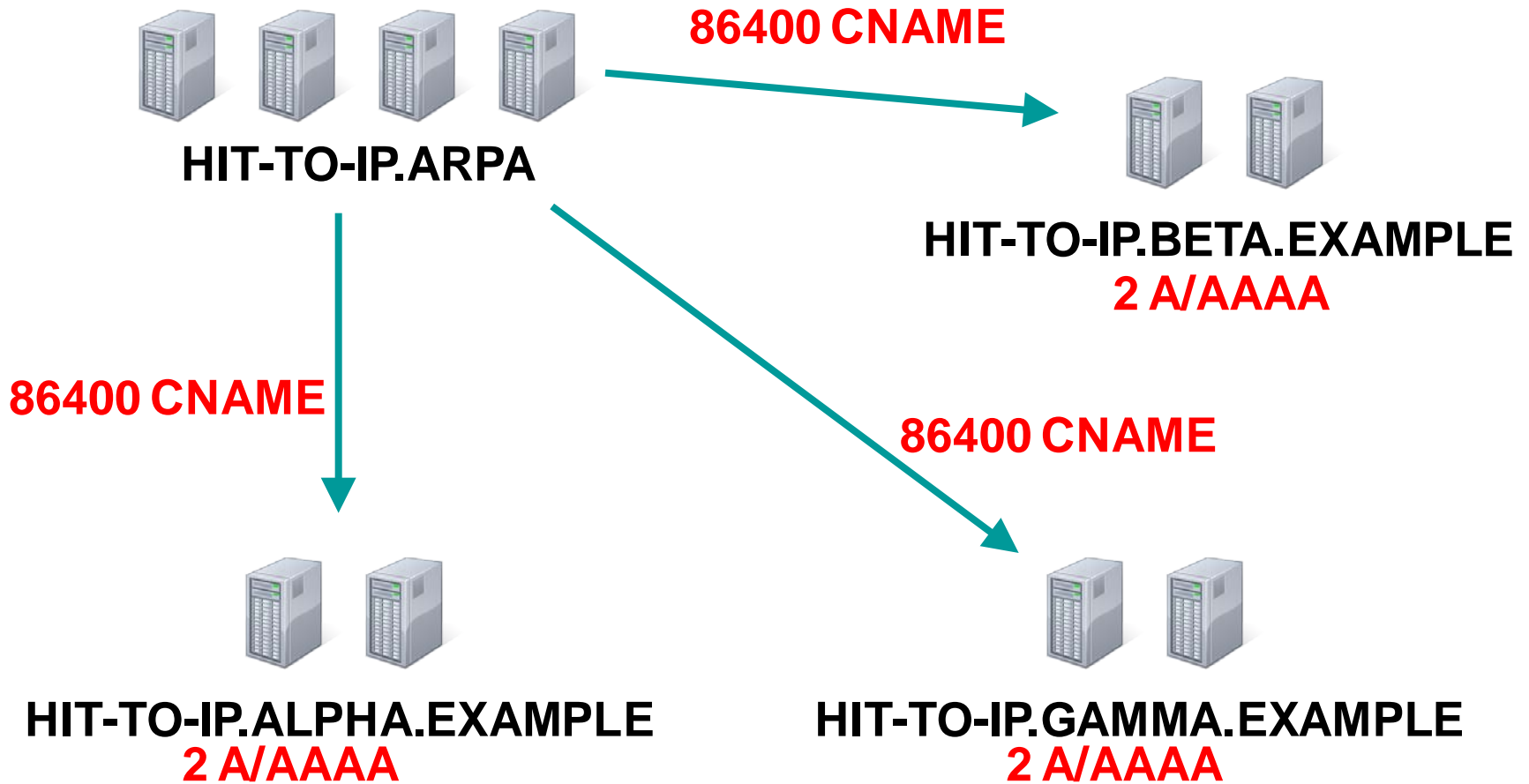
**HIP** COMPLETE HOST IDENTITY

- DNS is just an access interface
- Much experience, re-use existing resolvers

# UPDATES

- DNS UPDATE authenticated by HIP, only *2001:10:1234:5678:9ABC:DEF0:1234:5678* may change *8.7.6.5.4.3.2.1.0.F.E.D.C.B.A.9.8.7.6.5.4.3.2.1.0.1.0.0.1.0.0.2.HIT-TO-IP.*
- Stress-**tested** OpenHIP – 110 base exchanges / second on a single CPU
- BIND9 – 100.000 replies / second
- Enough for 100.000 active clients, assuming 15 minute update interval – **OK in local scale**

# TWO LEVELS



# SUMMARY

- Deployment is important
- Legacy applications must work
- Global mapping can be done



HELSINKI  
INSTITUTE FOR  
INFORMATION  
TECHNOLOGY



# BACKUP: SOME NUMBERS

- Root level: 100 bit (HIT) + 28 bit (index) = 16 bytes
- 32GB RAM (\$1500 server) – **two billion** identifiers
- 40 servers (40U) – **40 billion** identifiers (redundantly)
- Indirection for HIT-TO-IP.ARPA
- The same second level index for 1.0.0.1.0.0.2.IP6.ARPA

# BACKUP: TWO LEVELS

8.7.6.5.4.3.2.1.0.F.E.D.C.B.A.9.8.7.6.5.4.3.2.1.0.1.0.0.1.0.0.2.HIT-TO-IP.ARPA.

**86400 CNAME** 8.7.6.5.4.3.2.1.0.F.E.D.C.B.A.9.8.7.6.5.4.3.2.1.0.1.0.0.1.0.0.2.HIT-TO-IP.EXAMPLE.NET.

8.7.6.5.4.3.2.1.0.F.E.D.C.B.A.9.8.7.6.5.4.3.2.1.0.1.0.0.1.0.0.2.HIT-TO-IP.EXAMPLE.NET.

**2 A** 192.0.2.1

**2 AAAA** 2001:DB8::1

# BACKUP: LOCAL USAGE

- Application: EXAMPLE.COM. AAAA?
- EXAMPLE.COM.      HIP {2001...5678}  
                            A      192.0.2.1
- Local DNS server: EXAMPLE.COM. AAAA 2001...5678
- Add 8.7.6.5...1.0.0.2.HIT-TO-IP.EXAMPLE.NET.  
                            CNAME EXAMPLE.COM.

# BACKUP: CNAME HIT-TO-IP

- EXAMPLE.COM. CNAME  
8.7.6.5...1.0.0.2.HIT-TO-IP.EXAMPLE.
- 8.7.6.5...1.0.0.2.HIT-TO-IP.EXAMPLE.  
HIP {2001...5678}  
A 192.0.2.1



# BACKUP: LSI

- Application: EXAMPLE.COM. A?
- EXAMPLE.COM.      HIP {2001...5678}  
                          A      192.0.2.1
- Local DNS server: EXAMPLE.COM. A 1.7.8.9
- Add 9.8.7.1.LSI-TO-IP.EXAMPLE.NET.  
                          CNAME EXAMPLE.COM.

# BACKUP: IP6.ARPA

1.0.0.1.0.0.2.IP6.ARPA.

**86400 NS** A.HIP-SERVERS.NET.

**86400 NS** B.HIP-SERVERS.NET.

**86400 NS** C.HIP-SERVERS.NET.

8.7.6.5.4.3.2.1.0.F.E.D.C.B.A.9.8.7.6.5.4.3.2.1.0.1.0.0.1.0.0.2.IP6.ARPA.

**86400 CNAME**

8.7.6.5.4.3.2.1.0.F.E.D.C.B.A.9.8.7.6.5.4.3.2.1.0.1.0.0.1.0.0.2.HIT-TO-HOST.EXAMPLE.NET.

8.7.6.5.4.3.2.1.0.F.E.D.C.B.A.9.8.7.6.5.4.3.2.1.0.1.0.0.1.0.0.2.HIT-TO-HOST.EXAMPLE.NET.

**86400 PTR** EXAMPLE.COM.

# BACKUP: OpenDHT vs DNS

```
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?
0.8.9.b.4.0.1.1.1.8.5.0.7.8.7.3.5.d.f.6.1.f.c.4.1.1.0.0.1.0.0.2..hit-to-ip.net. (49)
16:46:00.396749 IP 137.226.12.31.domain > 137.226.59.118.46613: 36570 1/0/0 (65)
```

# LOCATOR in nat-traversal

