State of the Art in IPv6 Attack & Defense

Fernando Gont

SI6 NETWORKS

IT-DEFENSE 2015
Leipzig, Germany. February 4-6, 2015

About...

- Security Researcher and Consultant at SI6 Networks
- Published:
  - 20 IETF RFCs (9 on IPv6)
  - 10+ active IETF Internet-Drafts
- Author of the SI6 Networks' IPv6 toolkit
  - http://www.si6networks.com/tools/ipv6toolkit
- I have worked on security assessment of communication protocols for:
  - UK NISCC (National Infrastructure Security Co-ordination Centre)
  - UK CPNI (Centre for the Protection of National Infrastructure)
- More information at: http://www.gont.com.ar

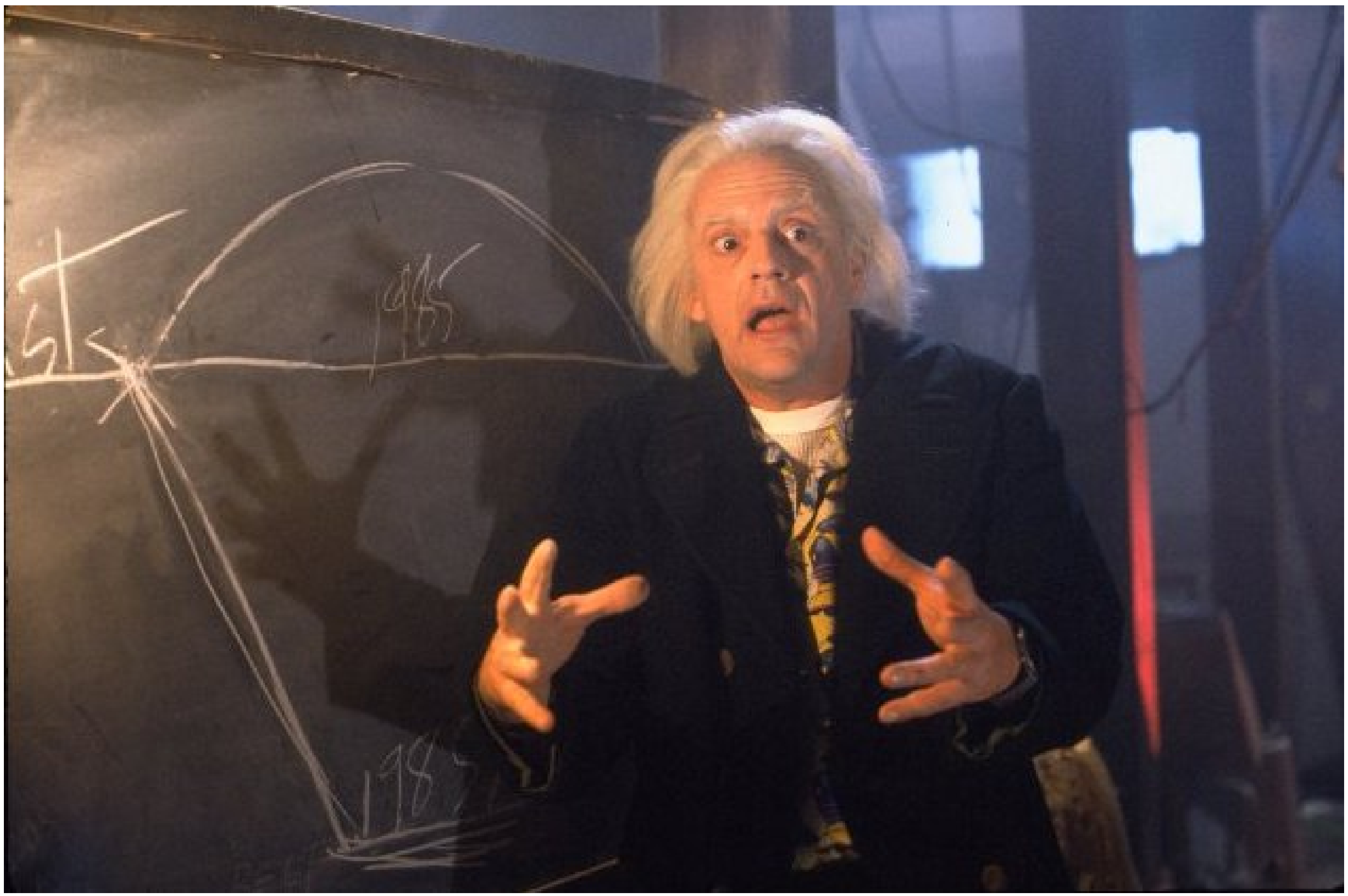
Motivation for this presentation

Motivation

- TCP & IPv4 were introduced in the early '80's
- Yet in the late '90s (and later!) we were still addressing security issues
  - SYN flood attacks
  - Predictable TCP Initial Sequence Numbers (ISNs)
  - Predictable transport protocol ephemeral port numbers
  - IPv4 source routing
  - etc.
- Mitigations typically researched **after** exploitation
- Patches applied on production systems

Motivation (II)

- We hope to produce an alternative future for IPv6



IPv6's Main Security Problem

IPv6's main security problem

IPv6's main security problem (II)

- Lots of myths about IPv6 security going around for ages
- Everyone has something to say
  - It's free! ;-)
- People just propagate whatever they hear
  - ~~Information era~~ -> **Communications** era?
- Outcome:
  - **You** get confused
  - The field doesn't advance much

Disclaimer

Lots of stuff

+

60' talk

=

Fasten your

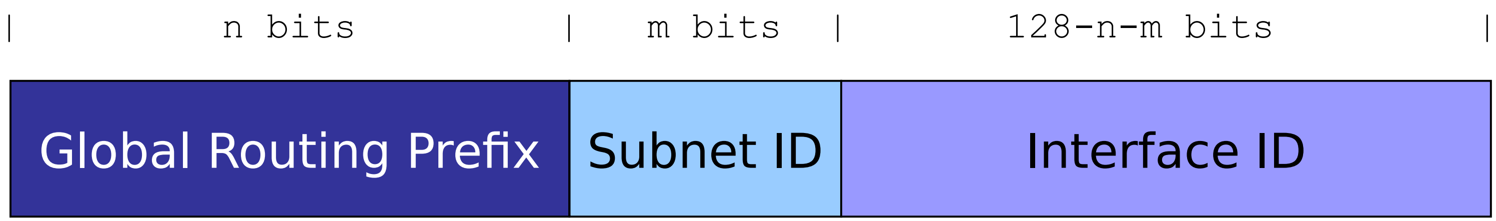
Seatbelts!

Part I: Standardization Efforts

IPv6 Addressing

Brief overview

IPv6 Global Unicast Addresses



- A number of possibilities for generating the Interface ID:
  - Embed the MAC address (traditional SLAAC)
  - Embed the IPv4 address (e.g. 2001:db8::192.168.1.1)
  - Low-byte (e.g. 2001:db8::1, 2001:db8::2, etc.)
  - Wordy (e.g. 2001:db8::dead:beef)
  - According to a transition/co-existence technology (6to4, etc.)

IPv6 Addressing

Overview of Security/Privacy Implications

Security Implications of IPv6 Addressing

- **Correlation of network activity over time**
  - 'cause the IID does not change over time
- **Correlation of network activity across networks**
  - 'cause the IID does not change across networks
  - e.g. 2001:db8::**1234:5678:90ab:cdef** vs. fc00:1::**1234:5678:90ab:cdef**
- **Network reconnaissance**
  - 'cause the IIDs are predictable
  - e.g. 2001:db8**::1**, 2001:db8**::2**, etc.
- **Device specific attacks**
  - 'cause the IID leaks out the NIC vendor
  - e.g. 2001:db8::**fad1:11**ff:fec0:fb33 -> Atheros

IPv6 Addressing
Network Reconnaissance Myths and Reality

Introduction

"Thanks to the increased IPv6 address space, IPv6 host scanning attacks are unfeasible. Scanning a /64 would take 500.000.000 years"

– Urban legend

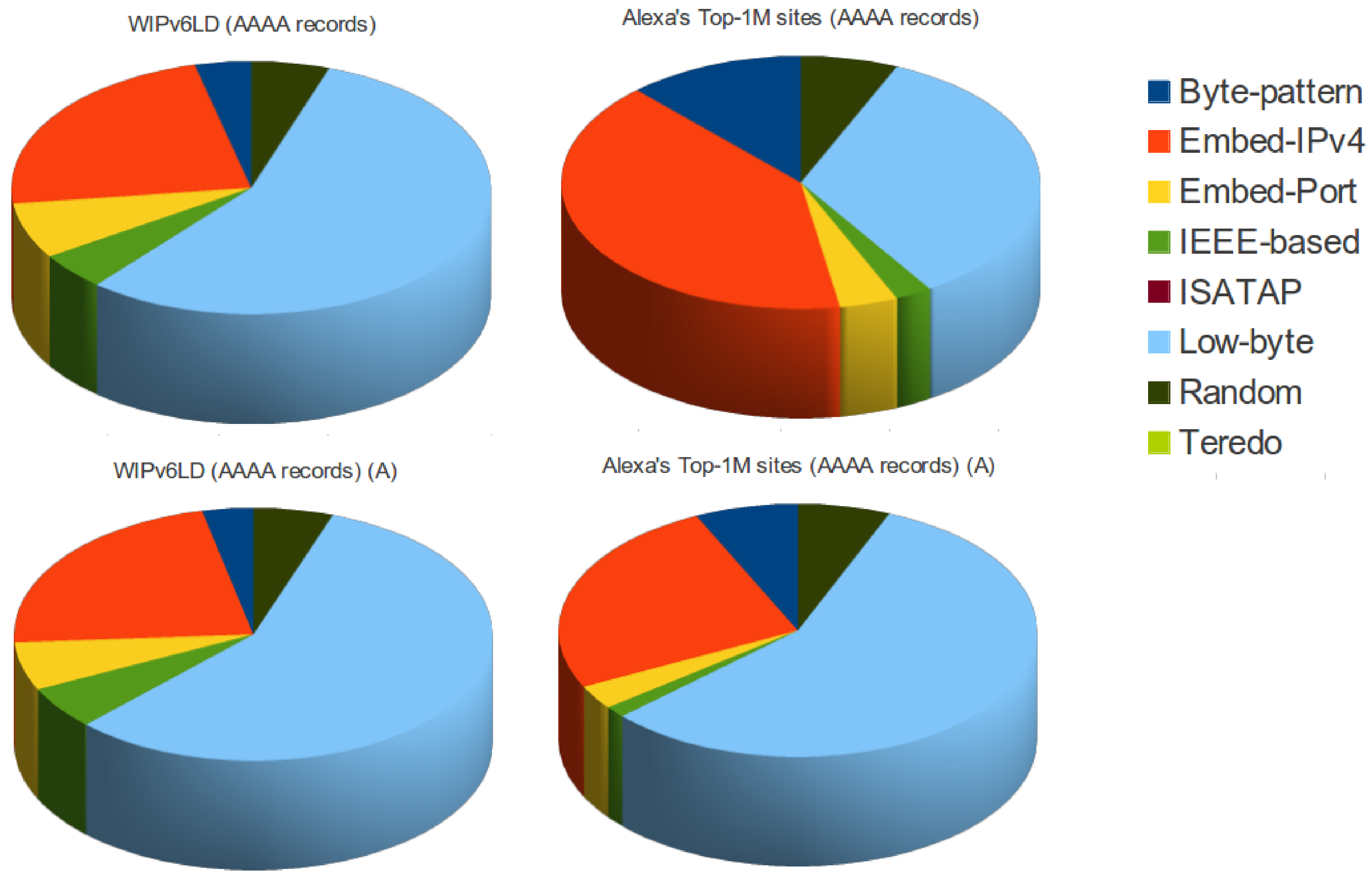
Is the search space for a /64 really 2^{64} addresses?

Short answer: No! (see: draft-ietf-opsec-ipv6-host-scanning)

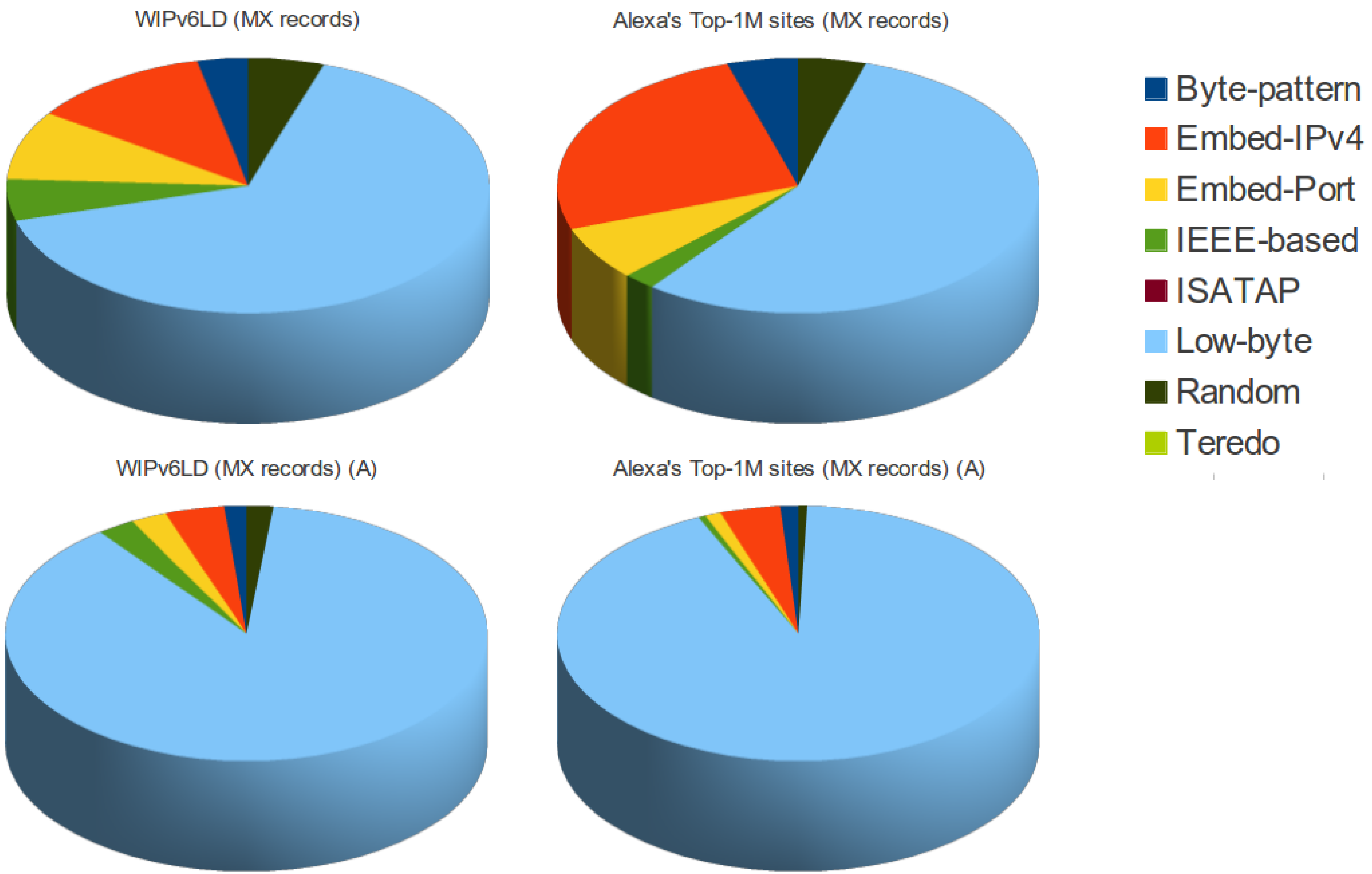
Our experiment

- Find “a considerable number of IPv6 nodes” for address analysis:
  - Alexa Top-1M sites + perl script + dig
  - World IPv6 Launch Day site + perl script + dig
- For each domain:
  - AAAA records
  - NS records -> AAAA records
  - MX records -> AAAA records
- What did we find?

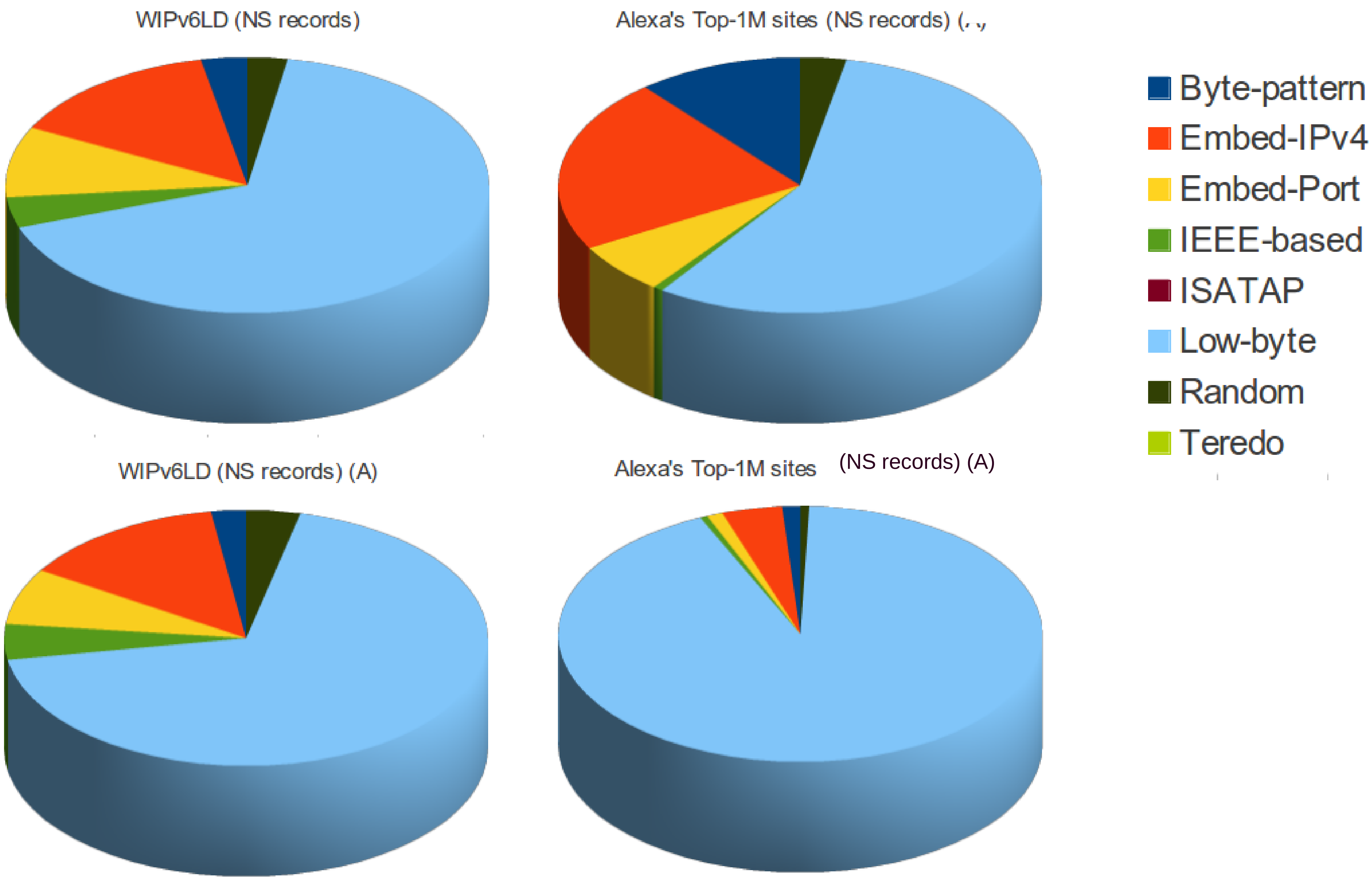
IPv6 address distribution for the web



IPv6 address distribution for MXs



IPv6 address distribution for the DNS



IPv6 Addressing
Mitigation of Security/Privacy Issues

Temporary Addresses (RFC4941)

- RFC 4941: privacy/temporary addresses
  - Random IIDs that change over time
  - Generated **in addition** to traditional SLAAC addresses
  - Traditional addresses used for server-like communications, temporary addresses for client-like communications
- Operational problems:
  - Difficult to manage!
- Security problems:
  - They do not fully replace the traditional SLAAC addresses (hende host-tracking is **only partially mitigated**)
  - They **do not** mitigate host-scanning attacks

SLAAC stable-privacy (RFC7217)

- Generate Interface IDs as:

  F(Prefix, Net_Iface, Network_ID, Counter, Secret_Key)

- Where:
  - F() is a PRF (e.g., a hash function)
  - Prefix is the SLAAC or link-local prefix
  - Net_Iface is some interface identifier
  - Network_ID could be e.g. the SSID of a wireless network
  - Counter is used to resolve collissions
  - Secret_Key is unknown to the attacker (and randomly generated by default)

SLAAC stable-privacy (RFC7217) (II)

- As a host moves:
  - Prefix and Network_ID change from one network to another
  - But they remain constant within each network
  - F() varies across networks, but remains constant within each network
- This results in addresses that:
  - Are stable within the same subnet
  - Have different Interface-IDs when moving across networks
  - For the most part, they have “the best of both worlds”
- There is a FreeBSD implementation
- A Linux implementation is in the works

DHCPv6's draft-ietf-dhc-stable-privacy

- Generate Interface IDs as:

  F(Prefix | Client_DUID | IAID | Counter | secret_key)

- Where:
  - F() is a PRF (e.g., a hash function)
  - Prefix: Represents the managed IPv6 address pool
  - Client_DUID is the Client's DHCPv6 DUID
  - IAID is a unique identifier for this address association
  - Counter is employed to resolve collisions
  - Secret_Key is unknown to the attacker (and randomly generated by default)

DHCPv6's draft-ietf-dhc-stable-privacy (II)

- Allows for multiple DHCPv6 servers to operate within the same subnet
- State about address leases is shared “algorithmically”
  - No need for a new protocol
- Even if the DHCPv6 lease file gets lost/corrupted, addresses will be stable

Other IETF work in this area

- draft-ietf-6man-ipv6-address-generation-privacy
  - Discusses the security implications of IPv6 addressing
- draft-ietf-6man-default-iids
  - Notes that implementations should default to RFC7217

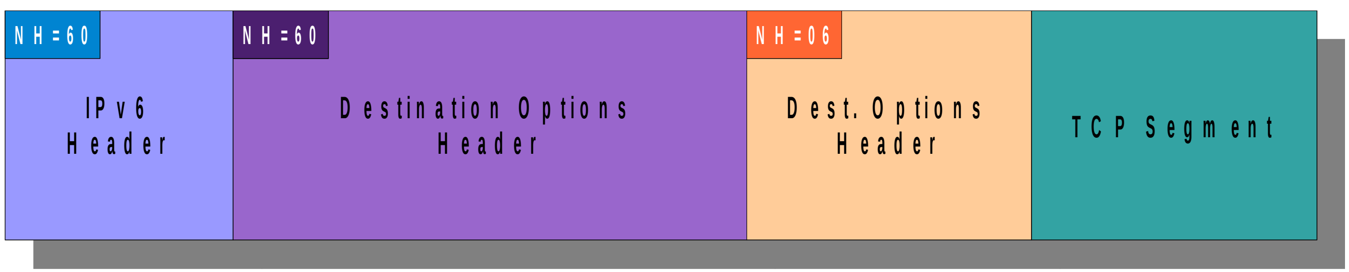
IPv6 Extension Headers

IPv6 Extension Headers

Overview

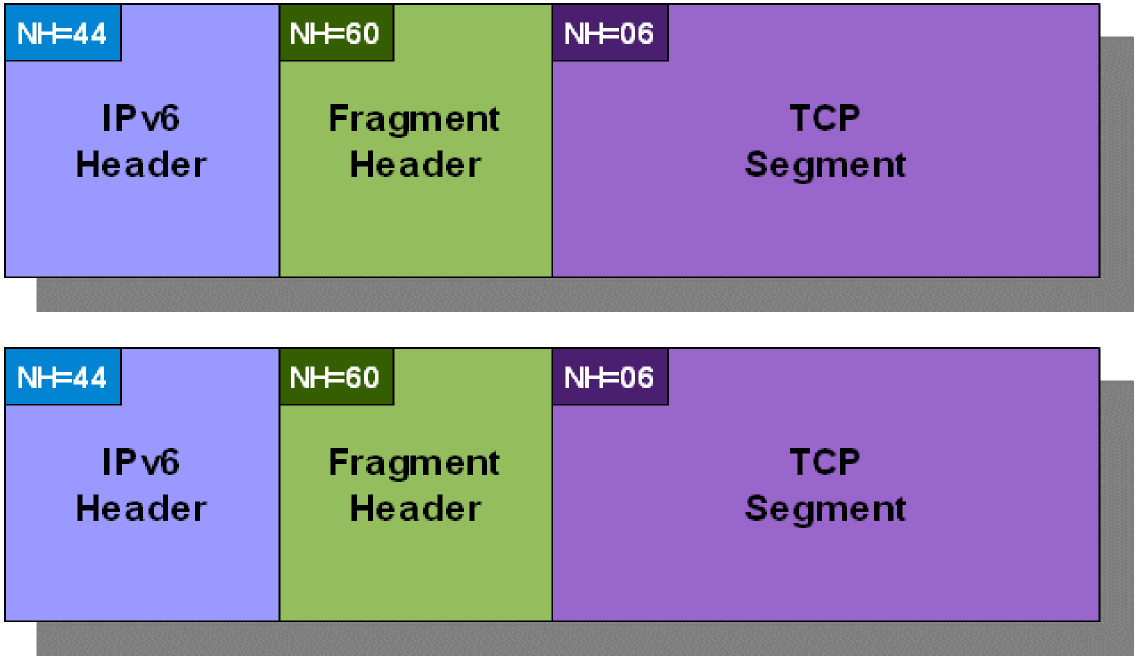
IPv6 Extension Headers

- Fixed-length base header
- Options conveyed in different types of Extension Headers
- Extension Headers organized as a daisy-chain structure



IPv6 Fragmentation

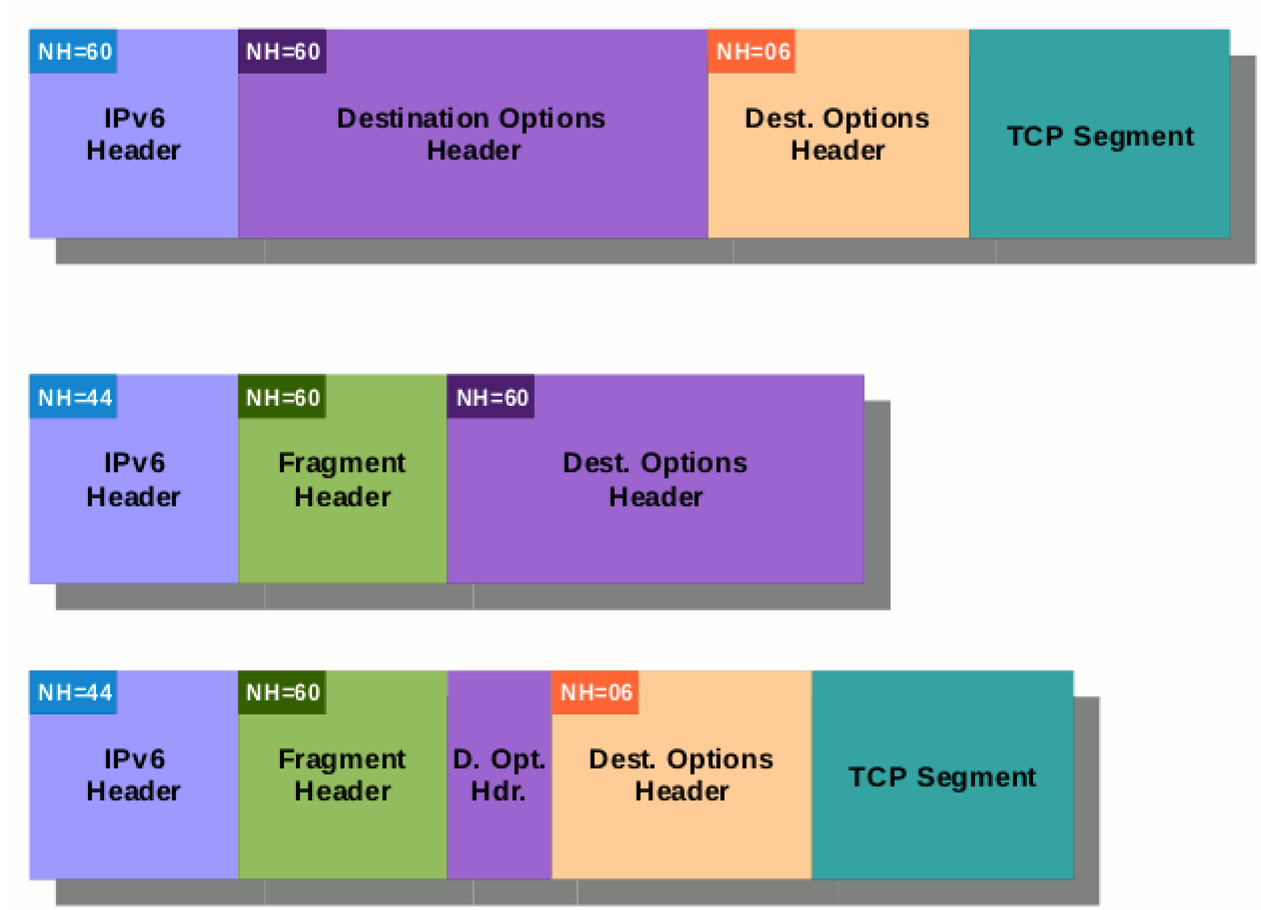
- Conceptually, same as in IPv4
- Implemented with an IPv6 Fragmentation Header



IPv6 Extension Headers
Reality

Finding Upper-layer information

- Finding upper-layer information is painful (if at all possible)



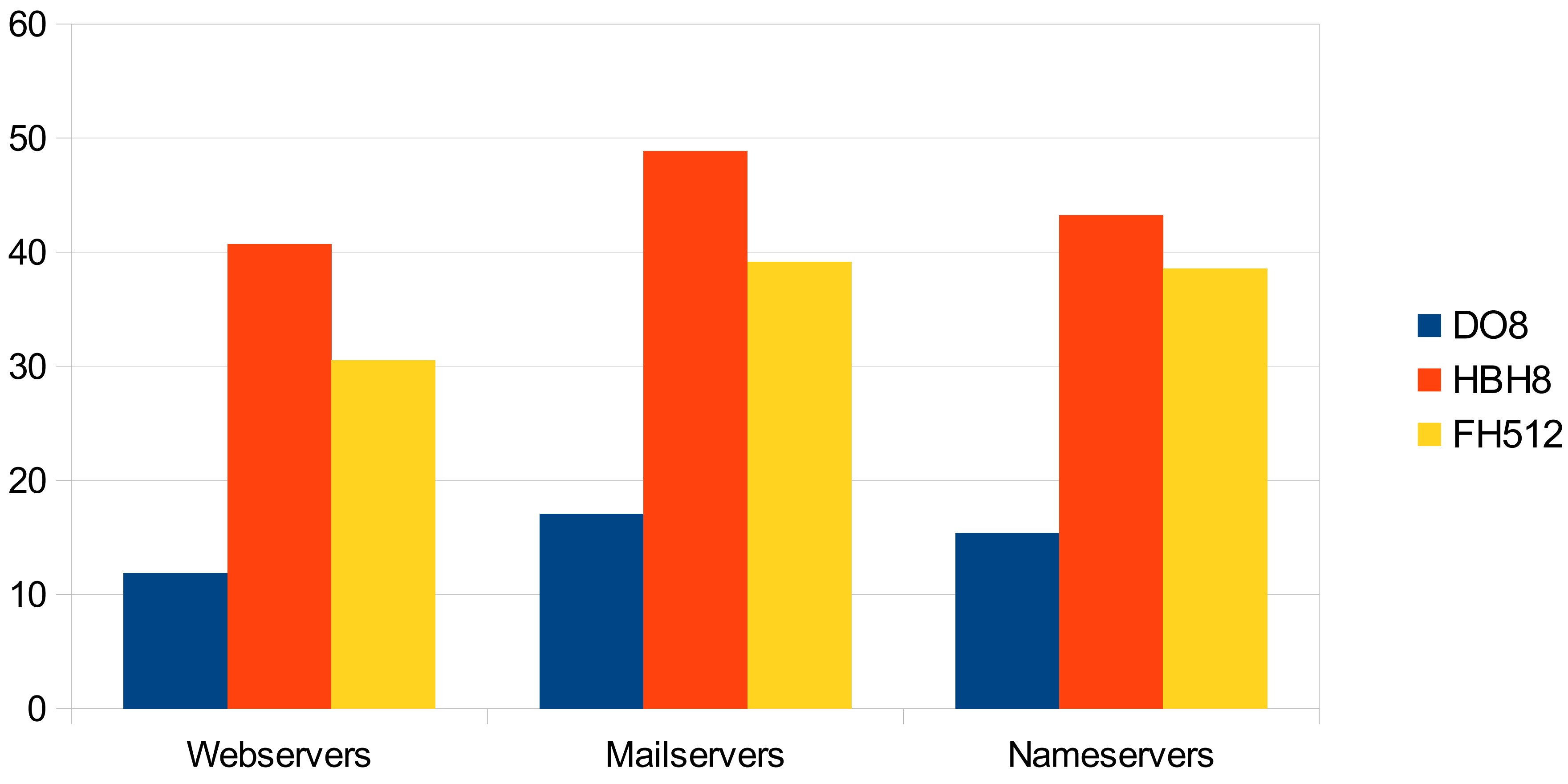
Processing the IPv6 header chain

- Processing the IPv6 header chain is expensive
  - May be CPU-intensive
  - Some implementations can inspect only up to 128 bytes (or even some smaller number)
- IPv6 fragmentation deemed as insecure
  - DoS vector
  - Evasion
  - Buggy implementations

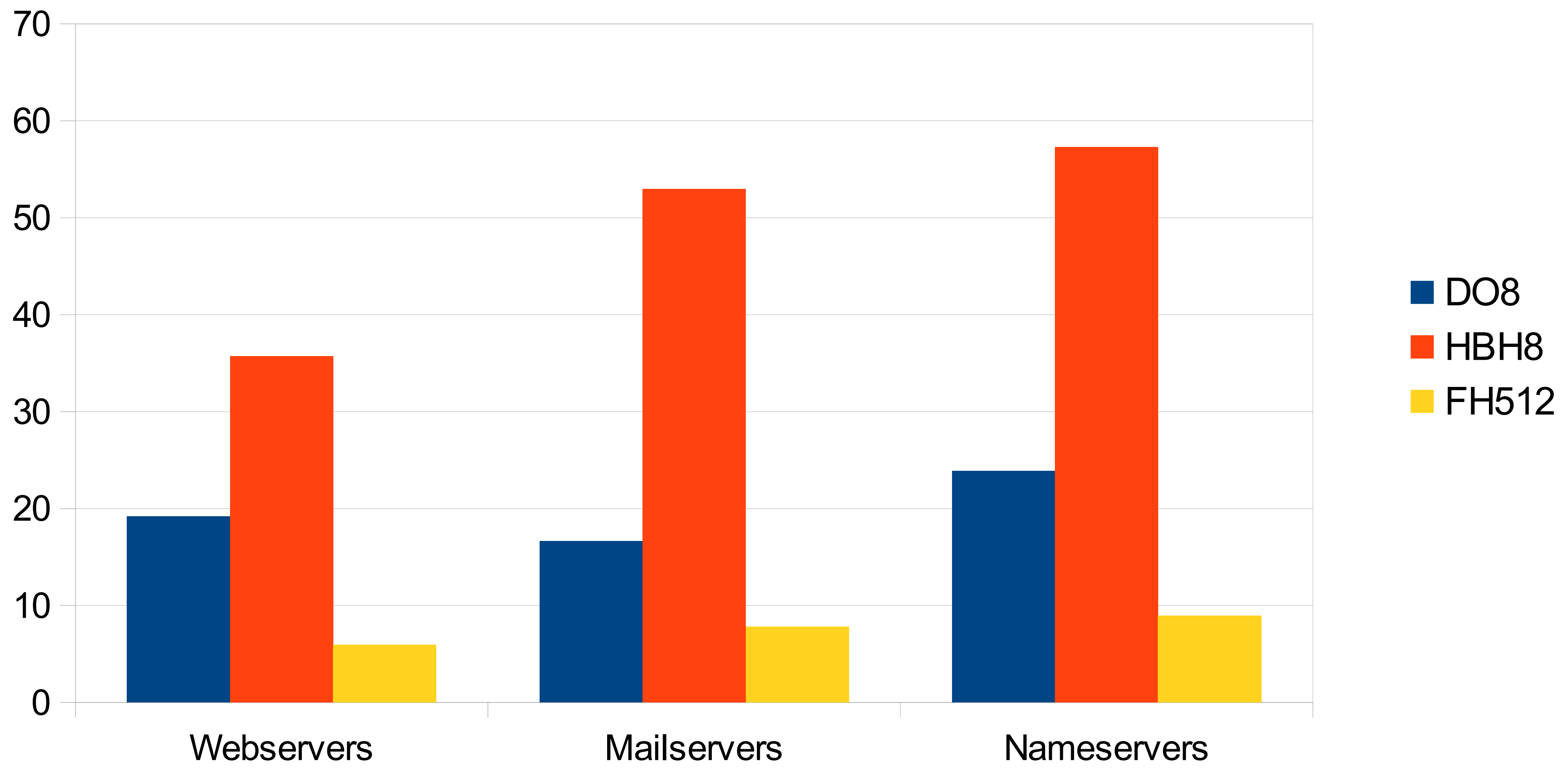
IPv6 EHs in the Real World

- Many operators allegedly filter them, as a result of:
  - Perceived issues with IPv6 Fragmentation and EH
  - Almost no current dependence on them
- But there was no real data...
- ... so we measured the IPv6 Internet ourselves

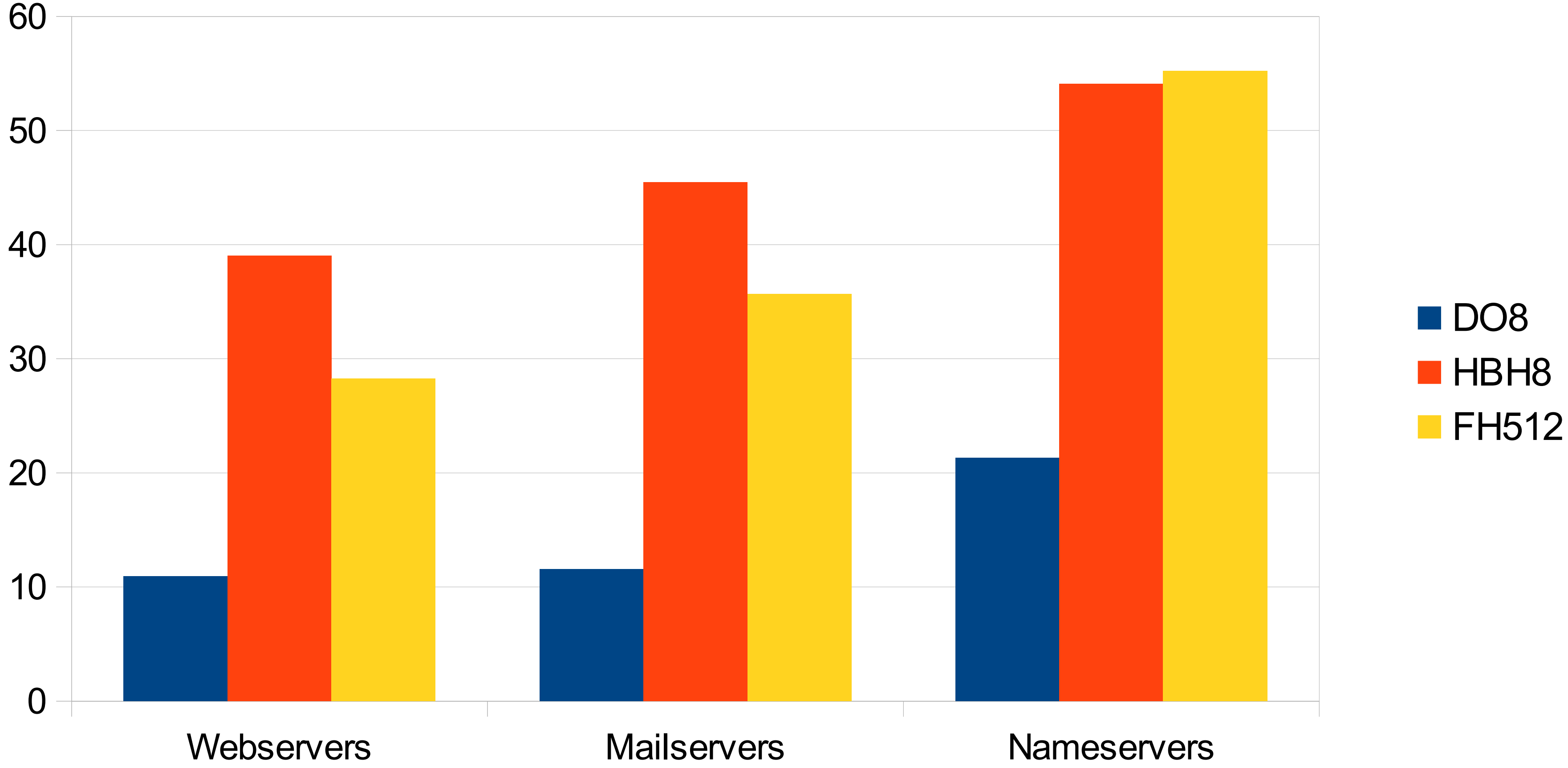
WIPv6LD dataset: Packet Drop rate



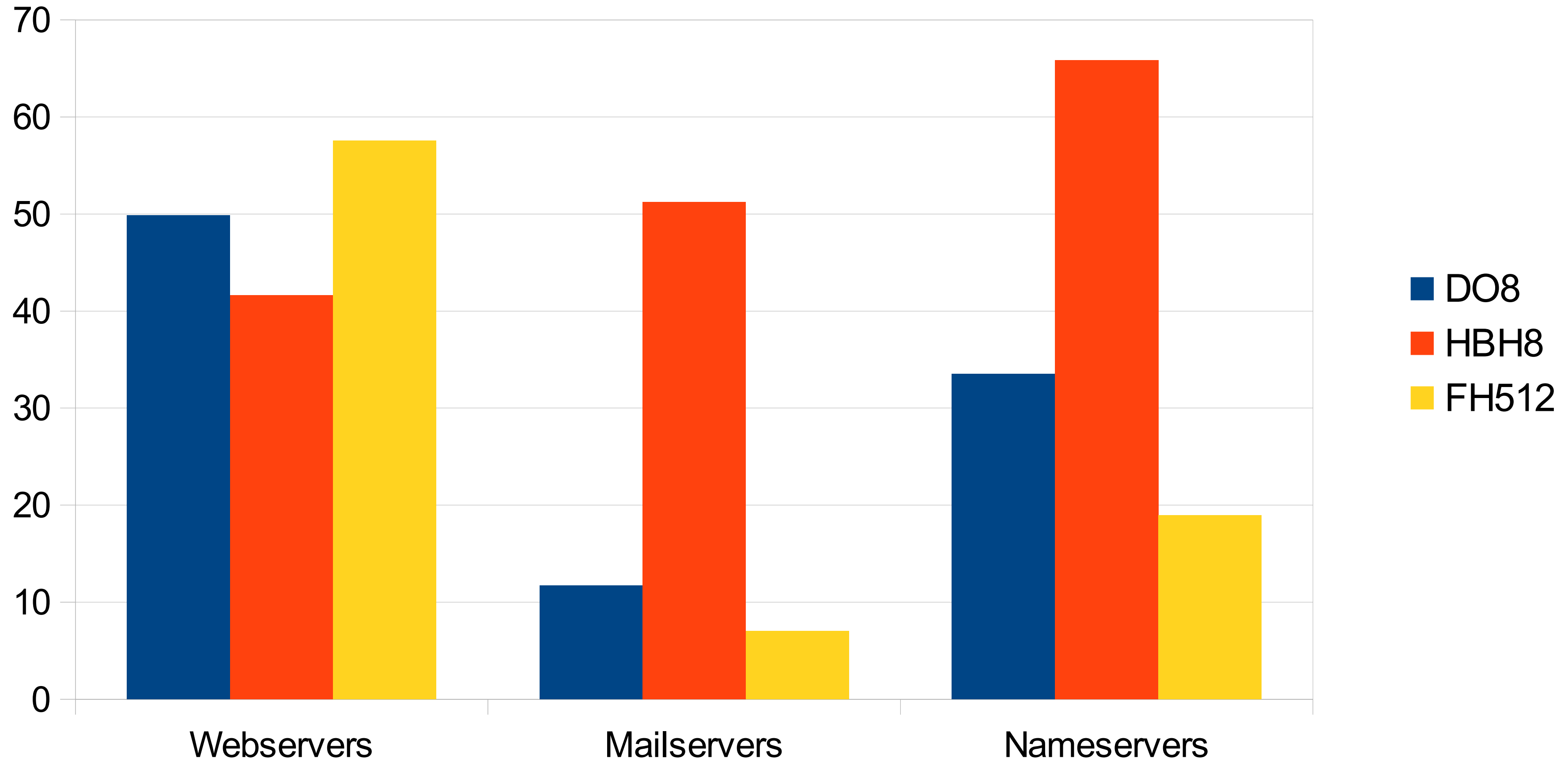
WIPv6LD dataset: Drops by diff. AS



Alexa dataset: Packet Drop rate



Alexa dataset: Drops by diff. AS



So... what does this all mean?

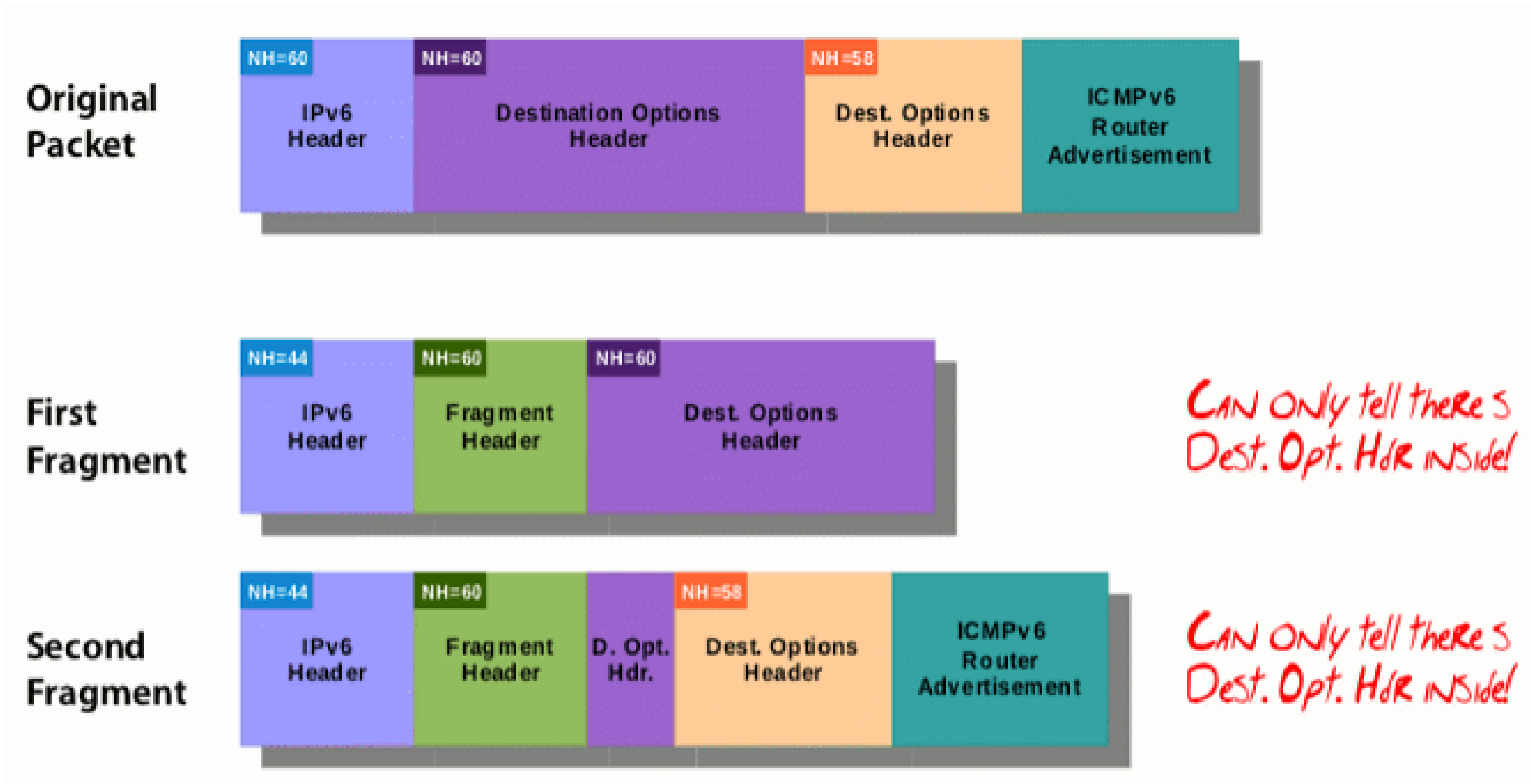
- Good luck with getting IPv6 EHs working in the public Internet!
  - They are widely dropped
- IPv6 EHs “not that cool” for evasion, either
  - Chances are that you will not even hit your target

IPv6 Extension Headers

Attacks

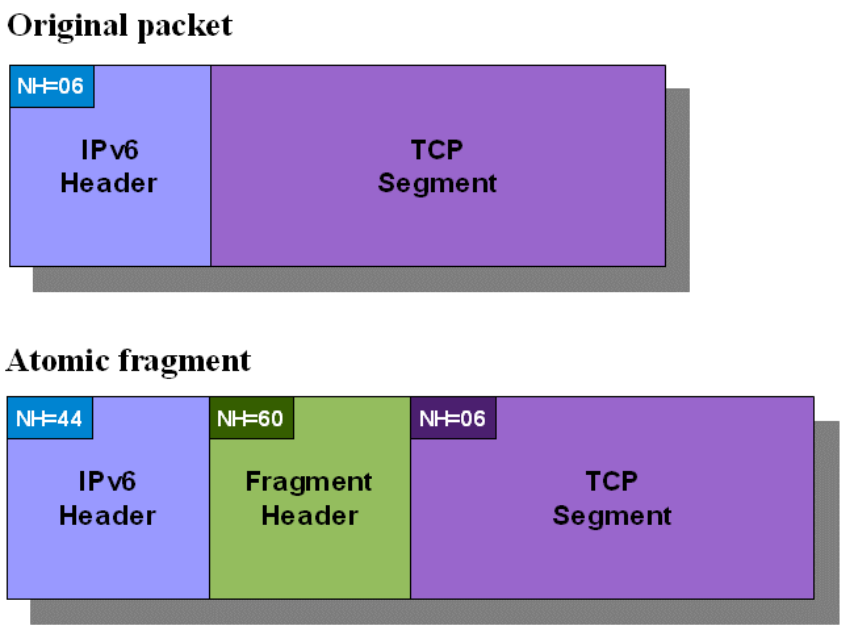
Old/obvious/boring stuff

- e.g. RA-Guard evasion



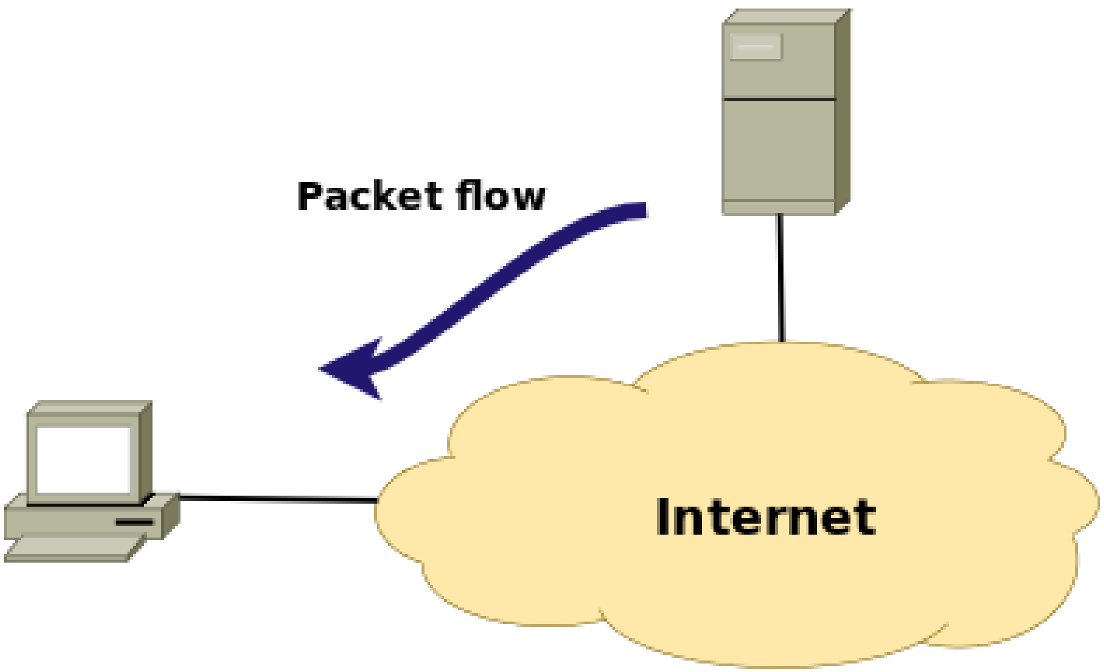
More interesting stuff

- If IPv6 frags are widely dropped...What if we triggered their generation?
  - Send an ICMPv6 PTB with an MTU<1280
  - The node will then generate IPv6 atomic fragments
  - Packets will get dropped



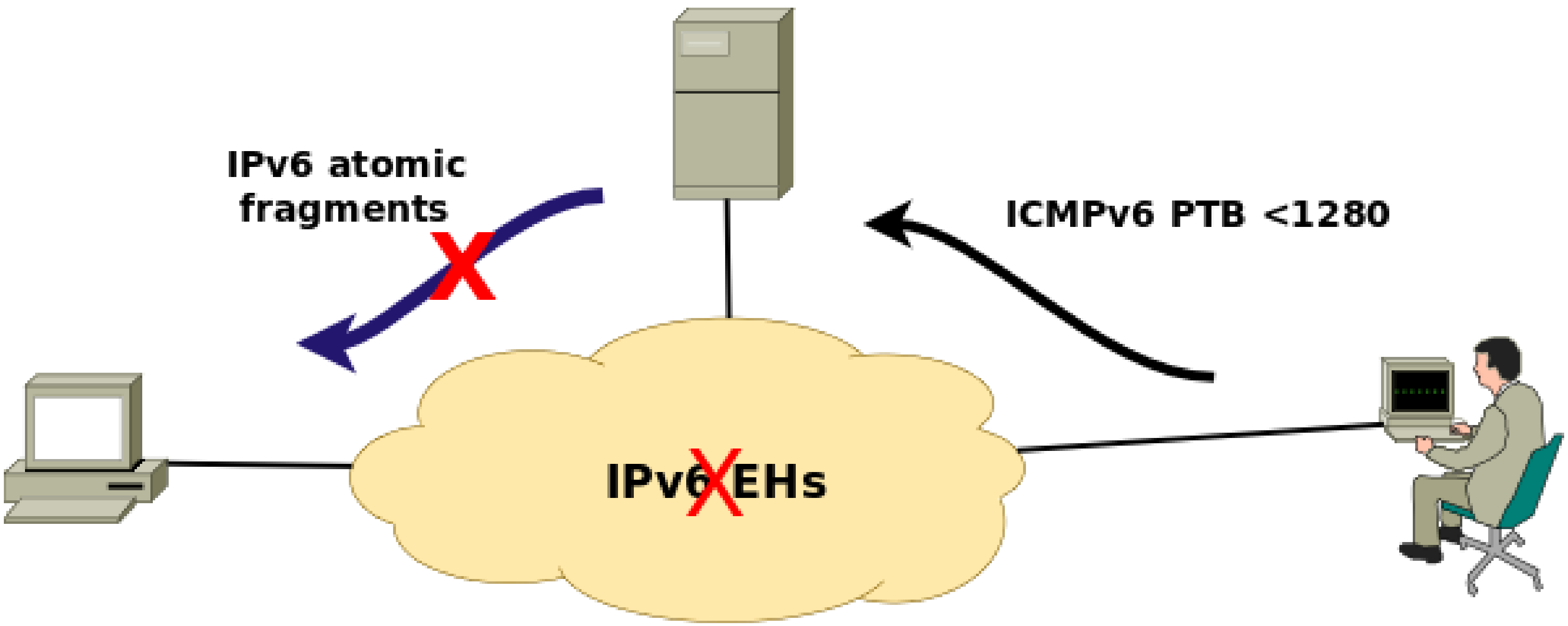
Attack Scenario #1

- Client communicates with a server



Attack Scenario #1 (II)

- Attacking client-server communications



Attack Scenario #1 (II)

- Simple way to reproduce it:
  - Attack and client machine is the same one
  - So we attack our own “connections”
- Attack:
  - Test IPv6 connetivity:

    telnet 2001:4f8:1:10:0:1991:8:25 80

  - Send an ICMPv6 PTB < 1280 to trigger atomic fragments

    sudo icmp6 --icmp6-packet-too-big -d 2001:4f8:1:10:0:1991:8:25 --peer-addr 2001:5c0:1000:a::a37 --mtu 1000 -o 80 -v

  - Test IPv6 connectivity again:

    telnet 2001:4f8:1:10:0:1991:8:25 80

SI6 NETWORKS

Attack scenario #2: Lovely BGP

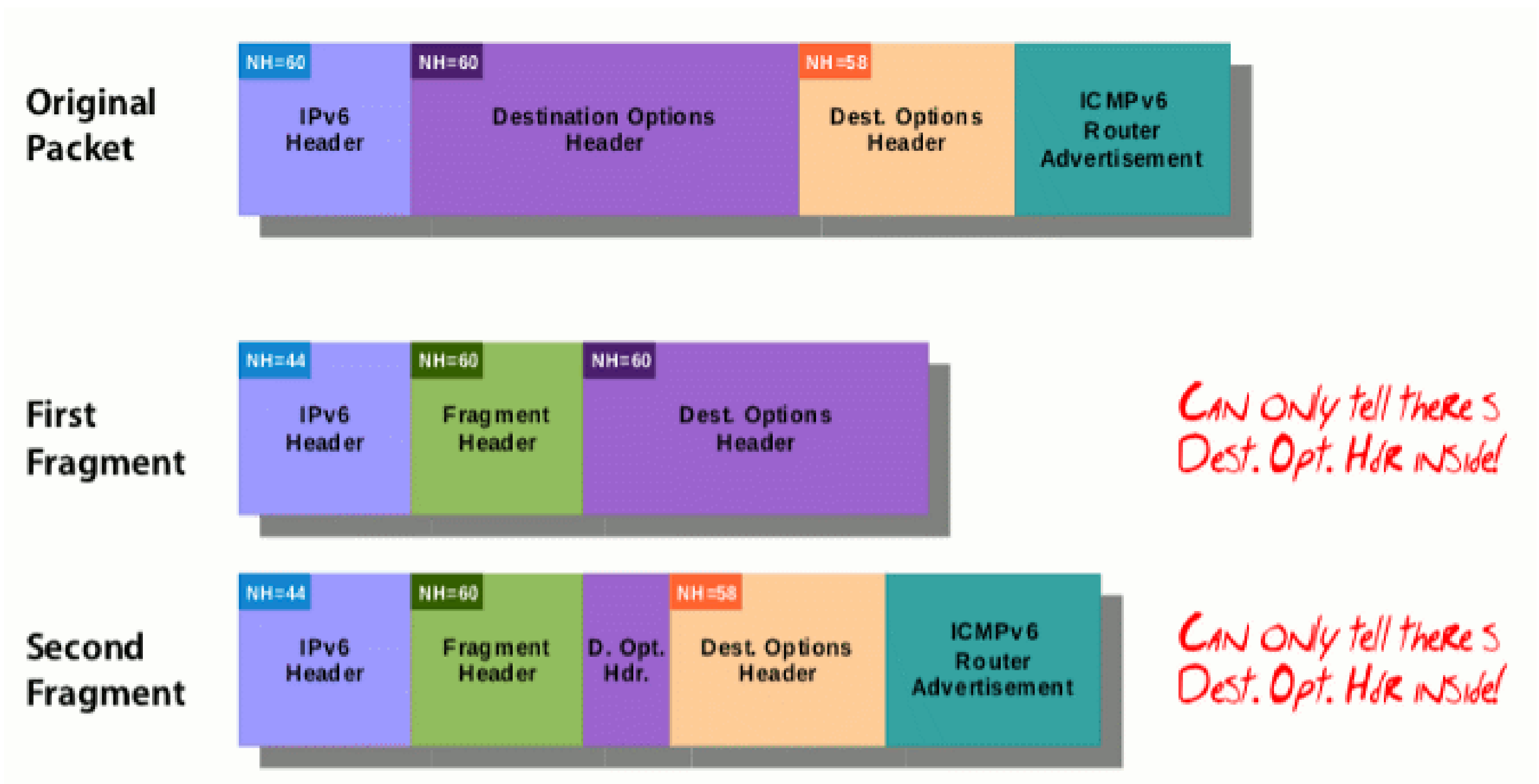
- Say:
  - We have two BGP peers
  - They drop IPv6 fragments “for security reasons”
  - But they do process ICMPv6 PTBs
- Attack:
  - Fire an ICMPv6 PTB <1280 (probably one in each direction)
- Outcome:
  - Packets get dropped (despite TCP MD5, IPsec, etc.)
  - Denial of Service

IPv6 Extension Headers

Improvements

Oversized IPv6 Header Chains

- RFC 7112 forbids oversized IPv6 header chains. e.g.:



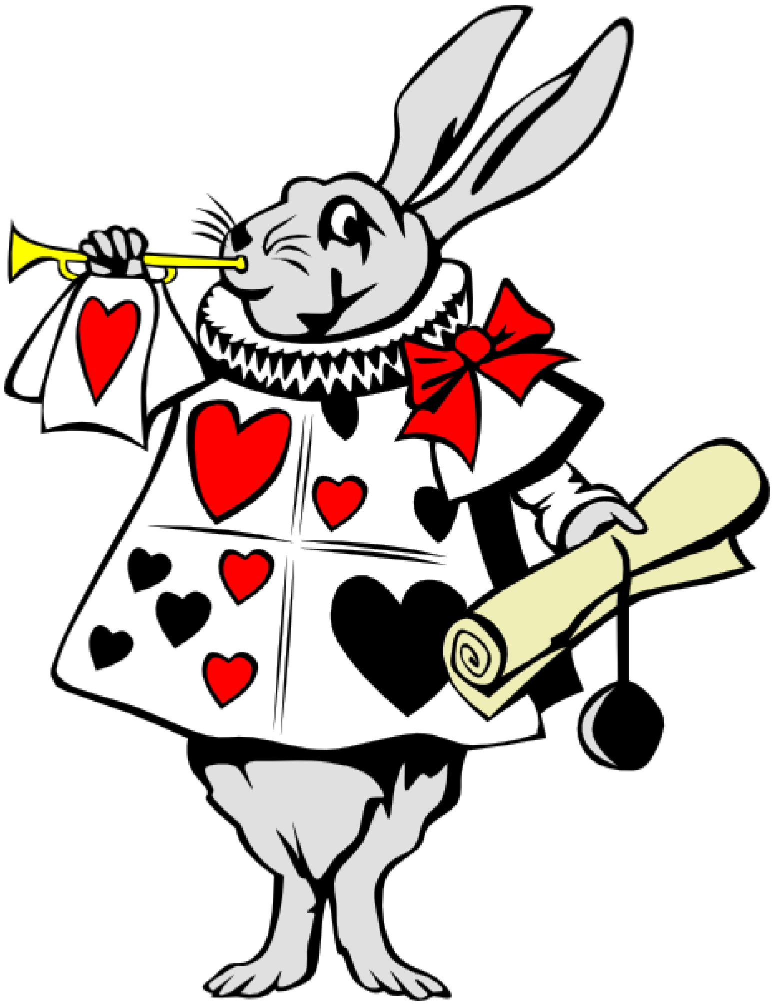
IPv6 atomic fragment generation

- draft-gont-6man-deprecate-atomfrag-generation
  - “Do not send IPv6 atomic fragments in response to ICMPv6 PTB < 1280”
  - Update SIIT (IPv6/IPv4 translation) such that it does not rely on them

Filtering of IPv6 Extension Headers

- There was no guidance in this area
- We produced draft-gont-opsec-ipv6-eh-filtering
    - Advice on filtering IPv6 packets that contain IPv6 Extension Headers

Part II: “Hack the talk”



IPv6 Toolkit v2.0!

SI6 Network's IPv6 Toolkit

- Supported OSes:
  - Linux, FreeBSD, NetBSD, OpenBSD, OpenSolaris, and Mac OS
- License:
  - GPL (free software)
- Home:
  - http://www.si6networks.com/tools/ipv6toolkit
- Collaborative development:
  - https://www.github.com/fgont/ipv6toolkit.git

SI6 Networks' IPv6 toolkit: Tools

- addr6: An IPv6 address analysis tool
- scan6: An IPv6 address scanner
- path6: A versatile IPv6-based traceroute
- frag6: Play with IPv6 fragments
- tcp6: Play with IPv6-based TCP segments
- udp6: Play with UDP datagrams
- ns6: Play with Neighbor Solicitation messages
- na6: Play with Neighbor Advertisement messages
- script6: Rather complex tasks made easy

SI6 Networks' IPv6 toolkit: Tools (II)

- rs6: Play with Router Solicitation messages
- ra6: Play with Router Advertisement messages
- rd6: Play with Redirect messages
- icmp6: Play with ICMPv6 error messages
- ni6: Play with Node Information messages
- flow6: Play with the IPv6 Flow Label
- jumbo6: Play with IPv6 Jumbograms

Get interesting addresses

Get domains and IPv6 addresses

- script6 can do batch-processing of domain names
- Get IPv6 addresses:

```
$ cat domains.txt | script6 get-aaaa
```

- Get mailserver addresses:

```
$ cat domains.txt | script6 get-mx | script6 get-aaaa
```

Filtering interesting addresses

- The addr6 tool can do virtually any kind of address filtering
- e.g., grab only traditional SLAAC addresses:

```
cat list.txt | addr6 -i -g ieee
```

Automatic smart IPv6 address scanning

- scan6 can automatically leverage patterns in IPv6 addresses
- Example:

```
File  Edit  View  Search  Terminal  Help
root@fgont-outside:~# scan6 -v -d scanme.nmap.org/64
Rate-limiting probe packets to 1000 pps (override with the '-r' option if neces
sary)
Target address ranges (1)
2600:3c01:0:0:0:0:0-100:0-1500

Alive nodes:
2600:3c01::2
2600:3c01::3
2600:3c01::a
2600:3c01::4b
2600:3c01::2:1002
2600:3c01::2:1003
2600:3c01::2:1001
2600:3c01::21:1000
```

EH-enabled IPv6 traceroute

path6 tool

- How far do your IPv6 EH-enabled packets get?
- No existing traceroute tool supported IPv6 extension headers
- Hence we produced our path6 tool
  - Supports IPv6 Extension Headers
  - Can employ TCP, UDP, or ICMPv6 probes
  - It's faster ;-)
- Example:

```
# path6 -u 100 -d fc00:1::1
```

Dst Opt Hdr

Finding IPv6 blackholes

blackhole6: Finding IPv6 blackholes

- How it works?
  - path6 without EHs + path6 with EHs + a little bit of magic

```
fgont@satellite:~$ sudo blackhole6 www.google.com do8
SI6 Networks IPv6 Toolkit v2.0
blackhole6: A tool to find IPv6 blackholes
Tracing www.google.com (2607:f8b0:400b:807::1012)...

Dst. IPv6 address: 2607:f8b0:400b:807::1012 (AS15169 - GOOGLE - Google
Inc.,US)
Last node (no EHs): 2607:f8b0:400b:807::1012 (AS15169 - GOOGLE - Google
Inc.,US) (13 hop(s))
Last node (DO 8): 2001:5a0:12:100::72 (AS6453 - AS6453 - TATA
COMMUNICATIONS (AMERICA) INC,US) (7 hop(s))
Dropping node: 2001:4860:1:1:0:1935:0:75 (AS15169 - GOOGLE - Google
Inc.,US || AS15169 - GOOGLE - Google Inc.,US)
```

blackhole6: Methodology

1) Run “normal” path6 to target (D), and save route (ROUTE)
2) Check that last “hop” in route is D
3) Run EH-enabled path6, and find last responding address (L)
4) Find “L” in “ROUTE” -> dropping system (X) is next in ROUTE
5) Compare AS(X) with AS(D), and produce other stats

blackhole6: Methodology (II)

- Given the output of path6 for no-EH and EHs:

No EHs

1. fc00:1:1:1000::1
2. fc00:1:1:2000::4
3. fc00:1:2:4000::1
4. fc00:2:1:4000::1
5. fc00:a:2:1000::1
6. fc00:a:4:4000::1
7. fc00:b:1:1000::1 — **DROP**
8. fc00:b:2:5000::1
9. fc00:b:4:5000::1
10. fc00:d::1

With EHs

1. fc00:1:1:1000::1
2. fc00:1:1:2000::4
3. fc00:1:2:4000::1
4. fc00:2:1:4000::1
5. fc00:a:2:1000::1
6. fc00:a:4:4000::1

blackhole6: Methodology (III)

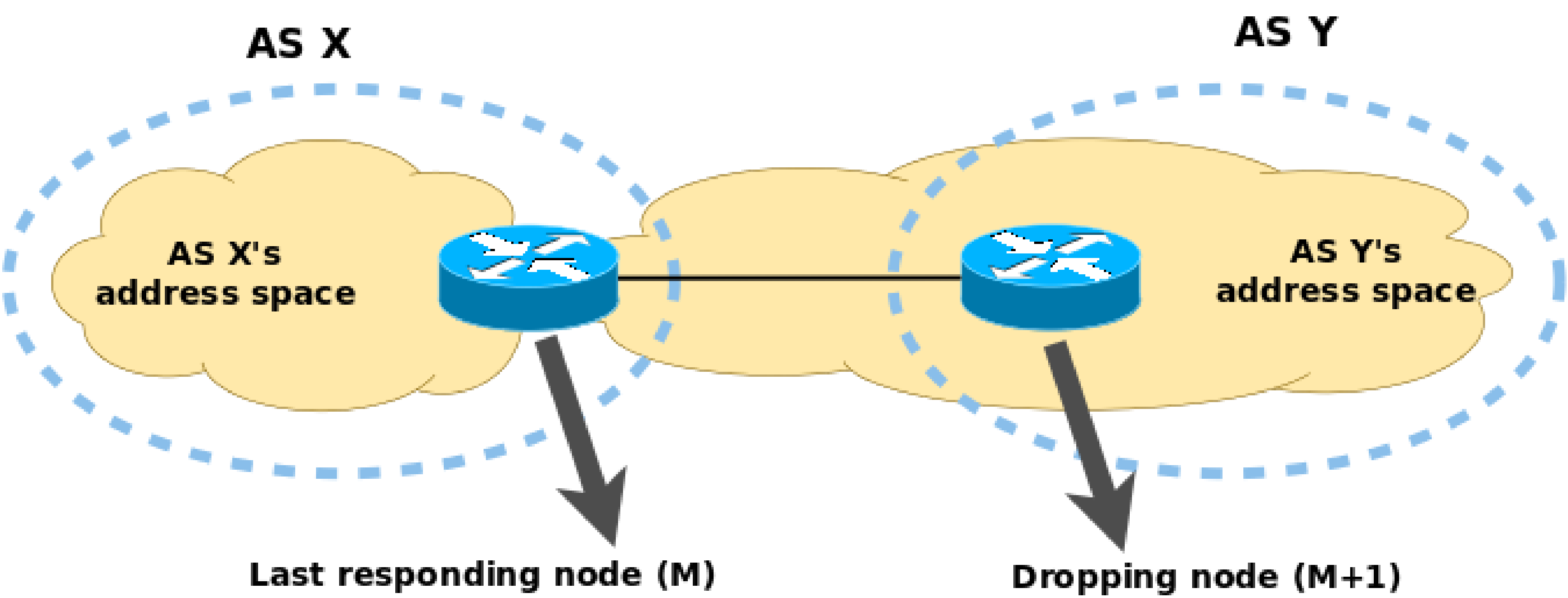
- We assume ingress filtering...
- Otherwise dropping node actually is M rather than M+1

blackhole6: ASes

- Lookup ASN of dropping node, but...
- There may be ambiguity when finding the AS of the dropping node:
  - who provides the address space for the peering?

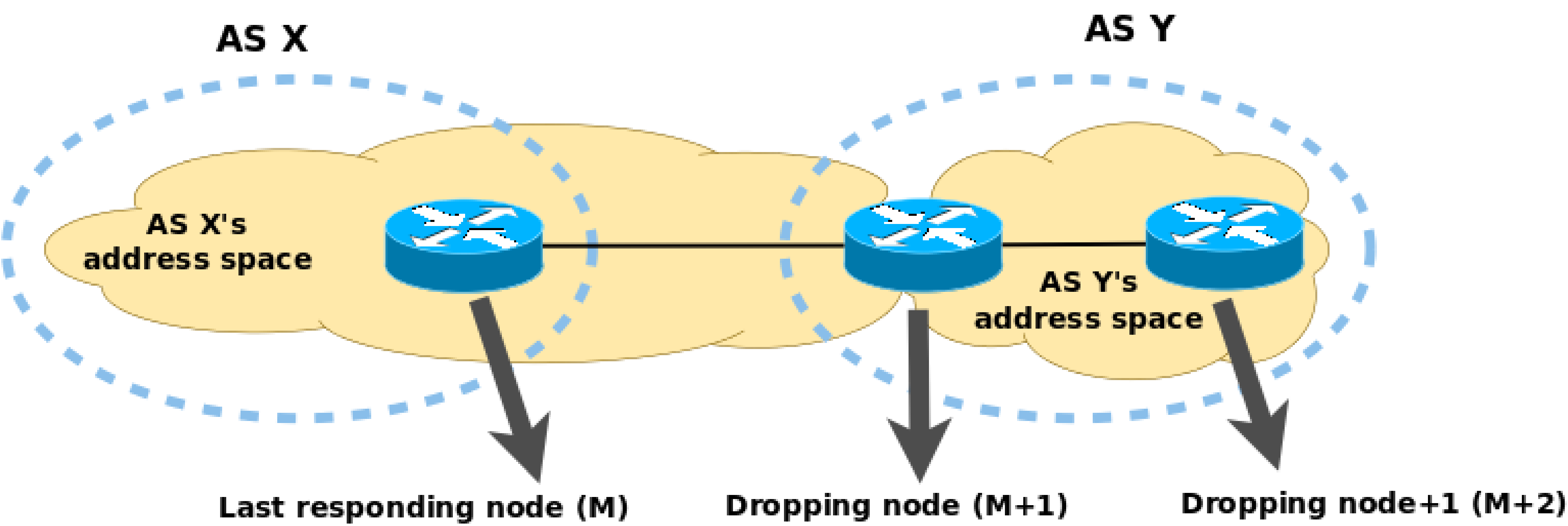
blackhole6: ASes (II)

- Case 1: Address space provided by AS Y



blackhole6: ASes (III)

- Case 2: Address space provided by AS X



Some conclusions

Some conclusions

- The IPv6 Internet is the IPv4 Internet of the '90's
- Still lots of stuff to be done in the IPv6 security arena
  - Improve the specs
  - Patch your IPv6 stack
  - Write code that demonstrates new ideas
- **Master IPv6 before it is too late**

Questions?

Acknowledgements

- Daniela Strobel & Cirosec crew
- Attendees to this week's “Hacking IPv6 Networks v3.0” training course

Thanks!

Fernando Gont

fgont@si6networks.com

IPv6 Hackers mailing-list

http://www.si6networks.com/community/

www.si6networks.com