IPV6

A home/work system fully utilizing IPV6

SLUUG Presentation by

David Forrest

ToBeDetermined

IPV6

20 years ago RFC1883 (1996), the RFC that formally defined IPv6, was published by the IETF. From 1996 untill 2006 the 6bone existed and functioned as a testing ground for IPv6. Since 2006, which is now a decade ago, IPv6 has been available worldwide in production from a increasing variety of ISPs.

During the last decade, IPv4 address space has also run out at most of the RIRs and most of the larger Internet properties have enabled IPv6 on their services.

IPV6

So it's now not ready for prime time use?

Google, Facebook, Comcast, Cox, and Maple Park Development use it

- And AT&T Internet Services, my ISP,

  sort of uses it

IPV6

MPDC of Kirkwood MO ↔ Paris, FR

- IPv4 speed 99.26.132.228
- ISP AT&T Internet Services
- Speed 11.5 Mbit/s

- IPv6 speed 2602:306:31a8:4e40:8073:a095:3096:3d85
- ISP AT&T Internet Services
- Speed 12.9 Mbit/s

IPV6

So why not used more?

Some applications don't connect reliably

Many ISPs don't offer it

Many think they don't need it

Host configuration

IPV6

So why do I use it?

Mainly - Secure Socket Layer connections

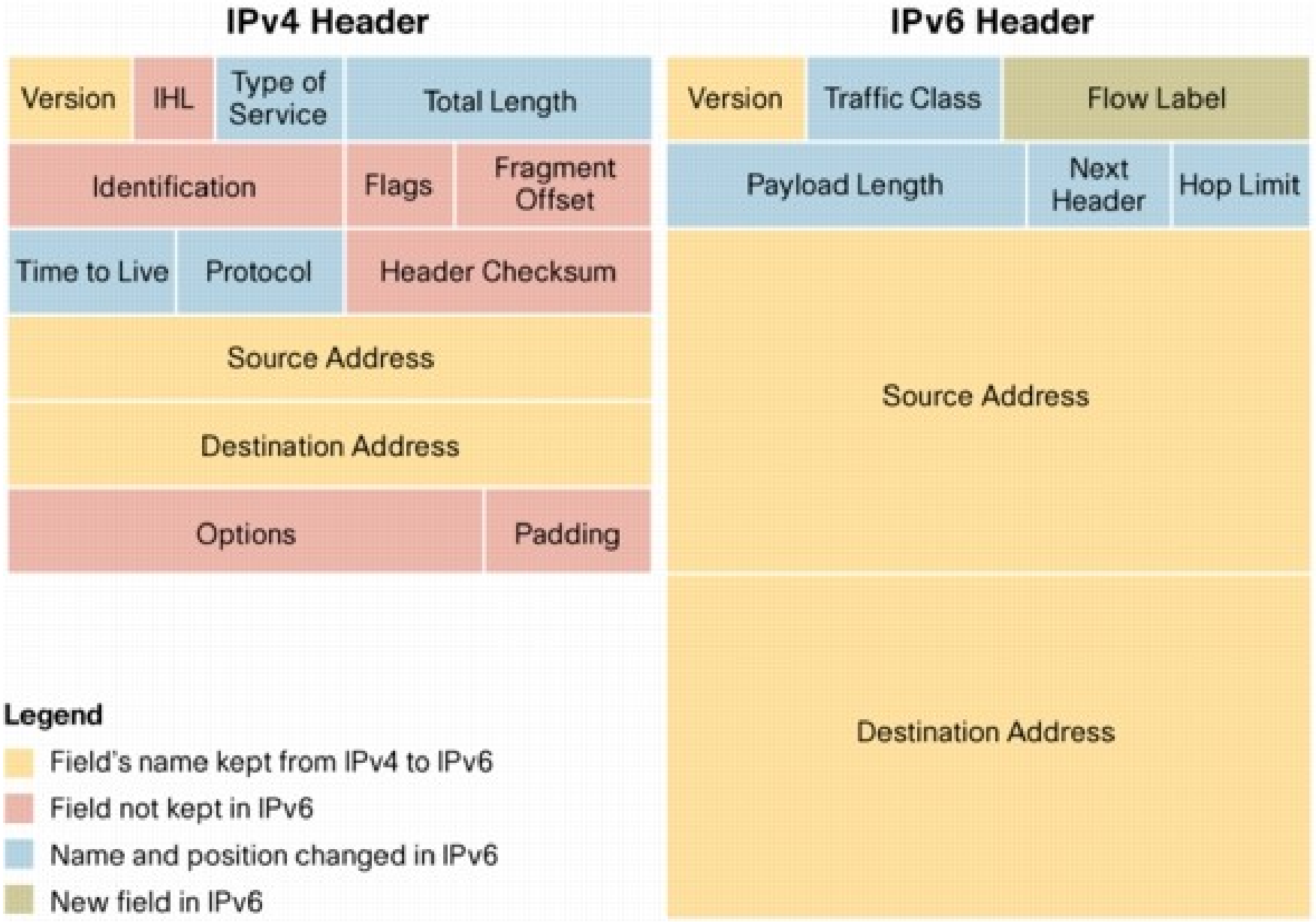
Prior experience

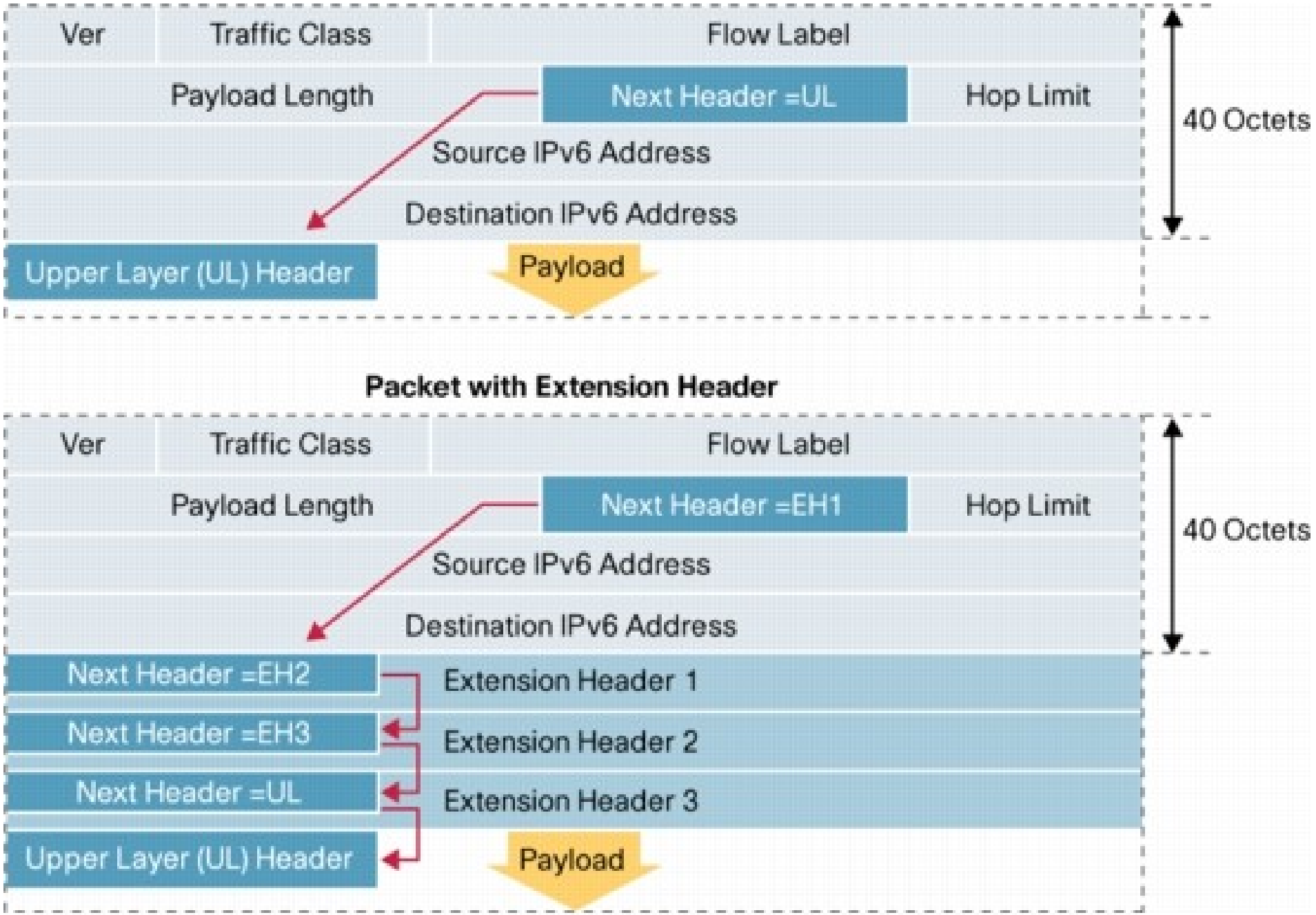
Effective and easier Firewalling

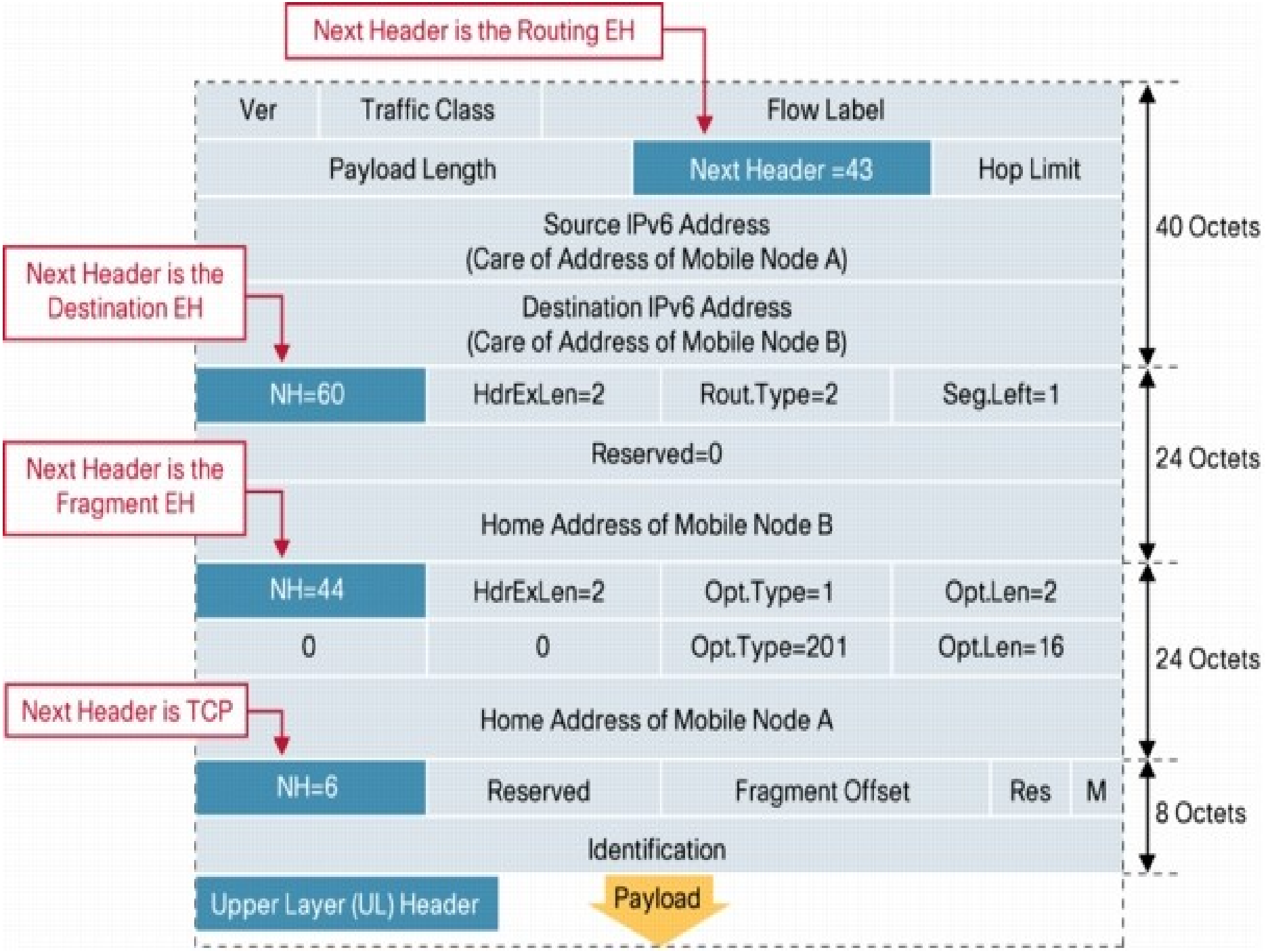
IPV6

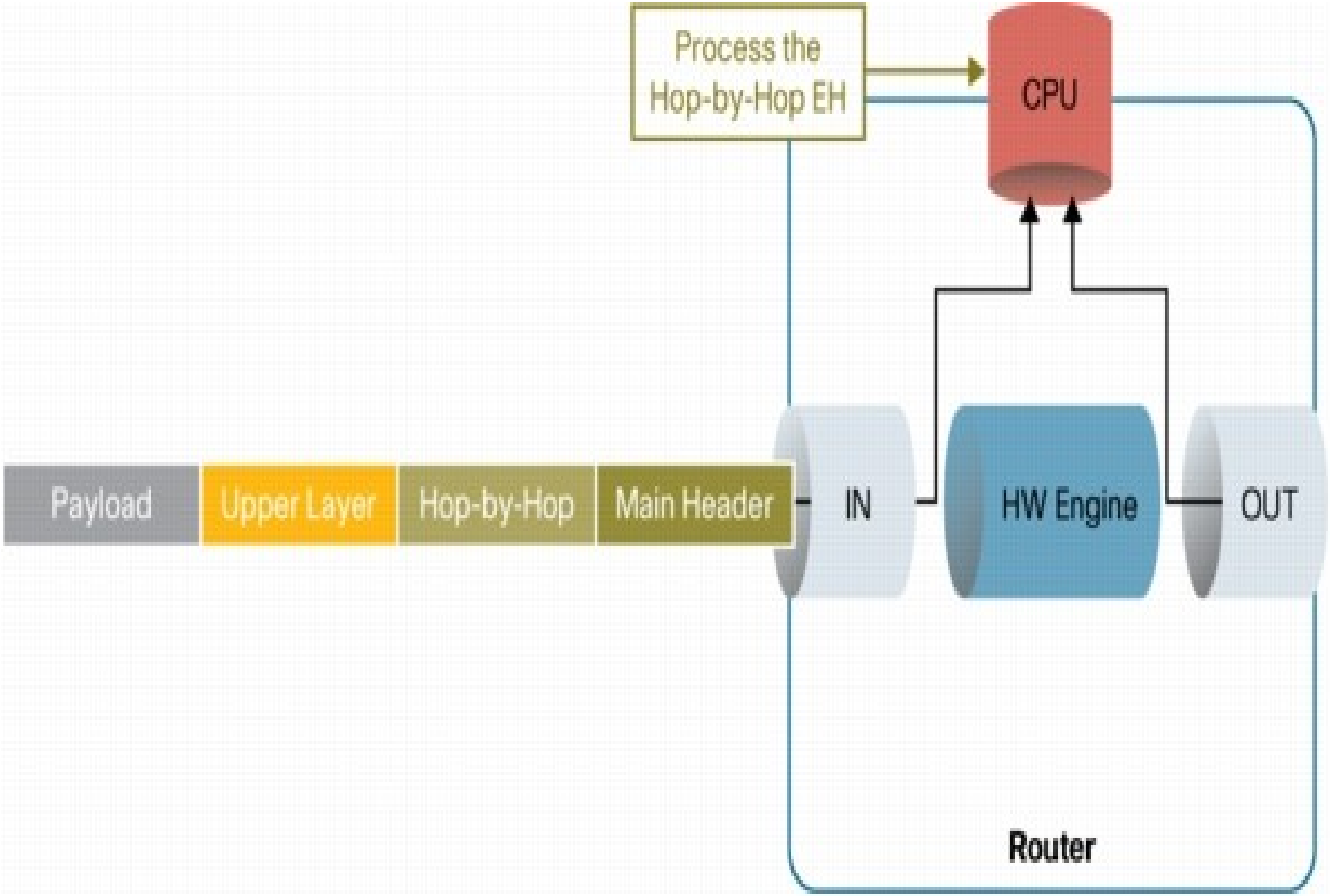
Operational Implications of IPv6 Packets with Extension Headers have caused some application problems as described here and the next couple of slides in probably too much detail but I'll review briefly

More fully for those inclined: https://www.ietf.org/proceedings/94/slides/slides-94-v6ops-6.pdf


IPv4 Header
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
IPv6 Header
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address
Legend
Field's name kept from IPv4 to IPv6
Field not kept in IPv6
Name and position changed in IPv6
New field in IPv6


Ver
Traffic Class
Flow Label
Payload Length
Next Header =UL
Hop Limit
Source IPv6 Address
Destination IPv6 Address
Upper Layer (UL) Header
Payload
40 Octets
Packet with Extension Header
Ver
Traffic Class
Flow Label
Payload Length
Next Header =EH1
Hop Limit
Source IPv6 Address
Destination IPv6 Address
Next Header =EH2
Extension Header 1
Next Header =EH3
Extension Header 2
Next Header =UL
Extension Header 3
Upper Layer (UL) Header
Payload
40 Octets


Next Header is the Routing EH
Ver
Traffic Class
Flow Label
Payload Length
Next Header =43
Hop Limit
Source IPv6 Address
(Care of Address of Mobile Node A)
Next Header is the Destination EH
Destination IPv6 Address
(Care of Address of Mobile Node B)
NH=60
HdrExLen=2
Rout.Type=2
Seg.Left=1
Reserved=0
Next Header is the Fragment EH
Home Address of Mobile Node B
NH=44
HdrExLen=2
Opt.Type=1
Opt.Len=2
0
0
Opt.Type=201
Opt.Len=16
Next Header is TCP
Home Address of Mobile Node A
NH=6
Reserved
Fragment Offset
Res
M
Identification
Upper Layer (UL) Header
Payload
40 Octets
24 Octets
24 Octets
8 Octets


Process the
Hop-by-Hop EH
CPU
Payload
Upper Layer
Hop-by-Hop
Main Header
IN
HW Engine
OUT
Router

The difficulty here is due to “Layer 4”, a part of the internet TCP/IP protocol that ties the transport of a single packet to a serially oriented stream as a possible connection. IPV6 encrypts the stream for security purposes and relies on the therefore secure stream for data transport. Hardwired (no CPU) routers in the transporting route must do this but the information in a fragmented stream may not be available to them to reassemble. So they just punt (return an ICMPv6 “Packet Too Big”).

IPV6

Hardwired routers do not do the additional processing on the content of packet as it requires CPU processing time to determine what has to be done but instead just return an ICMP packet that essentially says, “Packet is too large for me to even consider.” Fragments are therefore rejected as not read and the attempt fails.

The protocol assumes the sender notes this, adjusts the packet size, and tries again.

IPV6

Tips for success

- Allow ICMP thus avoiding timeouts with additional local network processing due to packet size
- Use the smallest protocol packet size (MTU=1280) for external traffic to eliminate fragmenting and thus insure routing ability outside your system. As most of us do not have gigabyte ISP access, the packet size effect is immaterial.

IPV6

- Internally I do have gigabyte access so I just use the maximum MTU (AT&T's 2-Wire 3800 (6rd?) mtu=1472 as set by their DHCP with auto config)

default via fe80::224:56ff:fea5:1d79 dev br0 proto ra metric 1024 expires 1516sec mtu 1472 hoplimit 64

IPV6

- [root@dave ~]157 # traceroute6 vp1 (Chicago)
- traceroute to vp1 (2600:3c00::f03c:91ff:fe56:7e17), 30 hops max, 80 byte packets
- 1 2602:306:31a8:4e40::1 (2602:306:31a8:4e40::1) 1.766 ms * *
- 2 * * * * * * *
- 9 * * *
- 10 vp1.maplepark.com (2600:3c00::f03c:91ff:fe56:7e17) 76.013 ms 74.266 ms 74.239 ms

IPV6

- [root@dave ~]159 # traceroute6 ns1 (local)
- traceroute to ns1 (2602:306:31a8:4e40::63b2:9929), 30 hops max, 80 byte packets
- 1 ns1.maplepark.com (2602:306:31a8:4e40::63b2:9929) 1.121 ms 1.124 ms 1.118 ms

IPV6

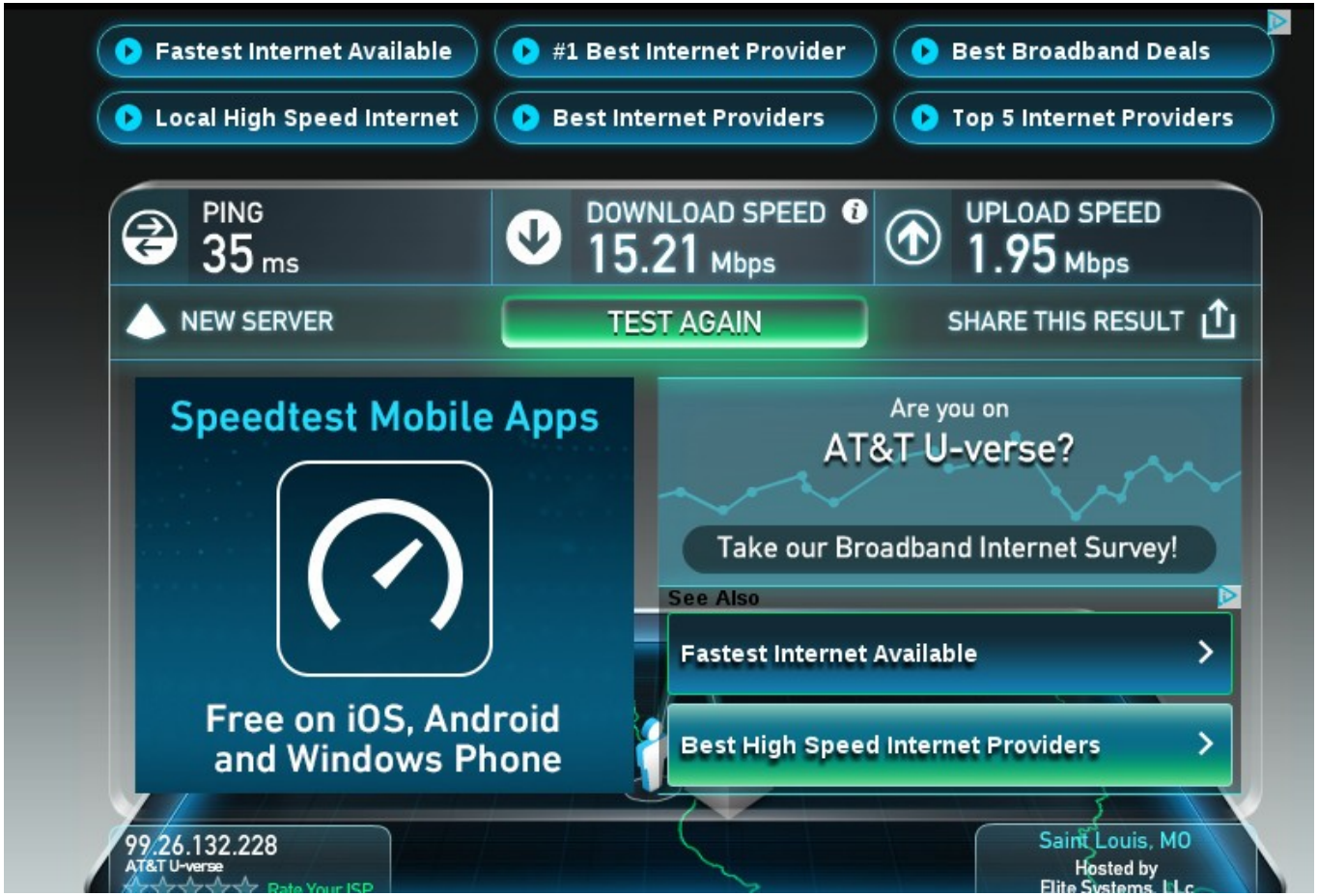
Updating my 4.3GB remote Chicago hosted website takes under 3 minutes by a cron job every 4 hours

[drf@ns1 ~]$ time /usr/local/bin/updatepublic_html
sending incremental file list

sent 167 bytes received 12 bytes 358.00 bytes/sec
total size is 425,077 speedup is 2,374.73

real 2m24.305s
user0m0.744s
sys 0m0.993s

My ISP speeds



IPV6

And we have universally accessible SSL(TLS), encrypted SSH available on my local host's interface MAC subject to the firewall configuration on my gateway machine.

The firewall is another talk.

- Questions?