Hacking the Network Stack to Enhance IPv4

Stephen ‘afterburn’ Janansky

Talk Outline

- 1992-1994: A Time Period of Significance to Today
- IPv6 Challenges
- Enhanced IP

Some Important R&D 1992-1994

- **Nov 1992 - publication of EIP as RFC 1385**
- **Jan 1993 - original ACM SIGCOMM paper describing NAT**
- **1992 - 1994 - Robert Ullman's IPv7 in 1992 became known as TP/IX in 1993. TP/IX proposed changes to IP and TCP at the same time. IP would use 64-bit addresses, 3 bytes for administrative domain, 3 bytes for network address, and 2 bytes for host. In 1994 proposal evolved into a completely new design called CATNIP but also keeping IPv7 name. CATNIP was about universal interoperability for IP, CLNP, and IPX.**
- **Dec 1993 - publication of RFP by IPng working group. Received proposals on SIP, CATNIP, & TUBA. EIP did not submit a proposal. SIP proposed increase of IP space from 32-bit to 64-bit addresses. TUBA and CATNIP used 160-bit ISO CLNP addresses. Later SIP merged with PIP and became SIPP.**

Some Important R&D 1992-1994

- **Mar 1994 - Brian Carpenter suggested use of IP options in protocol called AEIOU. In the Mar 1994 IETF meeting minutes, Steve Deering "noted that AEIOU should go into the same status as EIP: honored, revered, unimplemented."**
- **Jul 1994 - SIPP is chosen by the IPng Directorate to become IPv6 after changing address size from 64-bit to 128-bit**

IPv6 Implementation Problems

- The CPE Problem
- From Brian E. Carpenter’s 2010 IPv6 Task Force talk
    - Billing Systems, Handsets, management interfaces and systems, DSLAMs, Routers, Traffic mgmt boxes, load balancers, VPN boxes, SIP boxes, firewalls
- End to End principle violated to support LTE needs
    - NAT64 to support IPv6 LTE subscribers reachability to legacy IPv4

IPv6 Implementation Problems

- “Deploying IPv6 in the Google Enterprise Network. Lessons learned.” It’s hard for Google. They’re working with vendors as problems arise.
- Geoff Huston’s Nanog 53 Talk on IPv4 Address Exhaustion and how we’re presently running IPv4, IPv6, tunnelling, CGNs, CDNs, ALGs and how market forces are driving transitions or lack of transitions.
- The reality is all these things are happening at the same time. We’ve proposed adding other protocols like Enhanced IP into the market to make people think.

Enhanced IP

Enhanced IP

Enhanced IP

- 65.127.221.1.10.3.3.2

Enhanced IP

- **65.127.221.1.10.3.3.2**
- All Enhanced IP addresses have a site address and a host address

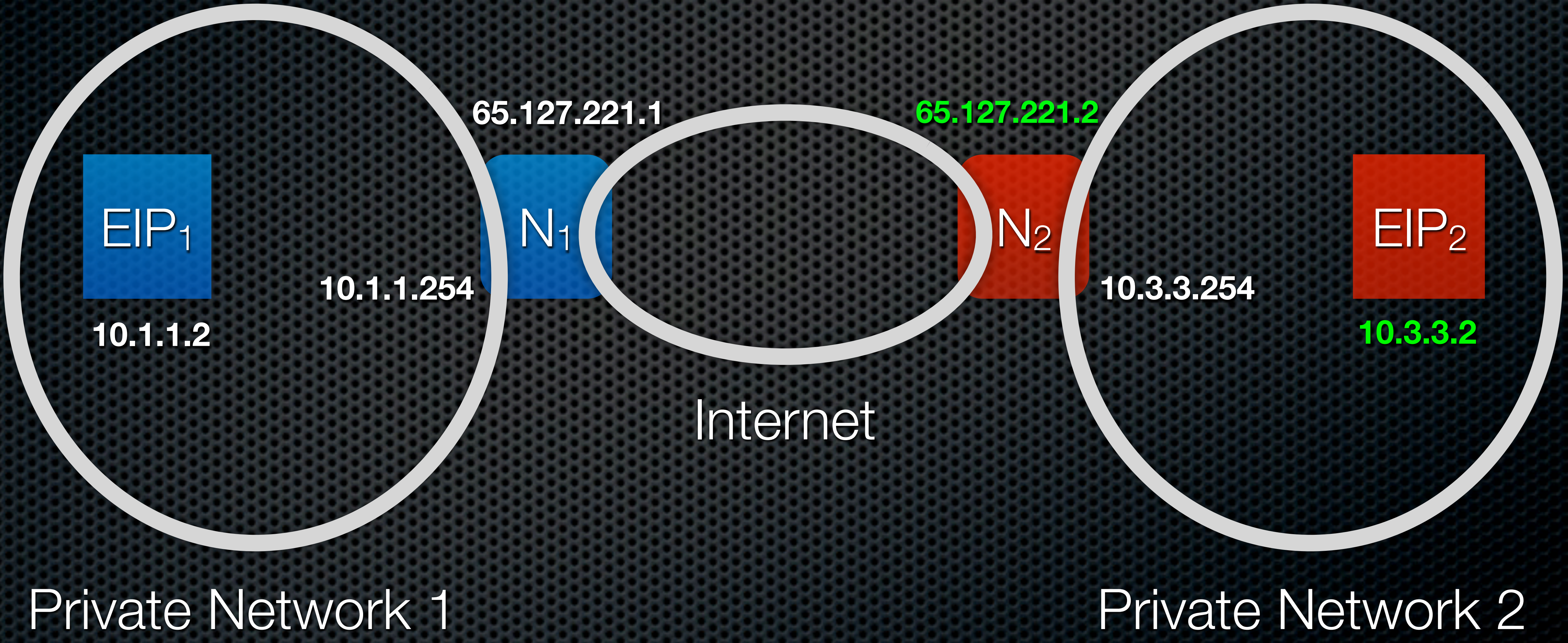
Enhanced IP

- **65.127.221.1.10.3.3.2**
- All Enhanced IP addresses have a site address and a host address
- **Site address: 65.127.221.1**, is used to route packets over the public Internet to a router/NAT that is aware of Enhanced IP packet format. This would generally be a public IPv4 address.

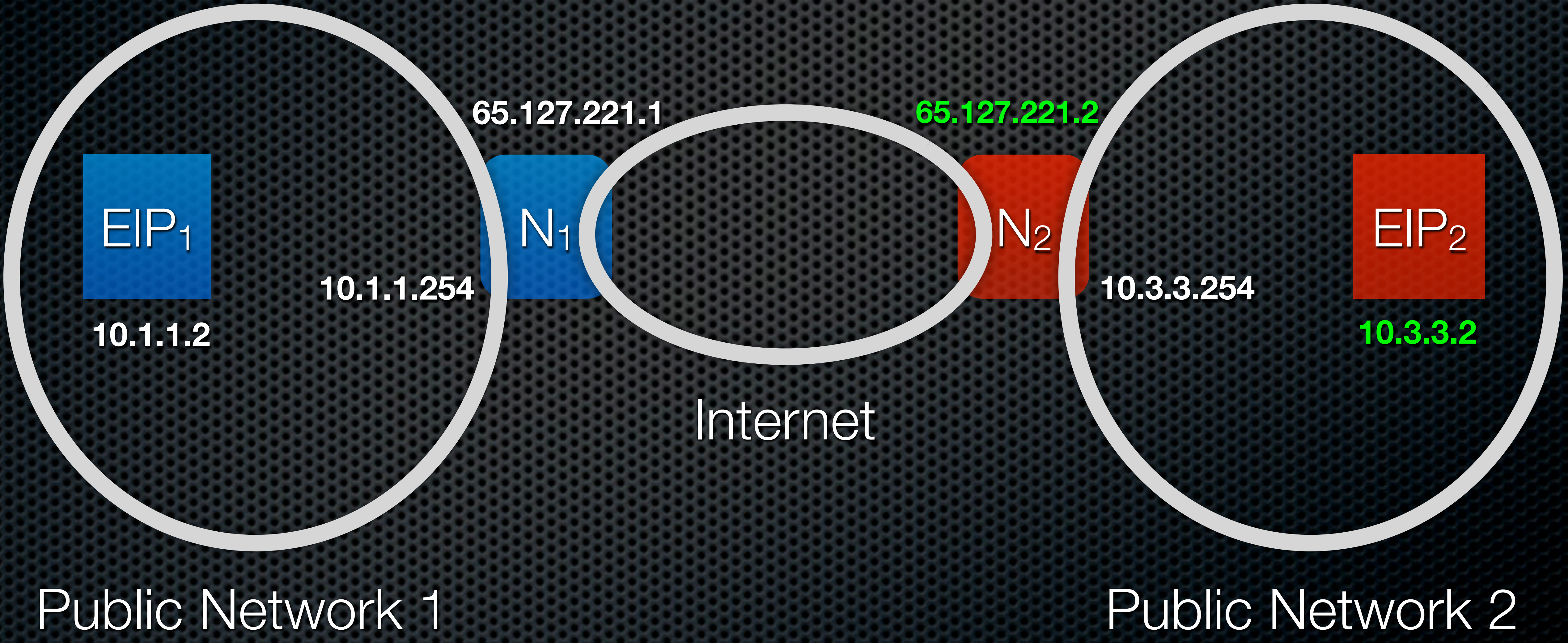
Enhanced IP

- **65.127.221.1.10.3.3.2**
- All Enhanced IP addresses have a site address and a host address
- **Site address: 65.127.221.1**, is used to route packets over the public Internet to a router/NAT that is aware of Enhanced IP packet format. This would generally be a public IPv4 address.
- **Host address: 10.3.3.2**, used to route packets to a node behind the router/NAT that has the outside address of 65.127.221.1

Enhanced IP Network



Enhanced IP Network



Enhanced IP

Enhanced IP

- Minimal changes at layer 2 and 3 of OSI to implement

Enhanced IP

- Minimal changes at layer 2 and 3 of OSI to implement
  - does not replace ARP, DHCPv4, routing protocols

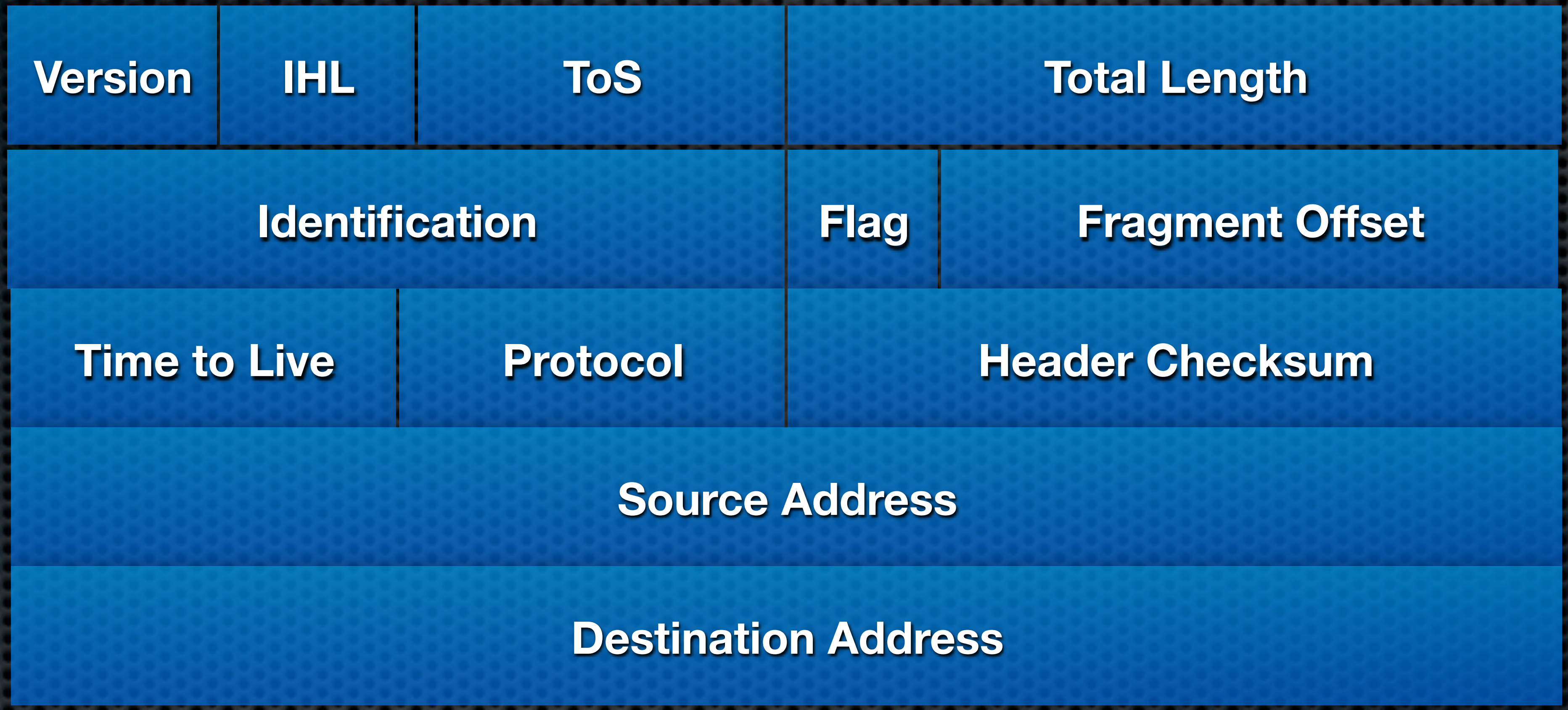
Enhanced IP

- Minimal changes at layer 2 and 3 of OSI to implement
  - does not replace ARP, DHCPv4, routing protocols
- Uses IP Option 26 to extend the length of the IP header, 12 bytes per packet

Enhanced IP

- Minimal changes at layer 2 and 3 of OSI to implement
  - does not replace ARP, DHCPv4, routing protocols
- Uses IP Option 26 to extend the length of the IP header, 12 bytes per packet
- The NAT functionality used in Enhanced IP is stateless as opposed to the stateful nature of IPv4 NAT.

IPv4 Header



Version | IHL | ToS | Total Length

Identification | Flag | Fragment Offset

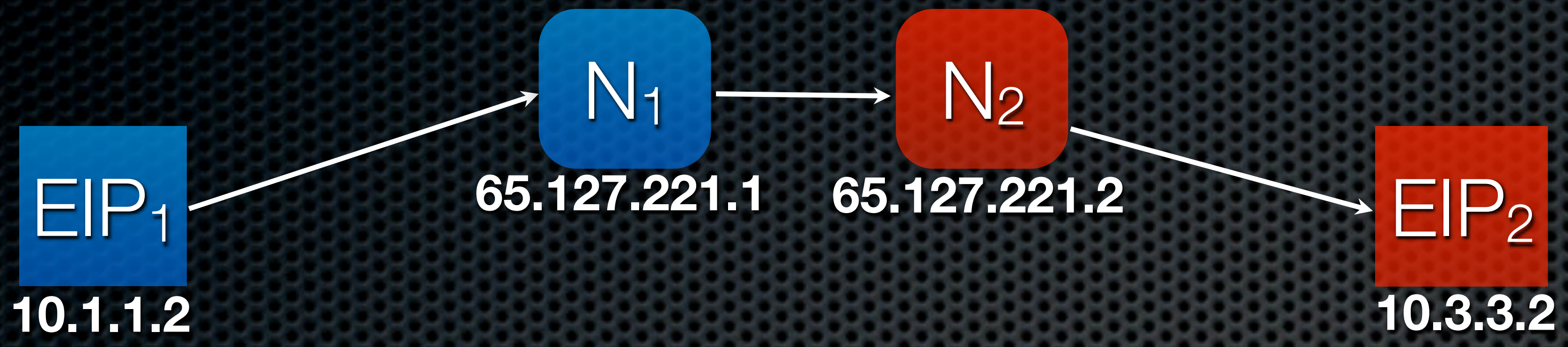
Time to Live | Protocol | Header Checksum

Source Address

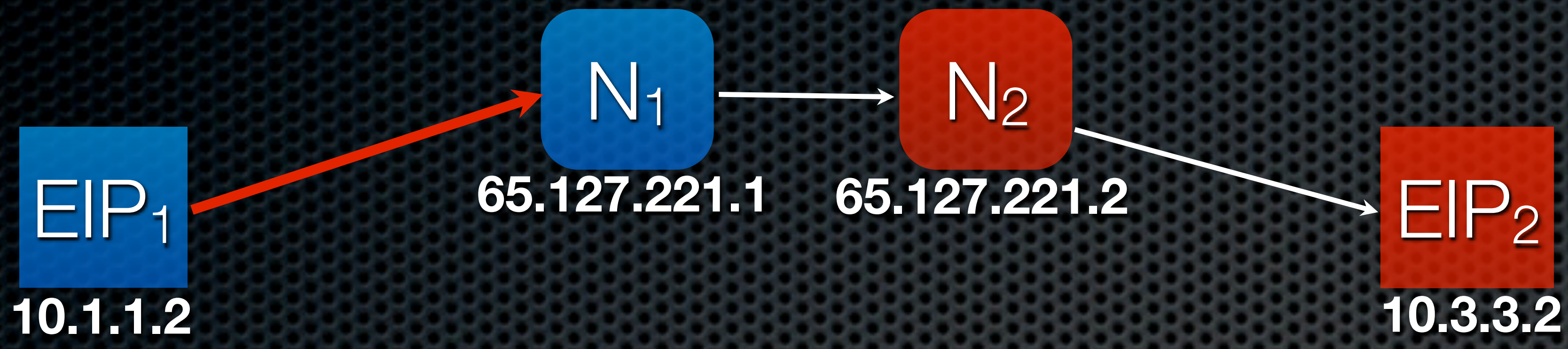
Destination Address

Enhanced IPv4 Header

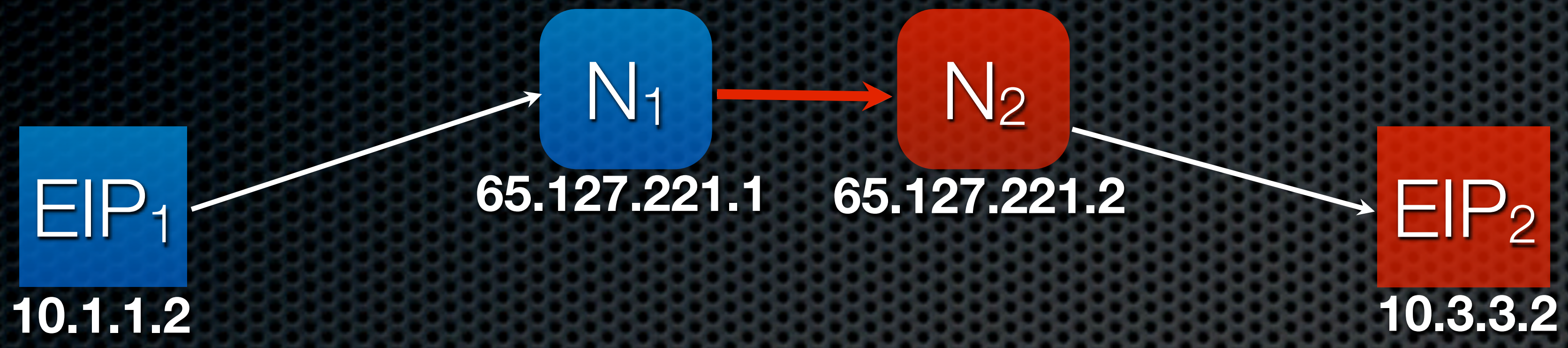
<table>
<tr><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="8">ToS</td><td colspan="16">Total Length</td></tr>
<tr><td colspan="16">Identification</td><td colspan="3">Flag</td><td colspan="13">Fragment Offset</td></tr>
<tr><td colspan="8">Time to Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td colspan="32">Source Address</td></tr>
<tr><td colspan="32">Destination Address</td></tr>
<tr><td colspan="8">Option ID</td><td colspan="8">Option Length</td><td colspan="3">ESP</td><td colspan="3">EDP</td><td colspan="10">Reserved</td></tr>
<tr><td colspan="32">Extended Source Address</td></tr>
<tr><td colspan="32">Extended Destination Address</td></tr>
</table>



<table>
<tr><td>Version</td><td>IHL</td><td colspan="2">ToS</td><td colspan="4">Total Length</td></tr>
<tr><td colspan="4">Identification</td><td colspan="2">Flag</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td colspan="2">Protocol</td><td colspan="4">Header Checksum</td></tr>
<tr><td colspan="8">10.1.1.2</td></tr>
<tr><td colspan="8">65.127.221.2</td></tr>
<tr><td colspan="2">0x9a</td><td colspan="2">12</td><td>0</td><td>1</td><td colspan="2">0</td></tr>
<tr><td colspan="8">255.255.255.255</td></tr>
<tr><td colspan="8">10.3.3.2</td></tr>
</table>

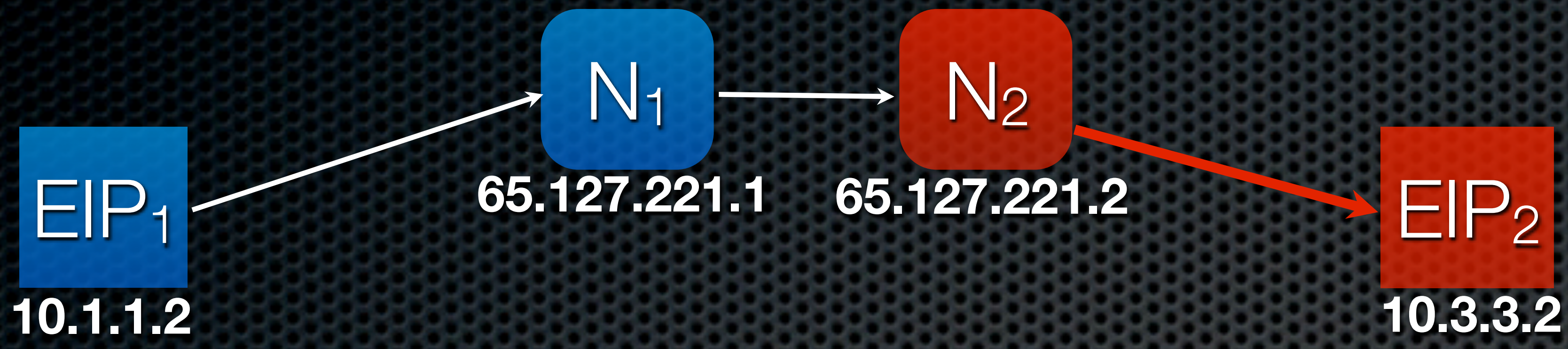


<table>
<tr><td>Version</td><td>IHL</td><td>ToS</td><td colspan="3">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flag</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol</td><td colspan="3">Header Checksum</td></tr>
<tr><td colspan="6">10.1.1.2</td></tr>
<tr><td colspan="6">65.127.221.2</td></tr>
<tr><td colspan="2">0x9a</td><td>12</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">255.255.255.255</td></tr>
<tr><td colspan="6">10.3.3.2</td></tr>
</table>



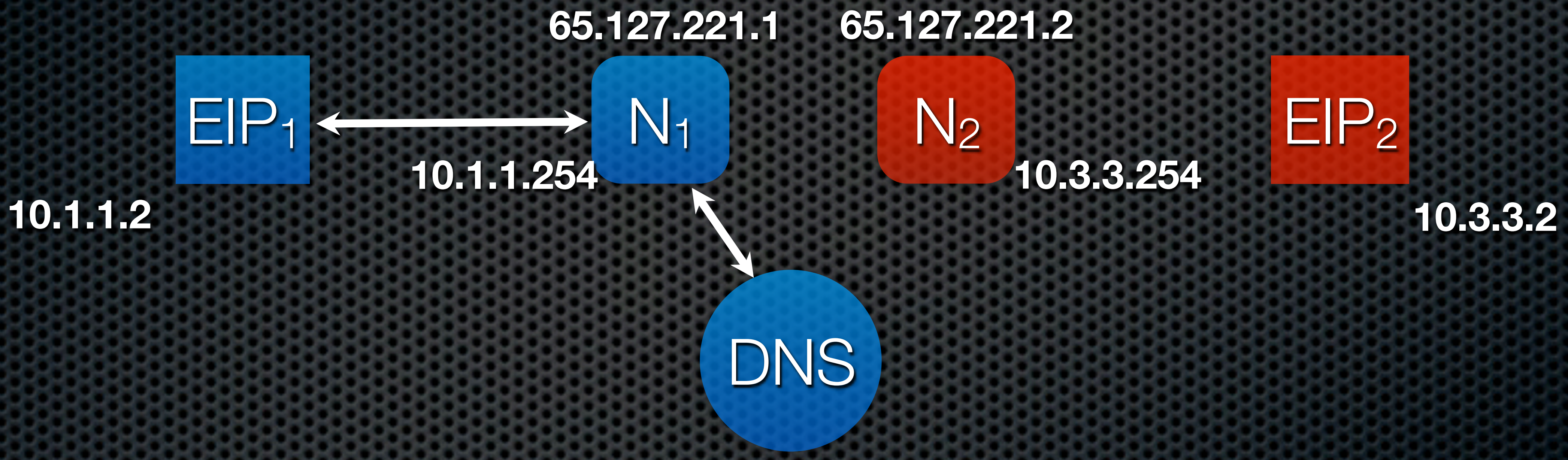
Version	IHL	ToS	Total Length
Identification		Flag	Fragment Offset
Time to Live	Protocol	Header Checksum	
65.127.221.1			
65.127.222.2			

0x9a	12	1	1	0
10.1.1.2				
10.3.3.2				



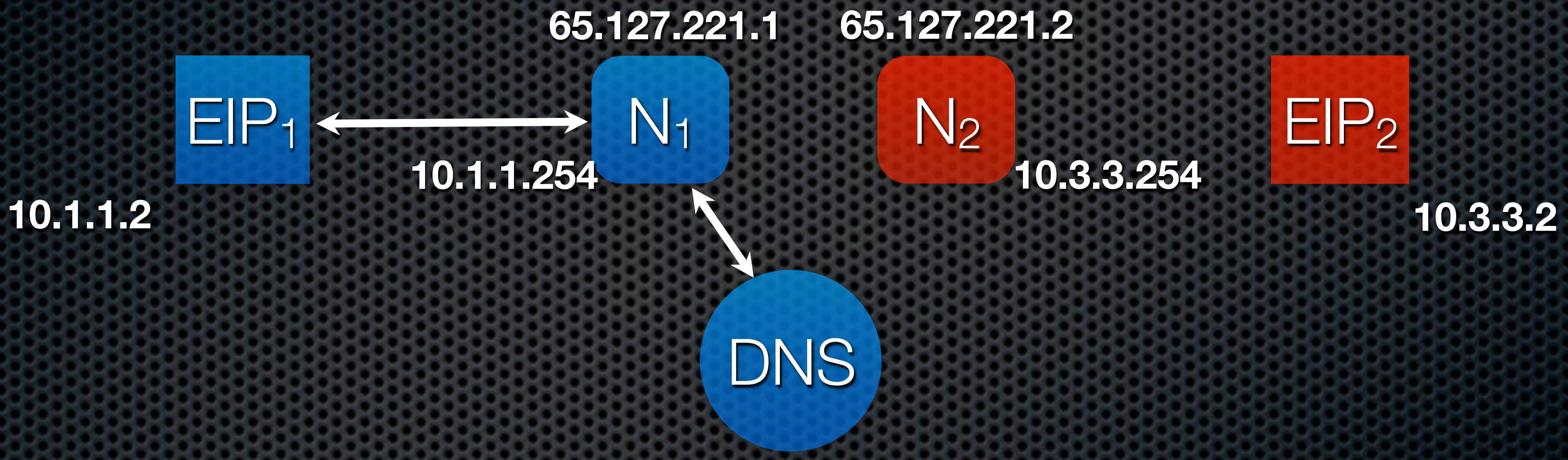
Version	IHL	ToS	Total Length	
Identification	Flag	Fragment Offset		
Time to Live	Protocol	Header Checksum		
65.127.221.1				
10.3.3.2				
0x9a	12	1	0	0
10.1.1.2				
0.0.0.0				

DNS: Using AAAA to lookup 64 bits



- EIP1 sends a AAAA request for eip2.somesite.com and receives back 2001:0101:417F:DD02:0a03:0302::0

DNS: Using AAAA to lookup 64 bits



- 2001:0101:417F:DD02:0a03:0302::0 is really 65.127.221.2.10.3.3.2

Enhanced IP code

Enhanced IP code

- Linux OS changes

Enhanced IP code

- Linux OS changes
  - ~200 lines in the kernel, ~500 lines in user-space

Enhanced IP code

- Linux OS changes
  - ~200 lines in the kernel, ~500 lines in user-space
- edge device (SOHO router)

Enhanced IP code

- Linux OS changes
  - ~200 lines in the kernel, ~500 lines in user-space
- edge device (SOHO router)
  - ~450 line driver, 8 line patch to NAT code

Enhanced IP code

- Linux OS changes
  - ~200 lines in the kernel, ~500 lines in user-space
- edge device (SOHO router)
  - ~450 line driver, 8 line patch to NAT code
- Linux Utilities

Enhanced IP code

- Linux OS changes
  - ~200 lines in the kernel, ~500 lines in user-space
- edge device (SOHO router)
  - ~450 line driver, 8 line patch to NAT code
- Linux Utilities
  - ~3500 lines: ping, traceroute, netcat-like program, measurement programs

Userspace Connect

```
#pragma pack(1)

struct sockaddr_ein{

        unsigned short sin_family;

        unsigned short sin_port;

        in_addr_t sin_addr1;

        in_addr_t sin_addr2;

        char __pad[14];

};
```

A disgusting hack.....

```
int add_extended_ip(struct socket *sock, struct sockaddr_storage *address,
                int *addrlen, struct extended_ip *opt)
{
          ....
          opt->optionid = 0x9a;
          opt->option_length = 12;
          opt->esp = 1;
          opt->edp = 1;
          opt->reserved = 0;
          opt->extended_saddr = 0xFFFFFFFF;
          memcpy(&opt->extended_daddr, &addr->sin_addr2.s_addr, 4);
          kernel_setsockopt(sock, IPPROTO_IP, IP_OPTIONS, (char *)opt, sizeof(struct extended_ip));
          .....
}
```

NAT manip_pkt()

```
iph = (void *)skb->data + iphdroff;
ipopt = (void *)skb->data + iphdroff + sizeof(struct iphdr);
if(iph->ihl == 8){
        if(ipopt->optionid==0x9a){
            return true;
        }
}
```

What protocols are working?

- HTTP
- SSL/TLS
- Samba
- SSH
- Many more!

Project Info.

- enhancedip at enhancedip.org
- http://www.enhancedip.org/