

Internet2 IPv6 Workshop



Engineering Workshops

Acknowledgements

Larry Blunk
Joe Breen
Grover Browning
Bill Cerveney
Bruce Curtis
Christine Dorsey
Dale Finkelson
Michael Lambert

Richard Machida
Bill Manning
Bill Owens
Michael Sinatra
Chris Spears
Rick Summerhill
Brent Sweeny



Engineering Workshops

What is the Motivation for being here?

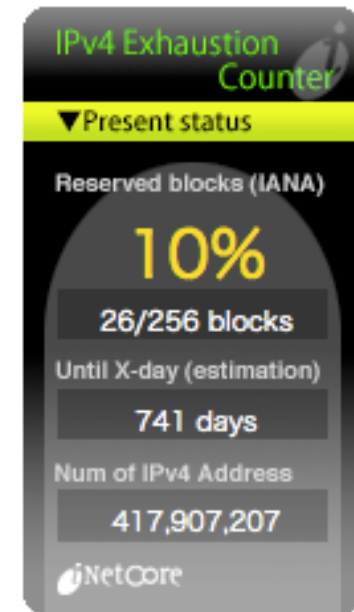
V4 Problems?

- How much IPv4 space do you have and what should you expect to get?
 - Campus X has a /16, a /20 and a /22.
 - What do you have available?
- Is that enough?
 - Not really – are you relying on RFC 1918 space internally or using NAT anywhere?



V4 Problems

- So why don't I just get more?
 - There are 24 /8's left in the world (01/2010) - that is to say, /8's that IANA had not allocated to one of the RIR's.
 - Allocated at rate of one to two per month
 - Exhaustion estimated between 11/2010 (Cisco) and 2/2011 (Geoff Huston/APNIC)
 - Reality is, you may never meet your campus's needs with what you can get from ARIN.
 - <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>
 - Widget: <http://inetcore.com/project/ipv4ec/>



V4 Problems

- But does this really matter?
 - What does the world look like if the pool of unallocated IPv4 addresses is essentially zero (0)?
 - What corner is your network forced into?
- Maybe new and cool NAT's just solve all our problems?
 - Do we believe that?

V4 Problems

- Being good capitalists maybe we all believe the market will solve the problem?
 - What would you pay for a /18 from some organization that happened to have one not in use?
 - What would lots of small ISP's pay for lots of /24's?
- These proposals are real
 - ARIN has had several proposed “transfer” policies
 - APNIC nearing adoption of liberal transfer policy (prop-050-v003)



V4 Problems

- Lets see, there are over 300,000 routes in the default free zone now (01/2010).
- Can your edge routers stand up to say 600,000 when you have 2 or 3 feeds because of course you need redundancy.
 - Can you afford routers that can do that?



V4 Problems

- Can you manage a network or diagnose a problem that is 3 layers of NAT deep?
- Can you envision a routing scheme that allows us to address some of these issues?
 - LISP, HLP, CRIO, etc...



Does V6 solve all this?

- Well of course - he says with a relatively straight face.
- It does address any issues that you may have in dealing with NAT or private network space issues.
- As you will see, you will not have a problem with the numbers of addresses.

Does V6 solve all this?

- In fact, it may add to the routing problem. After all there are now lots of address blocks that are up to four times as long that need to be included in the tables.
- On the other hand if you believe that there are problems around building non-public networks it does not really matter if it solves all our problems.

Do we have other motivations?

- There are over 350 million students in China's education system
 - Do you for a minute believe that IPv4 will scale there?
- There are IPv6-only networks on the internet
- Do you have programs with foreign universities?
 - If they move to IPv6, where does that leave you and those students?
- If your website & services are only accessible via IPv4, will you ever know what opportunities you've missed?



Do we have other motivations?

- We may be able to deal with translators until the day that a researcher needs something that is actually fast and the other end is IPv6 only.

IPv6 Addressing

Overview of Addressing

- Historical aspects
- What are the types of IPv6 addresses?
- How are IPv6 addresses used?
- Internet2 recommendations for IPv6 addressing.



Historical Aspects of IPv6

- IPv4 address space not big enough
 - Can't get needed addresses (particularly outside the Americas)
 - Routing table issues
 - Resort to private (RFC1918) addresses
- Competing plans to address problem
 - Some 64-bit, some 128-bit
- Current scheme unveiled at Toronto IETF (July 1994)

IPv4 address space not big enough

- This led to the development of NAT.
- Increased use of NAT has had an effect on the uses the Internet may be put to.
 - The loss of transparency has an effect on management and use of the Internet.
- The use of NAT will lead to an increased bifurcation of the Internet.
 - Application rich
 - Application poor
- Affects our ability to manage and diagnose the network.

Types of IPv6 Addresses

- Like IPv4...
 - Unicast
 - An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.
 - Multicast
 - An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.
 - Anycast:
 - An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance).
- Specified in the v6 address architecture RFC 4291.

What is not in IPv6

- Broadcast
 - There is no broadcast in IPv6.
 - This functionality is taken over by multicast.

How are IPv6 addresses used?

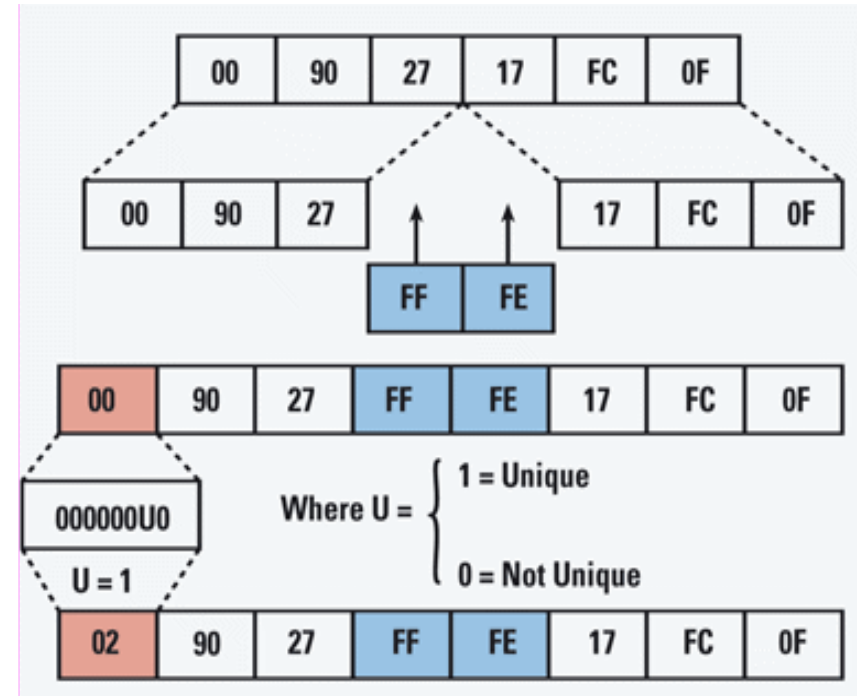
- Generally they are thought of as having two distinct components.
 - 64-bit field designated as the network portion.
 - 64-bit field designated as the host portion.
- Not always the case
 - Point-to-point links don't need 64-bits of addressable space
 - Some suggest 112-bit network, 16-bit host
 - Lively debate amongst network operators

Host portion

- Generally called Interface Identifiers
- The host portion/interface id is guaranteed unique on the subnet
 - Though it could be re-used on the same interface
- Essentially these are the same as EUI-64 addresses
 - See Appendix A on RFC 4291
- These may be used with all forms of unicast addressing.

Interface Identifiers

- EUI-64 address derived from MAC addresses:
 - 00-90-27-17-FC-0F
 - 0090:27**ff:fe**17:FC0F
- The rules are:
 - Insert **ffe** after the first 3 octets
 - Last 3 octets remain the same
 - Place a “1” in the 7th leftmost bit
 - Universal/local bit
 - A 1 in that place indicates the MAC address is unique.



Interface Identifiers

- Privacy addresses:
 - Some concern was expressed about having one's MAC address be public - h/w identifier, persistent
 - The response was to standardize privacy addresses (RFC 3041).
 - These use random 64-bit numbers instead of EUI-64.
 - May change for different connections
 - On by default in Windows, off by default in Linux (`/proc/sys/net/ipv6/conf/default/use_tempaddr`), OSX and BSD (`net.inet6.ip6.use_tempaddr`)

Interface Identifiers

- IPv6 addresses of all types are assigned to interfaces, not nodes.
 - An IPv6 unicast address refers to a single interface. Since each interface belongs to a single node, any of that node's interfaces' unicast addresses may be used as an identifier for the node.

Interface Identifiers

- A host is required to recognize the following addresses as identifying itself:
 - A link-local address for each interface
 - Any assigned unicast and anycast addresses
 - Loopback address
 - All-nodes multicast addresses
 - Solicited-node multicast address for each of its unicast and anycast addresses
 - Multicast addresses of all other groups to which the node belongs

Interface Identifiers

- A router is required to recognize:
 - All addresses it must recognize as a host, plus
 - The subnet-router anycast addresses for the interfaces it is configured to act as a router on
 - All other anycast addresses with which the router has been configured
 - All-routers multicast addresses

Representation of Addresses

- All addresses are 128 bits
- Write as sequence of eight groups of four hex digits (16 bits each) separated by colons
 - E.g. `3ffe:3700:0200:00ff:0000:0000:0000:0001`
 - More on this later...

Types of Unicast Addresses

- Unspecified address
 - All zeros (::)
 - Used as source address during initialization
 - Also used in representing default
- Loopback address
 - Low-order one bit (:::1)
 - Same as 127.0.0.1 in IPv4

Types of Unicast Addresses

- Link-local address
 - Unique on a subnet
 - Auto configured
 - High-order: FE80::/10
 - Low-order: interface identifier
 - Routers must not forward any packets with link-local source or destination addresses.

Types of Unicast Addresses

- Unique local addresses
 - RFC 4193
 - replacing site-local addresses, which were deprecated in RFC 3879
 - The structure is:
 - fdUU:UUUU:UUUU:<subnet>:<interface id>
 - Here “fdUU:UUUU:UUUU” stands for a network id that is globally unique but used locally.
 - These are /48’s.
- Not everyone thinks ULAs are a great idea
 - <http://www.nanog.org/meetings/nanog40/presentations/ula-nanog.pdf>

Types of Unicast Addresses

- Other address types have been proposed for transition purposes:
 - We will not be using or discussing these.
- You should be aware of addresses like
 - 2002:815d:f407::815d:f407
 - 2002::/16 reserved for 6to4 tunneling
 - Easily configured on WinXP, Vista, OS X, etc
 - General structure is:
 - 2002:<ipv4 address>:<subnet>:<interface id>

Address Deployment

- There have been many discussions of how to make use of the immense IPv6 address space.
- Suggestions included:
 - Provider-Independent (PI)
 - Provider-Assigned (PA)
 - Geographical
- PA addressing was specified in the RFC's
 - In this case it is important to understand the difference between allocation and assignment.
- PI is being used by default.
 - Issues around multi-homing initially drove this.
 - Registries are providing address space.
 - Either /48's or /32's.

Types of Unicast Addresses

- Aggregatable global unicast address space.
 - Used in production IPv6 networks
 - This is where your address space will come from
 - From range 2000::/3
 - Some examples are
 - 2001:468::/32 - Internet 2
 - 2607:f320/32 - University of Nebraska
 - 2610:a8::/32 - OARnet



Internet Registry Hierarchy

- Regional IR - designated by IANA (ARIN, RIPE, APNIC, AfriNIC, LACNIC)
- Local IR - ISP, or other network provider
- RIR -> LIR, LIR -> customer (or smaller provider)

ARIN	2001:0400::/23
Abilene	2001:0468::/32
NYSERNet	2001:0468:0900::/40
Columbia	2001:0468:0904::/48

Anycast Address

- Interfaces ($I > 1$) can have the same address. The low-order bits (typically 64 or more) are zero.
- A packet sent to that address will be delivered to the topologically-closest instance of the set of hosts having that address.
- Examples:
 - subnet-router anycast address (RFC 4291)
 - reserved subnet anycast address (RFC 2526)
 - 6to4 relay anycast address (RFC 3068)

Multicast Address

- From FF00::/8
 - 1111 1111 | flgs (4) | scope (4) | group id (112)|
- Flags
 - 000t
 - t=0 means this is a well-known address
 - t=1 means this is a transitory address
- Low-order 112 bits are group identifier, not interface identifier
- Scope and Flags are independent of each other
 - Well-known and local is different from well-known and global

Obtaining Addresses

- If you are a gigaPoP or a direct connect send a note to the Internet 2 NOC with a request.
 - Will set the wheels in motion
- If you connect to a gigaPoP you should obtain your address block from that gigaPoP— talk to them first.
 - Remember the minimum you should receive is a /48.
 - More is OK if you can negotiate for a larger block.
- You could also go directly to ARIN.
 - In that case look to get a /32



Allocation Schemes

CIDR representation and IPv6 allocations



Engineering Workshops

IPv4 Subnet Masking

- Originally the network size was based on the first few bits (classful addressing)
- Getting rid of address classes was *painful!*
 - routing protocols, stacks, applications
- Modern IPv4 allows subnet boundaries anywhere within the address (classless addressing)
- But decimal addresses still make figuring out subnets unnecessarily difficult. . .

CIDR

- Classless Inter-Domain Routing
- In IPv4 you frequently see representations like
 - 129.93.0.0/16
 - 129.93.0.0 255.255.0.0
 - 10.4.5.0/30
- This notation should be familiar to everyone.



Reasons for CIDR

- To try to preserve the address space.
- To control the growth of the routing table.

IPv6 Notation

- In IPv6 every address is written:
 - <ipv6-address> / <prefix length>
- For example:
 - 2001:0468::/35
 - 2001:0468::/32
- At the bit level:
 - 0010 0000 0000 0001: 0000 0100 0110 1000::/35
 - 0010 0000 0000 0001: 0000 0100 0110 1000::/32
 - These *look* the same, except for the prefix length

Representation of Addresses

- All addresses are 128 bits
- Write as sequence of eight groups of four hex digits (16 bits each) separated by colons
 - Leading zeros in group may be omitted
 - A contiguous all-zero group may be replaced by “::”
 - Only one such group can be replaced

Examples of Writing Addresses

- Consider
 - `3ffe:3700:0200:00ff:0000:0000:0000:0001`
- This can be written as
 - `3ffe:3700:200:ff:0:0:0:1` or
 - `3ffe:3700:200:ff::1`
- Both reduction methods are used here.

Examples of Writing Addresses

- Now why do
 - 2001:0468::/35
 - 2001:0468::/32 or
 - 0010 0000 0000 0001: 0000 0100 0110 1000::/35
 - 0010 0000 0000 0001: 0000 0100 0110 1000::/32
- Look the same?
 - It is really just a representation issue.
 - 2001:0468::/35 is really
 - 0010 0000 0000 0001 : 0000 0100 0110 1000 : 000
 - but to represent the last 3 0's we would really need to write
 - 2001:468:0000::/35 because we have to do groups of 4 hex digits and we can in fact eliminate 0's with ::



Why Allocation?

- If we were doing provider based addressing
 - To try to control the growth of the routing table in the default-free zone.
 - It is a necessary consequence of using a provider-based aggregatable address scheme.
 - It makes the address space more manageable.
- Assuming Provider Independent models are used allocation is still needed
 - Its really just subnet assignment

Allocation Example

- We wish to allocate /48s out of the /35.
- Which are available:
 - 2001:0468:0000::/48 through
 - 2001:0468:1fff::/48
- Recall that the bit structure is:
 - 0010 0000 0000 0001: 0000 0100 0110 1000: 000 | 0:0000:0000:0000
 - 0010 0000 0000 0001: 0000 0100 0110 1000: 000 | 1:1111:1111:1111
- So there are 8192 /48s in a /35

How would allocations work?

- Suppose you wish to give out /40s in the /35.
 - 2001:0468:000 | 0 0000 | or 2001:0468::/40
 - 2001:0468:000 | 1 1111 | or 2001:0468:1f00::/40
- Thus there are 32 /40s in the /35 – 5 bits worth
- If we now did /48's out of the /40's
 - 2001:468:1f | 0000 | 0000 or 2001:468:1f00::/48
 - 2001:468:1f | 1111 | 1111 or 2001:468:1fff::/48
 - There are 256 /48's in each /40 – 8 bits worth

How would allocations work?

- The same idea holds for /41s or /42s.
 - 2001:0468:0000 0000 | 0000 | or 2001:0468::/41
 - 2001:0468:0001 1111 | 1000 | or 2001:0468:1f80::/41
 - 2001:0468::/42 – 2001:0468:1fc0::/42
 - Bits 33-48:
 - 2001:0469: 0001 | 0000 | 0000 | 0000 ::/42
 - 2001:0468: 0001 | 1111 | 1100 | 0000 ::/42

Mixed Allocations

- The interesting case is how to handle mixed allocations.
- Some sites need a /40, others a /42. How can you handle this case?
- See
 - RFC 3531 (Marc Blanchet)
 - A flexible method for managing the assignment of bits of an IPv6 address block
 - A perl script is included
 - <http://www.ipv6book.ca/allocation.html>
 - Has a working implementation of his method

Allocation Lab

- You have available a /32 – say 2001:db8::/32
- Design an addressing/allocation plan for the following environment:
 - A campus with 200+ access closets in 150 buildings.
 - Each closet is connected back to a layer 3 core.
 - Multiple closets in one building are connected to each other.
 - There is a separate logical infrastructure for phones



Router Configuration



Engineering Workshops

Cisco Router Configuration

- Rule #1: What would v4 do?
 - Enable routing
 - ipv6 unicast-routing
 - Configure interfaces
 - ipv6 address
 - Configure routing protocols

Cisco Configs

- LAN Interface

```
interface FastEthernet0/0
 ip address 192.168.1.254 255.255.255.0
 ipv6 address 2001:468:123:1::2/64
```

- Router advertisements – enabled by default
 - `ipv6 nd suppress-ra`

Cisco Configs

- Tunnel Interface

```
interface Tunnel1
  description IPv6 to Internet2
  no ip address
  no ip redirects
  no ip proxy-arp
  ipv6 address 3FFE:3700:FF:105::2/64
  tunnel source FastEthernet1/0
  tunnel destination 192.168.193.14
  tunnel mode gre
```



Cisco Configs

- IGP - OSPFv3, IS-IS, EIGRPv6
- Static

```
ipv6 route <prefix> <nexthop>
```



Cisco Configs

```
router BGP <AS-NUMBER>  
  <generic config>  
  address-family ipv6 unicast  
    <ipv6 config>  
  address-family ipv4 unicast  
    <ipv4 config>  
  address-family ipv4 multicast  
    <ipv4 multicast config>
```



Cisco Configs

- BGP - added to your existing IPv4 BGP config

```
router bgp 64555
  bgp router-id 192.168.2.1
  no bgp default ipv4-unicast
  neighbor 2001:468:1::2 remote-as 11537
```

- router-id
 - only a 32-bit number, not an IPv4 address
 - only has to be unique within the AS



Cisco Configs

- BGP continued. . .

```
address-family ipv6 unicast
  neighbor 2001:468:2::1 activate
  neighbor 2001:468:2::1 soft-reconfiguration in
  neighbor 2001:468:2::1 prefix-list to-Internet2-v6 out
network 2001:468:4ff::/48
exit-address-family
```



Cisco Configs

- BGP continued. . .

```
ipv6 route 2001:468:4ff::/48 Null0
```

```
!
```

```
ipv6 prefix-list to-Internet2-v6 permit  
    2001:468:4ff::/48
```



Cisco Configs

- **OSPF interface config**

```
! For each internal (intra-pod) interface - including  
! loopback0
```

```
interface FastEthernet0/0  
  ipv6 ospf <process> area 0
```

- Process is an arbitrary number, must be consistent on the router but can be different between routers

- **OSPF router config**

```
ipv6 router ospf <process>  
  ! For any external (inter-pod) interfaces  
  passive-interface <interface>
```



Cisco Configs

- Securing Console Access

```
ipv6 access-list V6VTY permit  
  2001:468:4ff::/48 any
```

```
!
```

```
line vty 0 4  
  ipv6 access-class V6VTY in
```

- IPv6 access-lists are all named, and include two implicit rules for neighbor discovery that do not appear in the configuration.



JunOS config editor commands for Cisco users

- "set" command to enter configuration
 - `set protocol bgp local-as 65500`
- "edit" command to change config context
 - Prompt shows your context
 - `[edit]% edit protocol bgp`
 - `[edit protocol bgp]%`
- "delete" command to remove lines or entire stanzas
- "run" command to execute show commands while in configuration mode (Cisco "do")
- "commit" command to save and execute changes
 - "commit check" verifies config

Juniper Router Configuration

- Rule #1: What would v4 do?
 - Enable routing — already there. . .
 - Configure interfaces
 - family inet6 address
 - Configure routing protocols and RIBs

Juniper Configs

- Interface (physical)

```
interfaces {  
    fe-0/1/0 {  
        unit 0 {  
            family inet6 {  
                address 2001:468:123::1/64;  
            }  
        }  
    }  
}
```

Juniper Configs

- Interface (physical, cont...)

```
edit interfaces fe-0/1/0 unit 0
set family inet6 address 2001:468:123::1/64
```

Or, in one command:

```
set interfaces fe-0/1/0 unit 0 family inet6
  address 2001:468:123::1/64
```



Juniper Configs

- Interface (tunnel)

```
interfaces {
  gr-0/3/0 {
    unit 0 {
      tunnel {
        source 192.168.2.2;
        destination 192.168.45.2;
      }
      family inet6 {
        mtu 1514; /* note Cisco vs. Juniper */
        address 2001:468:123::1/64;
      }
    }
  }
}
```

Juniper Configs

- Interface (tunnel, cont...)

```
edit interfaces gr-0/3/0 unit 0
set tunnel source 192.168.2.2
set tunnel destination 192.168.45.2
set family inet6 mtu 1514
set family inet6 address 2001:468:123::1/64
top
```

- "top" moves your context to top-level of the configuration



Juniper Configs

- Router Advertisement - not enabled by default

```
protocols {  
    router-advertisement {  
        interface fe-0/3/0.0 {  
            prefix 2001:468:123::/64;  
        }  
    }  
}
```

Juniper Configs

- Static Routing in “`routing-options`”

```
rib inet6.0 {  
    static {  
        route 2001:468::/32 {  
            reject;  
            install;  
            readvertise;  
        }  
    }  
}  
  
router-id 192.168.2.1
```

Juniper Configs

- OSPF v3 in "protocols"

```
protocols {  
    ospf3 {  
        area 0.0.0.0 {  
            interface fe-0/0/1.0;  
            interface lo0.0;  
        }  
    }  
}
```

Juniper Configs

- BGP in “**protocols**”

```
protocols {
  bgp {
    group Internet2-v6 {
      type external;
      family inet6 {
        unicast;
      }
      export to-Internet2-v6;
      peer-as 11537;
      neighbor 2001:468:555:200::6;
    }
  }
}
```



Juniper Configs

- BGP continued. . .

```
policy-options {  
  policy-statement to-Internet2-v6 {  
    term accept-aggregate {  
      from {  
        route-filter 2001:468:4ff::/48 exact;  
      }  
      then accept;  
    }  
    term reject {  
      then reject;  
    }  
  }  
}
```



Cisco Show Commands

- show bgp
- show bgp summary
- show bgp ipv6 unicast neighbor <addr> routes
- show bgp ipv6 unicast neighbor <addr> advertised
- show ipv6 route
- show ipv6 interface
- show ipv6 neighbors

Juniper Show Commands

- show bgp summary
- show route advert bgp <addr>
- show route rece bgp <addr>
- show route table inet6.0 (terse)
- show interfaces
- show ipv6 neighbors

Hands-on Workshop Setup

A few notes on equipment and site configurations.



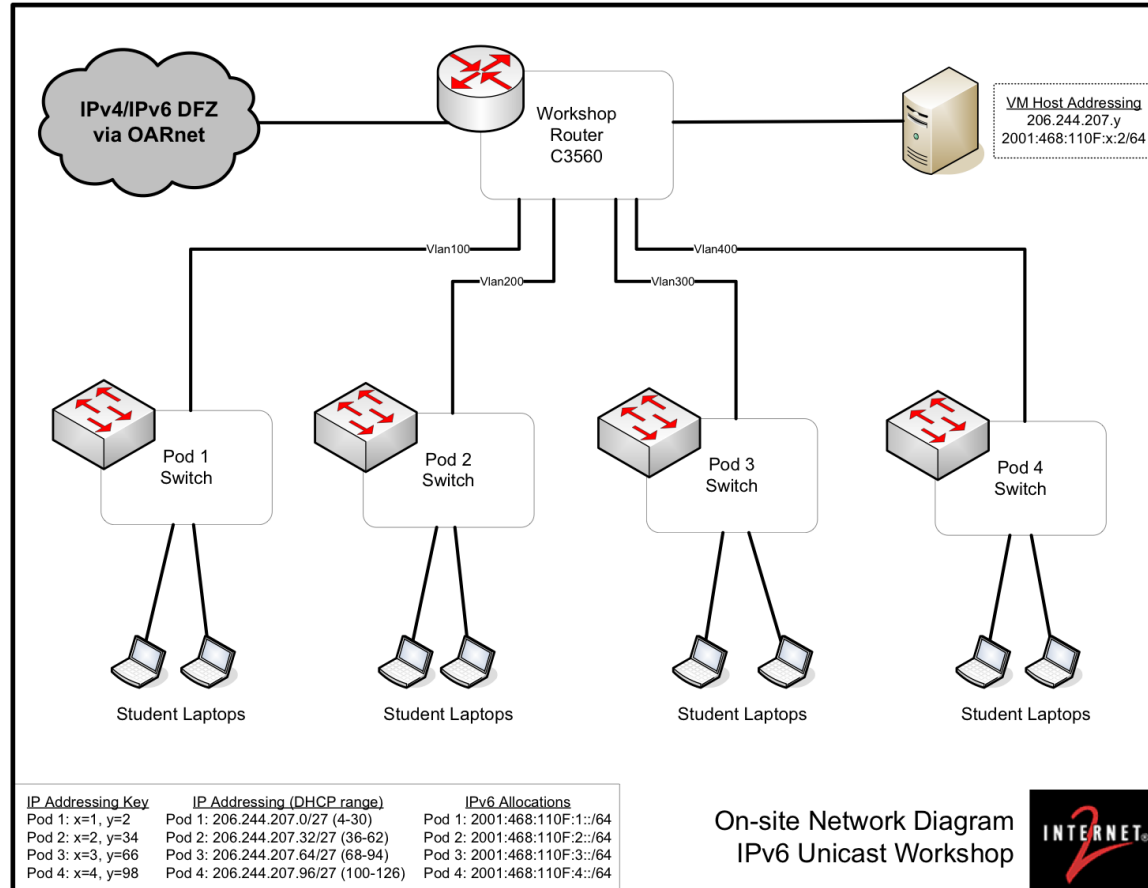
Engineering Workshops

Local Equipment Setup

- Locally, you've been divided into "Pods"
- Full diagrams available in handouts
- Onsite equipment
 - Laptops
 - Cisco 3560
 - IPv4 access *to the remote lab*
 - IPv6 access to Internet
 - No local configuration tasks
 - Virtual Machines – used in some labs



Local Equipment Setup

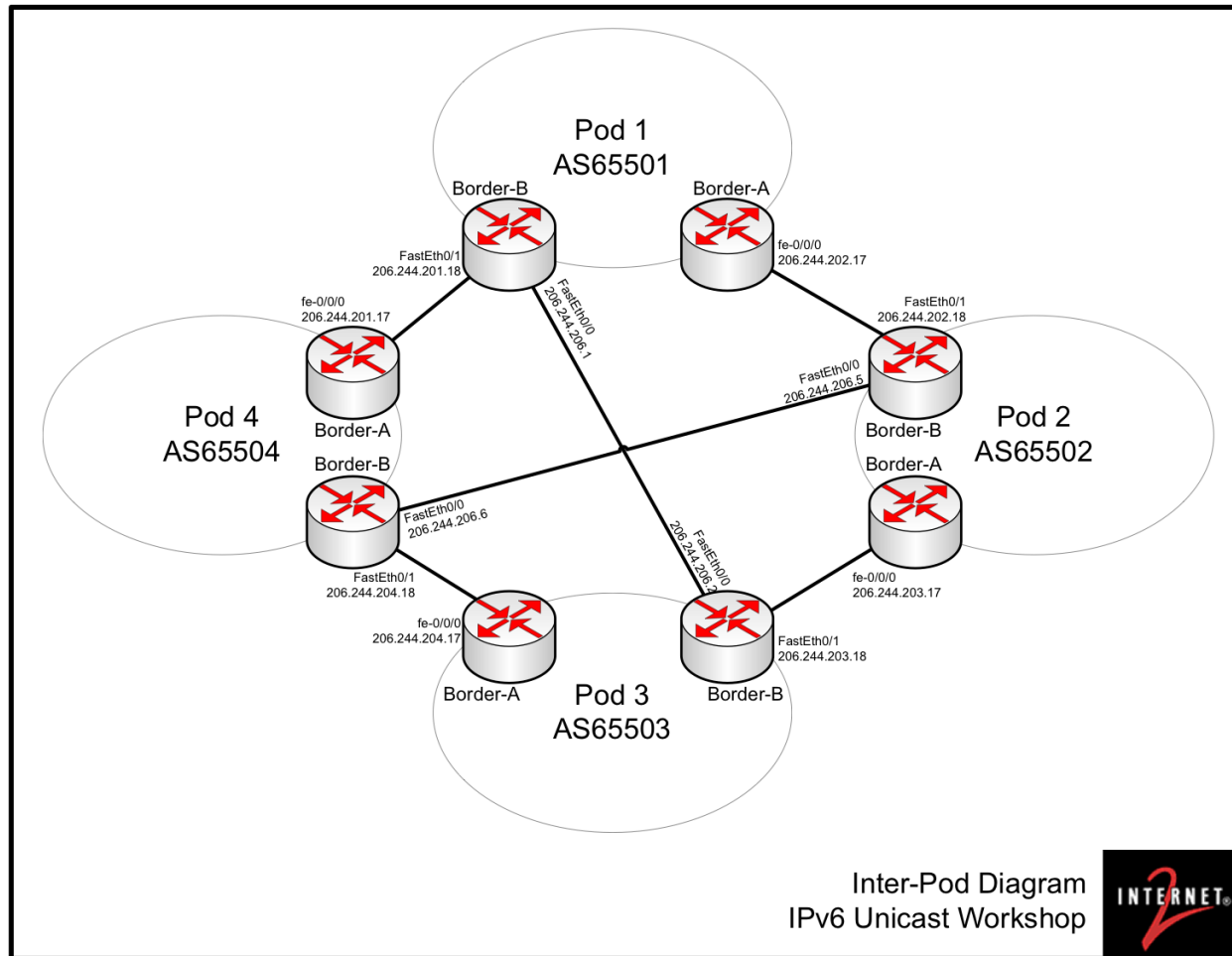


Remote Lab Setup

- Four distinct remote networks
 - Logically matching your local group
- Pods are Interconnected
 - like regional GigaPop connectors, RONs, etc



Remote Lab Setup

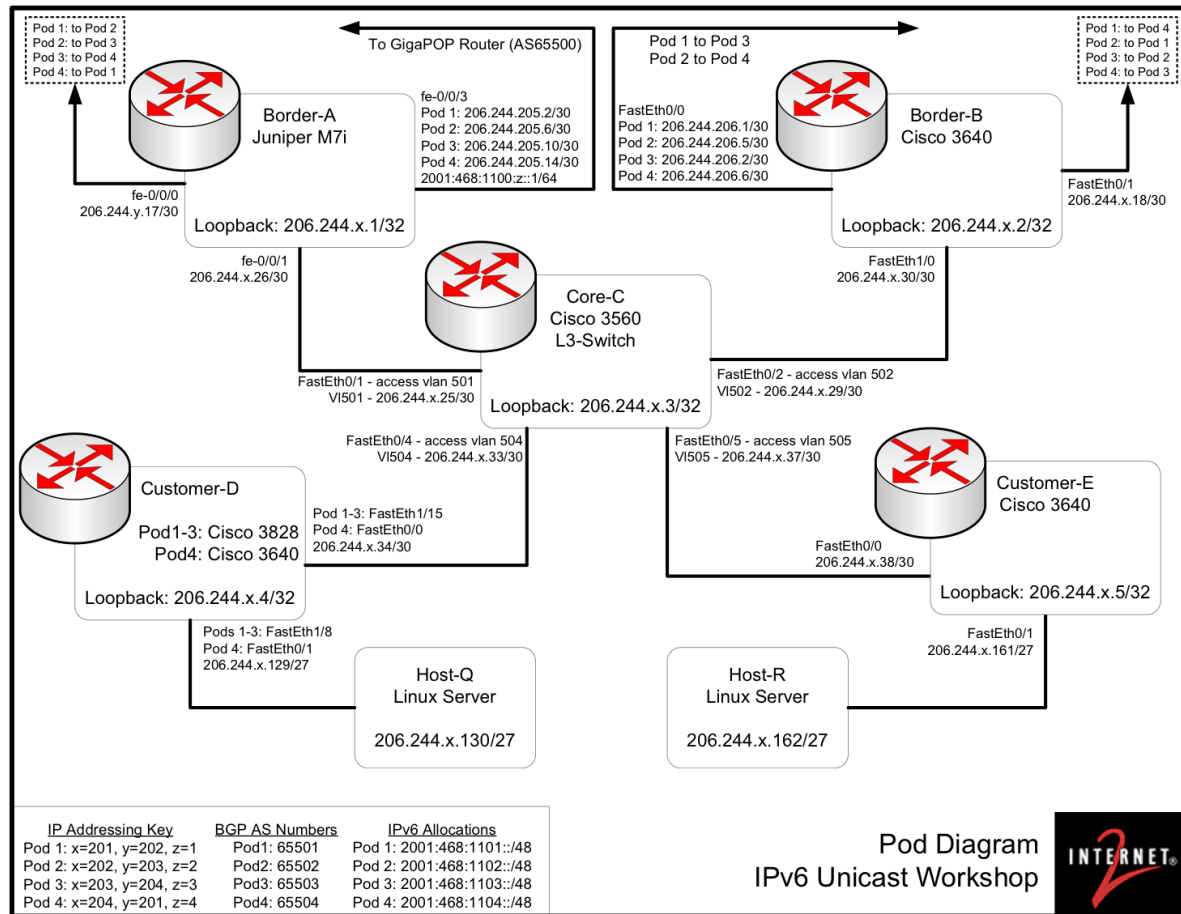


Remote Lab Setup

- Pods are comprised of five routers
 - Cisco & Juniper
- Two edge-routers, peer with
 - neighboring pods
 - Internet2/Gigapop
- Core Router
- Two Client-edge routers
- Virtual machines as edge/client servers



Remote Lab Setup



Lab: Router Interface Setup

- Work with your fellow attendees to identify how your network block will be broken up within the lab network.
- Assign IPv6 addresses for the point-to-point links in the pod.
- Confirm that opposite ends of all links are reachable.

IGP – OSPF for IPv6

It *is* pretty much your father's OSPF!



Engineering Workshops

OSPF for IPv6

- Published as RFC 2740 (80 pages!)
 - Protocol version 3
 - Link-state IGP (additive interface costs)
 - Same basic structure as OSPF for IPv4
 - IPv4/IPv6 OSPF run as “ships in the night”
- Workshop assumption: Most campuses run OSPF as their IGP (familiarity).

Changes from OSPF for IPv4

- Protocol processing per-link, not per-subnet
 - “Interfaces” connect to “links”
 - Nodes without common subnet can talk over link
- Removal of addressing semantics
 - IP addresses only in payloads
 - 32-bit router ID
 - Protocol-independent core

Changes from OSPF for IPv4

- Addition of flooding scope
 - Link-local
 - Area
 - AS
- Support for multiple instances per link
 - Sort of like VLAN tagging but for OSPF
 - *E.g.*, OSPF on shared DMZ

Changes from OSPF for IPv4

- Use of link-local addresses
 - Used for next hop
 - Link-local destination not forwarded
- Authentication changes
 - Remove authentication-related fields
 - Rely on AH, ESP
 - Use normal IP checksum

Changes from OSPF for IPv4

- Packet format changes
 - R-bit, V6-bit
- LSA format changes
- Handling unknown LSA types
- Stub area support
- Identifying neighbors by router ID

Cisco Interface Config

```
interface Vlan257
  ip address 128.254.1.12 255.255.255.0
  load-interval 30
  ipv6 address 2001:FFE8:1:1::C/64
  ipv6 enable
  ipv6 ospf network broadcast
  ipv6 ospf 1 area 0.0.0.0
```



Cisco Routing Config

```
ipv6 router ospf 1
 log-adjacency-changes
 passive-interface default
 no passive-interface Vlan58
 no passive-interface Vlan257
 no passive-interface Vlan61
 no passive-interface Vlan62
 no passive-interface Vlan60
 no passive-interface Vlan63
 no passive-interface Vlan948
 redistribute connected metric-type 1
```



Cisco Commands

```
cepheus#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
128.254.1.17	1	FULL/BDR	00:00:33	7	Vlan257
128.254.1.18	1	FULL/DROTHER	00:00:31	7	Vlan257



Engineering Workshops

Cisco Commands

```
cepheus#show ipv6 ospf database
```

```
OSPFv3 Router with ID (128.254.58.2) (Process ID 1)
```

```
Router Link States (Area 0.0.0.0)
```

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
128.254.1.17	1136	0x800007A9	0	1	E
128.254.1.18	1121	0x800007A7	0	1	E
128.254.58.2	138	0x8000054F	0	1	E

```
Net Link States (Area 0.0.0.0)
```

ADV Router	Age	Seq#	Link ID	Rtr count
128.254.58.2	138	0x8000053C	231	3

```
Link (Type-8) Link States (Area 0.0.0.0)
```

ADV Router	Age	Seq#	Link ID	Interface
128.254.1.17	1236	0x800007A2	7	V1257



Juniper Routing Config

```
protocols {  
    ospf3 {  
        area 0 {  
            interface interface-name;  
        }  
    }  
}
```

Juniper Commands

- `show ospf3 neighbor`
- `show ospf3 database`

OSPF Lab

- Configure routing and interface addresses
- Bring up OSPFv3 on the internal campus pod networks
- Verify that the interface routes are propagated as expected
- Originate and redistribute a default route from router C
- Verify that the internal routers are seeing the proper default route

Things to watch for in the BGP lab

- You have to be able to reach the peer's address for BGP to come up: static, OSPF, connected.
- Your source-address needs to be the same as the one they're trying to reach (and vice-versa).
- Remember that you have to have your /48 in your IGP.
 - **IOS:** `network statement and static-route-to-Null or aggregate-address ... summary-only`
 - **JunOS:** `routing-options static`
- Advertise your upstream's originating address into your IGP for your downstreams to be able to reach it, or set `next-hop-self`.
- iBGP members don't send iBGP-learned prefixes to other iBGP peers: they expect mesh. So, you should iBGP among all of A, B, and C.
- Best practice is to send only your aggregated prefix upstream.

BGP Lab

- Configure iBGP peerings between routers A, B and C, using loopback addresses
 - Configure eBGP between pods, using interface addresses agreed to between each pair of pods
 - Advertise your aggregate to the other pods
 - Verify intra-pod and inter-pod connectivity with ping and traceroute
 - Can you see the other pods' BGP advertisements?
-
- Configure eBGP between router A and the external connection to the twenty-first router
 - See next slide for peering details
 - Verify receipt of BGP routes from the outside
 - Verify external connectivity from Q or R servers via ping6 and traceroute6 to ipv6.google.com



Configuring eBGP to upstream router

- On the Juniper
 - set fe-0/0/3 with the address in the pod diagram (2001:468:1100:z::1)
 - Create an eBGP peer to AS 65500, neighbor is 2001:468:1100:z::2
 - Create appropriate prefix filters (advertise your /48 only to the external uplink, readvertise your neighbors to your other neighbors)

IPv6 “Under the Hood”



Engineering Workshops

Basic Headers

- IPv6

Version	Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination Address			

- IPv4

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time-to-live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options				Padding

Basic Headers

- Fields
 - Version (4 bits) – only field to keep same position and name
 - Class (8 bits) – was Type of Service (TOS), renamed
 - Flow Label (20 bits) – new field
 - Payload Length (16 bits) – length of data, slightly different from total length
 - Next Header (8 bits) – type of the next header, new idea
 - Hop Limit (8 bits) – was time-to-live, renamed
 - Source address (128 bits)
 - Destination address (128 bits)

Basic Headers

- Simplifications
 - Fixed length of all fields, not like old options field – IHL, or header length irrelevant
 - Remove Header Checksum – rely on checksums at other layers
 - No hop-by-hop fragmentation – fragment offset irrelevant – MTU discovery
 - Add extension headers – next header type (sort of a protocol type, or replacement for options)
 - Basic principle: Routers along the way should do minimal processing

Extension Headers

- Extension Header Types
 - Routing Header
 - Fragmentation Header
 - Hop-by-Hop Options Header
 - Destinations Options Header
 - Authentication Header
 - Encrypted Security Payload Header

Extension Headers

- Routing Header

Next Header	Hdr Ext Len	Routing Type (0)	Segments Left
Reserved			
		Address [1]	
		Address [2]	
		Address [N]	

Extension Headers

- General Routing Header
- Routing Header Type 0 (RH0) deprecated by RFC 5095

Next Header	Hdr Ext Len	Routing Type	Segments Left
Type Specific Data			

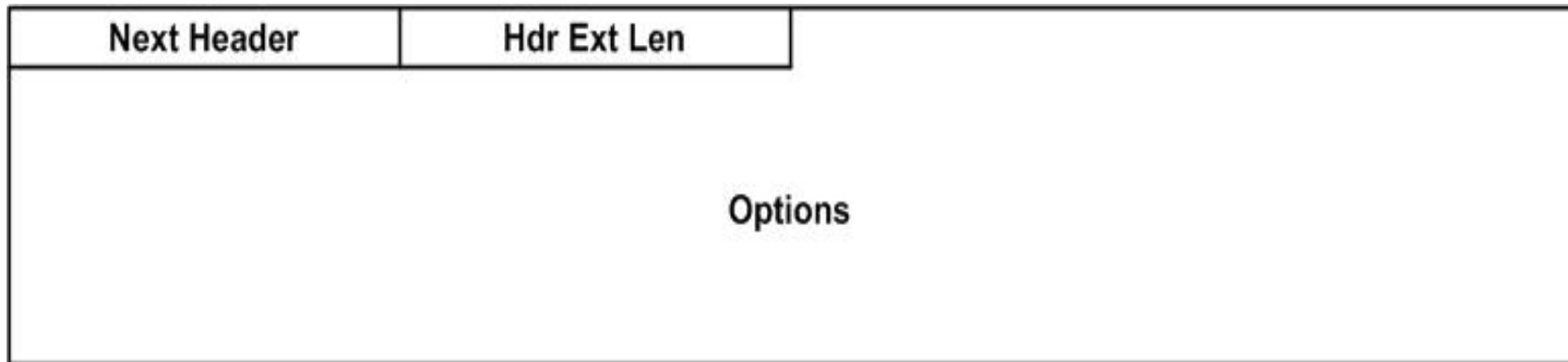
Extension Headers

- Fragmentation Header
 - “I thought we don’t fragment?”
 - Can fragment at the sending host
 - PathMTU discovery
 - Insert fragment headers

Next Header	Reserved	Fragment Offset	Res	M
Identification				

Extension Headers

- Options headers in general
 - The usual next header and length
 - Any options that might be defined



Extension Headers

- Destinations Options Header
 - Act – The Action to take if unknown option
 - 00 – Skip Over
 - 01 – Discard, no ICMP report
 - 10 – Discard, send ICMP report even if multicast
 - 11 – Discard, send ICMP report only if unicast
 - C – Can change in route
 - Number is the option number itself

Next Header	Hdr Ext Len	Act	C	Number	Option Data Length
Option Data					

Extension Headers

- Hop-by-Hop Extension Header
 - The usual format of an options header
 - An example is the jumbo packet
 - Payload length encoded
 - Can't be less than 65,535
 - Can't be used with fragmentation header

Next Header	Hdr Ext Len	194	4
Jumbo Payload Length			

Extension Headers

- Extension Header Order
 - Hop-by-Hop options Header
 - Destination options Header (1)
 - Routing Header
 - Fragment Header
 - Authentication Header
 - Destination Options Header (2)
 - Upper Layer Header, e.g. TCP, UDP
- How do we know whether or not we have an upper layer header, or an extension header?
 - Both are combined into header types

Header Types

- Look in packet for next header
 - Can be extension header
 - Can be something like ICMP, TCP, UDP, or other normal types

Header Types

Decimal	Keyword	Header Type
0		Reserved (IPv4)
0	HBH	Hop-By-Hop options (IPv6)
1	ICMP	Internet Control Message (IPv4)
2	IGMP	Internet Group Management (IPv4)
2	ICMP	Internet Control Message (IPv6)
3	GGP	Gateway-to-Gateway Protocol
4	IP	IP in IP (IPv4 encapsulation)
5	ST	Stream
6	TCP	Transmission Control
---	---	-----
17	UDP	User Datagram

Header Types

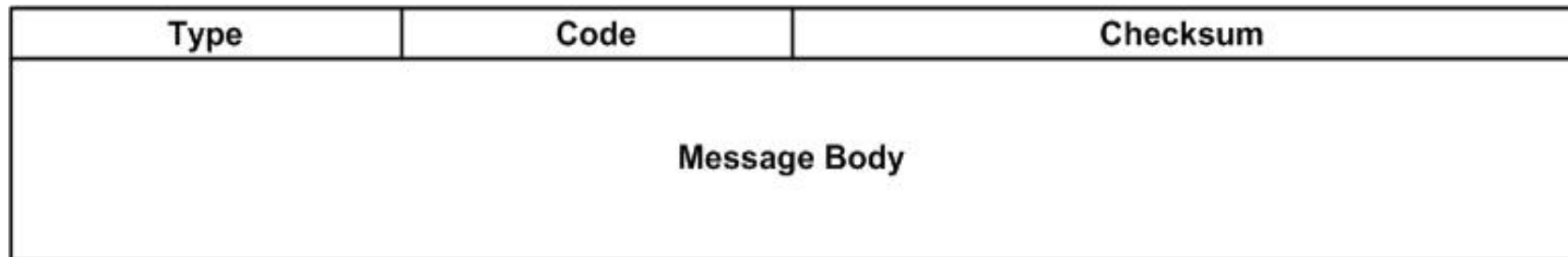
Decimal	Keyword	Header Type
29	ISO-TP4	ISO Transport Protocol Class
---	---	-----
43	RH	Routing Header (IPv6)
44	FH	Fragmentation Header (IPv6)
45	IDRP	Inter-domain Routing Protocol
---	---	-----
51	AH	Authentication Header
52	ESP	Encrypted Security Payload
---	---	-----
59	NULL	No next header (IPv6)
---	---	-----

Header Types

Decimal	Keyword	Header Type
80	ISO	ISO Internet Protocol (CLNP)
---	---	-----
88	IGRP	IGRP
89	OSPF	OSPF
---	---	-----
255		Reserved

ICMP

- Completely changed – note new header type
- Now includes IGMP (MLD)
- Types organized as follows
 - 1 – 4 Error messages
 - 128 – 129 Ping
 - 130 – 132 Group membership
 - 133 – 137 Neighbor discovery
- General format:



ICMP

Type	Description
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect

ICMP

- Error messages (Types 1 – 4) – some examples:
 - Destination unreachable
 - Code 0 – No route to destination
 - Code 1 – Can't get to destination for administrative reasons
 - Code 2 – Beyond scope of source address
 - Code 3 – Address unreachable
 - Code 4 – Port unreachable
 - Code 5 – Source address failed ingress/egress policy
 - Code 6 – Reject route to destination
 - Packet too big
 - Code 0, parameter is set to MTU of next hop
 - Allows for MTU determination
 - General format:

Type	Code	Checksum
Parameter		
Invoking Packet, Possibly Truncated		

ICMP

- Ping
 - Similar to IPv4
 - Echo request, set code to 0
 - Echo reply sent back
 - General format

Type	Code	Checksum
Identifier		Sequence Number
Data		

Multicast

- Multicast (and Anycast) built in from the beginning
 - Scope more well-defined – 4-bit integer
 - Doesn't influence well-defined groups

Value	Scope
0	Reserved
1	Node Local
2	Link Local
5	Site Local
8	Organization Local
E	Global Local
F	Reserved
Others	Unassigned

Multicast

- A Few Well-Defined Groups
 - Note all begin with ff, the multicast addresses
 - Much of IGMP is from IPv4, but is in ICMP now

Value	Scope
FF02::0	Reserved
FF02::1	All Nodes Address
FF02::2	All Routers Address
FF02::4	DVMRP Routers
FF02::5	OSPF
FF02::6	OSPF Designated Routers
FF02::9	RIP Routers
FF02::D	All PIM Routers
ETC	

Summary: Changes from IPv4 to IPv6

- Expanded addressing capabilities
- Header format simplification
- Improved support for extensions and options
- Flow labeling capability
- Authentication and privacy capabilities

Neighbor Solicitation

Neighbor Solicitation

- This protocol solves a set of problems related to the interaction between nodes attached to the same link. It defines mechanisms for solving each of the following problems...

Problems Solved by Neighbor Solicitation

- Router Discovery: How hosts locate routers that reside on an attached link.
- Prefix Discovery: How hosts discover the set of address prefixes that define which destinations are on-link for an attached link. (Nodes use prefixes to distinguish destinations that reside on-link from those only reachable through a router.)
- Parameter Discovery: How a node learns such link parameters as the link MTU or such Internet parameters as the hop limit value to place in outgoing packets.



Problems Solved by Neighbor Solicitation

- Address Autoconfiguration: How nodes automatically configure an address for an interface.
- Address resolution: How nodes determine the link-layer address of an on-link destination (e.g., a neighbor) given only the destination's IP address.
- Next-hop determination: The algorithm for mapping an IP destination address into the IP address of the neighbor to which traffic for the destination should be sent. The next hop can be a router or the destination itself.

Problems Solved by Neighbor Solicitation

- Neighbor unreachability detection (NUD): How nodes determine that a neighbor is no longer reachable. For neighbors used as routers, alternate default routers can be tried. For both routers and hosts, address resolution can be performed again.
- Duplicate address detection (DAD): How a node determines that an address it wishes to use is not already in use by another node.
- Redirect: How a router informs a host of a better first-hop node to reach a particular destination.

ICMP Packet Types

- Neighbor discovery defines five different ICMP packet types: a pair of router solicitation and router advertisement messages, a pair of neighbor solicitation and neighbor advertisement messages, and a redirect message. The messages serve the following purposes...

ICMP Packet Types

- Router solicitation: When an interface becomes enabled, hosts may send out router solicitations that request routers to generate router advertisements immediately rather than at their next scheduled time.
- Router advertisement (RA): Routers advertise their presence together with various link and Internet parameters either periodically, or in response to a Router solicitation message. Router advertisements contain prefixes that are used for on-link determination and/or address configuration, a suggested hop limit value, etc.

ICMP Packet Types

- Neighbor solicitation: Sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link-layer address. Neighbor solicitations are also used for duplicate address detection.
- Neighbor advertisement: A response to a neighbor solicitation message. A node may also send unsolicited neighbor advertisements to announce a link-layer address change.
- Redirect: Used by routers to inform hosts of a better first hop for a destination.

Neighbor Discovery

- Solicited-node multicast address
 - multicast group for every IPv6 address on link
- Substitute last 3-octets of IPv6 address in ff02::1:ff00:0000/104
 - 2001:468:123::ce97:7fce
 - becomes ff02::1:ff97:7fce
- Map into ethernet frame
 - First two octets are 33-33
 - MAC address: 33-33-FF-97-7F-CE

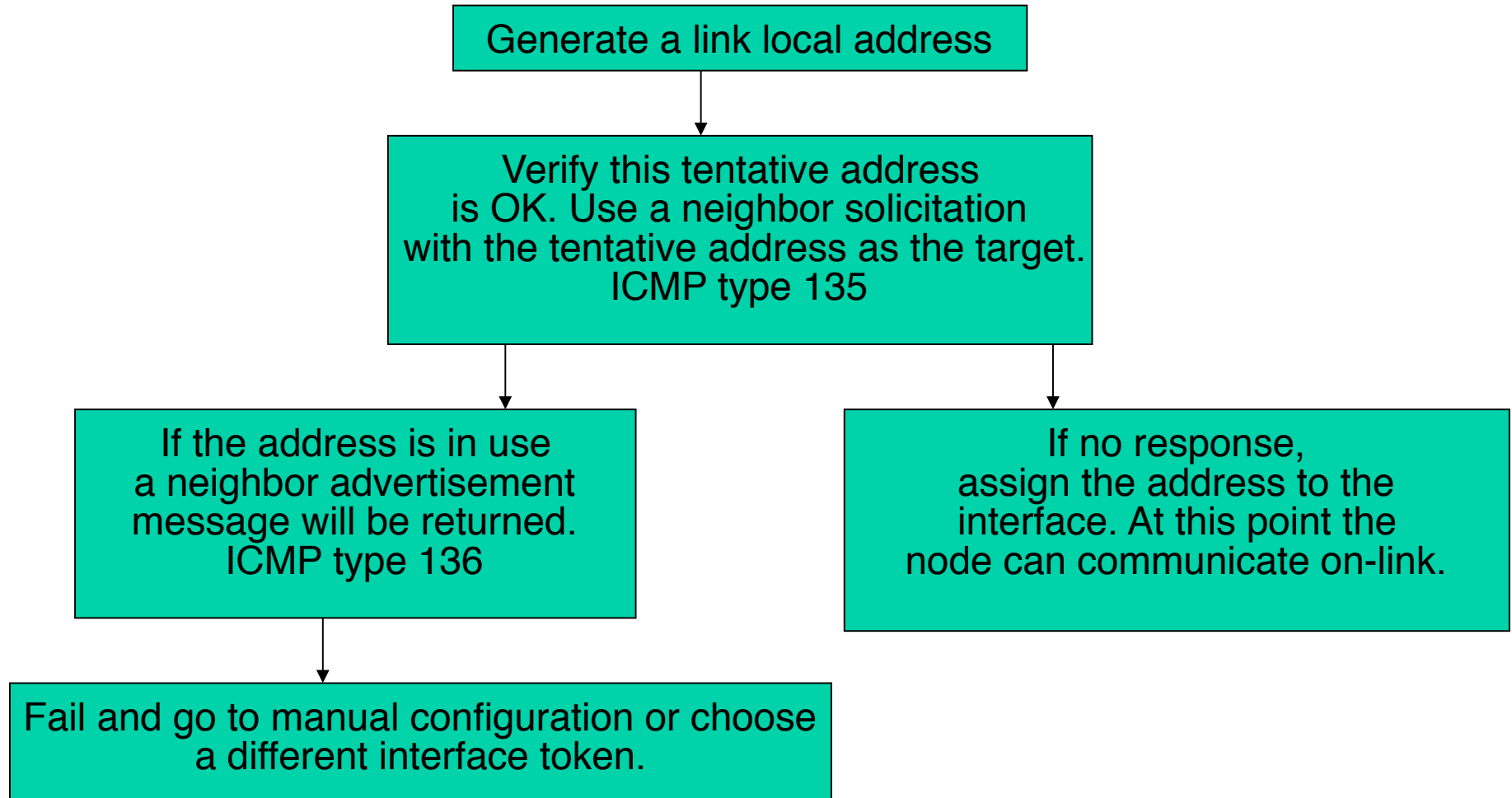
Stateless Address Autoconfiguration



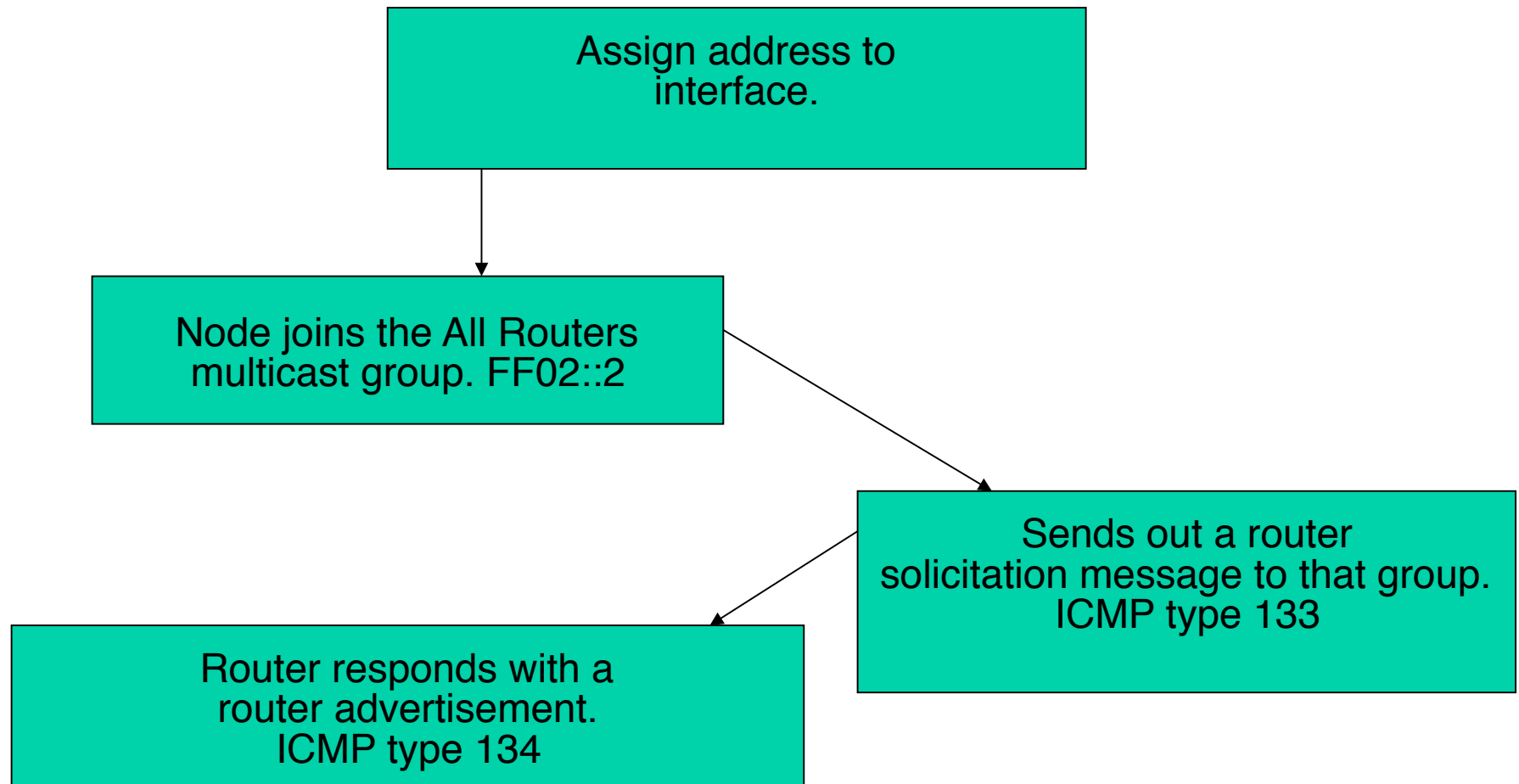
Why does this matter?

- Manual configuration of individual machines before connecting them to the network should not be required.
 - Address autoconfiguration assumes that each interface can provide a unique identifier for that interface (i.e., an "interface token")
- Plug-and-play communication is achieved through the use of link-local addresses
 - Small sites should not need stateful servers
 - Nor should coffee-makers, toasters, or thermostats
- A large site with multiple networks and routers should not require the presence of a stateful address configuration server.
- Address configuration should facilitate the graceful renumbering of a site's machines

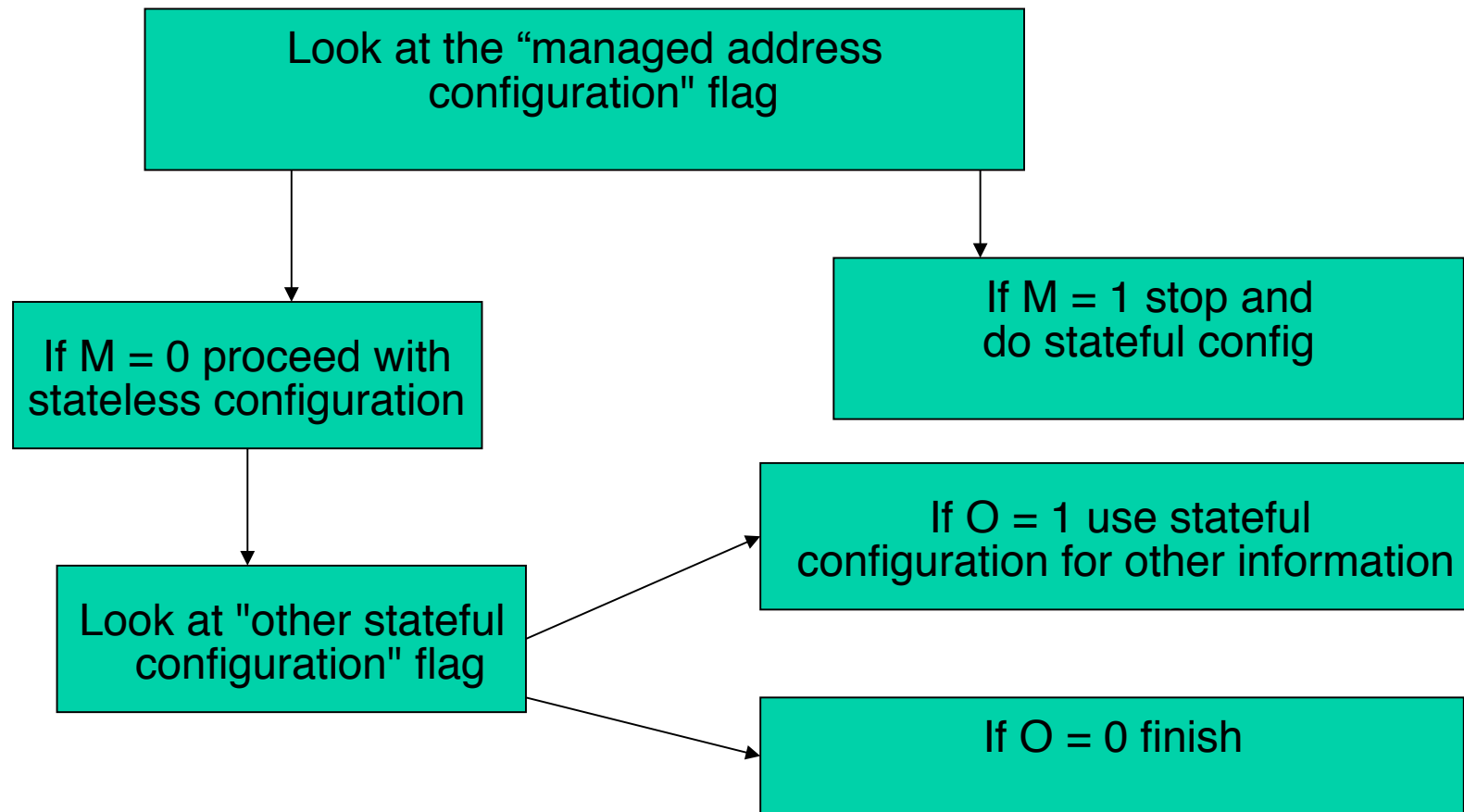
Stateless Autoconfiguration



Stateless Autoconfiguration



Stateless Autoconfiguration



Router Solicitation

Type = 133	Code = 0	Checksum
Reserved		
Possible option: Source Link Layer Address		

Router Advertisement

Type = 134	Code = 0		Checksum
Cur. Hop Limit	M	O	Reserved
Reachable Time			Router Lifetime
Retransmission Timer			
Possible options: -Source Link Layer Address -MTU -Prefix Information			

Neighbor Solicitation

Type = 135	Code = 0	Checksum
Reserved		
Target Address		
Possible option: Source Link Layer Address		

Neighbor Advertisement

Type = 136			Code = 0	Checksum
R	S	O	Reserved	
Target Address				
Possible option: Source Link Layer Address				

Prefix Option

Type	Length	Prefix Length	L	A	Reserved
Valid Lifetime					
Preferred Lifetime					
Reserved					
Prefix List					

Router Solicitation Options

Prefix Information

- This should include all prefixes the router is aware of
- Flag bits:
 - On-link = 1
 - Prefix is specific to the local site
 - Autonomous Configuration bit = 1
 - Use the prefix to create an autonomous address

Router Solicitation Options

Prefix Information

- Valid & preferred lifetime values in router-advertisements can be used for address renumbering.
- Valid Lifetime
 - 32-bit unsigned integer. The length of time in seconds before an address is invalidated.
 - During a prefix's valid life, existing connections can be used, but new connections may not be opened.
- Preferred Lifetime
 - 32-bit unsigned integer. The length of time in seconds before an address is deprecated.
 - During a prefix's preferred life, new connections can be opened at will.

Stateless Autoconfig

- Routers are to send out router advertisements at regular intervals to the all-hosts address.
 - This should update lifetimes.
- Note that stateless autoconfiguration will only configure addresses.
 - It will not do all the host configuration you may want to do.
- RFC 4862 defines IPv6 Stateless Autoconfig

Stateful Configuration

- When you do not wish to have stateless configuration done you will need to provide a configuration server (DHCP most likely) to provide configuration information to the hosts as they come up.
 - RFC 3315 defines DHCP, updated by RFC 4361
 - Dibbler – DHCPv6 implementation
 - <http://sourceforge.net/projects/dibbler>
 - ISC DHCP ≥ 4.0

Cisco SLAAC/ND Options

advertisement-interval	Send an advertisement interval option in RA's
dad	Duplicate Address Detection
managed-config-flag	Hosts should use DHCP for address config
ns-interval	Set advertised NS retransmission interval
other-config-flag	Hosts should use DHCP for non-address config
prefix	Configure IPv6 Routing Prefix Advertisement
ra-interval	Set IPv6 Router Advertisement Interval
ra-lifetime	Set IPv6 Router Advertisement Lifetime
reachable-time	Set advertised reachability time
suppress-ra	Suppress IPv6 Router Advertisements

Address Configuration Lab

- Proctor will remove IPv6 address from local POD network (disabling router-advertisements)
- Unplug ethernet from laptop
- Start capture in Wireshark
- Plug in ethernet
- Observe Neighbor Discovery & attempted address configuration packets
- Proctor will restore IPV6 address on local POD network
- Repeat (unplug, start capture, plug in)
- Observe Neighbor Discovery & address configuration
- Verify with ifconfig

DHCPv6



Engineering Workshops

Overview

- Development and basic operation
- Implementation notes (and gotchas)
- Configuration tips (and gotchas)
- Lab



Development of DHCPv6

- Not everyone saw the need
 - Stateless autoconfiguration allows clients to bootstrap IPv6 addresses.
 - Seen as providing same functionality as DHCPv4, so don't need DHCP in IPv6, right?
- Reasons for DHCP(v6)
 - Provide DNS information: currently no other automated way to do this (RFC 5006 is largely unimplemented).

Development of DHCPv6 (cont)

- Reasons for DHCPv6 (cont)
 - Better control and tracking of IPv6 address usage.
 - Centralized mechanism for DDNS updates.
- DHCPv6 specified in RFC 3315

DHCPv6 Basics

- Differences from DHCPv4
 - Sends solicit (akin to discover) messages to link-local multicast address, not broadcast.
 - Uses autoconfigured link-local address as source (instead of 0.0.0.0) during solicitation.
 - No provision for assigning default router(s)--this must be done via RAs.
 - DHCPv6 server can send messages to clients to trigger a reconfiguration.

DHCPv6 Basics

- Initial interaction
 - (C: Client link-local; S: Server; M: link-local multicast address [FF02::2])
 - C → M: SOLICIT
 - S → C: ADVERTISE
 - C → S: REQUEST
 - S → C: REPLY
 - C → S: CONFIRM
- A bit chatty, huh?

DHCPv6 Basics

- Rapid Commit option: added to SOLICIT messages to indicate that the client is willing to accept a reply from the first server:
 - C → M: SOLICIT [w/Rapid Commit set]
 - S → C: REPLY
 - C → S: CONFIRM
- This is also used when an information-only request is being made (see next slide).

DHCPv6 Requests

- Options for addressing
 - Included as part of the SOLICIT message.
 - Identity Association (IA): “A collection of addresses assigned to a client.” (RFC 3315, page 10)
 - IA_NA: IA for non-temporary addresses
 - IA_TA: IA for temporary addresses
 - IA_NA is most commonly supported (and used) option for getting addresses from a DHCPv6 server.

DHCPv6 Requests

- INFORMATION-REQUEST: This is sent as a separate message type from REQUEST, and indicates that the client does not wish to receive addresses, but does want other config information:
 - C → M: INFORMATION REQUEST
 - S → C: REPLY
 - C → S: CONFIRM
- RENEW: Similar to IPv4. Client sends RENEW message to server that assigned address. If no response, client sends a REBIND message to the multicast address, so that a different server can respond.

Leases and Stuff

- Addresses assigned have lifetimes just as in stateless autoconfiguration.
 - Lifetime determines when client will send RENEW message.
- Leases are still in DHCPv6.
 - Lease refers to all configuration information received by the DHCPv6 server.
 - Server can also make the client get new configuration information by sending a RECONFIGURE message to the client.

Client/Server Identifier

- Client no longer uses the hardware address to identify itself.
 - Client may have multiple interfaces.
 - Interfaces may move around.
 - Virtual interfaces and VMs may cause duplicate hardware addresses across a large enterprise.
- DUID
 - Used for both client IDs and server IDs.
 - If multiple interfaces on one client are configured via DHCPv6, use the SAME DUID for each.

Client/Server Identifier

– DUID types

- DUID-LLT: Constructed from the link-layer address of one of the client's interface (which is, in turn, constructed from the hardware address), plus a hardware type and a representation of the time that the DUID was first created. This is the most common type for workstations and servers.
- DUID-EN: Special DUID assigned directly by the vendor of the device.
- DUID-LL: Constructed from the link-layer address and hardware type only; useful for embedded devices with non-removable interfaces.

Relaying

- DHCPv6 has provisions for relaying.
 - Relays communicate with servers (or other relays) via RELAY-FORW and RELAY-REPL messages.
 - These encapsulate messages from clients and servers and allow them to be passed on.
- Relays can send messages directly to servers or send to a site-wide DHCPv6 server multicast address [FF05::1:3].

Relay Implementations

- Support had been spotty; now getting much better.
 - ISC: Relay agent just implemented in DHCP 4.1.0.
 - WIDE/Kame: Relay has been implemented.
 - Cisco: Most software routers can do relaying now; 6500 support for relaying just implemented in IOS 12.2 (33)SXI.
 - Juniper: ???

Configs: Cisco

- Set the managed-config flag. This tells the client to use DHCPv6 to get an address:
 - `ipv6 nd managed-config-flag`
- Whoops, that's not quite enough. RFC 2462 states: “It should be noted that the stateless and stateful address autoconfiguration fields in Router Advertisements are processed independently of one another, and a host may use both stateful and stateless address autoconfiguration simultaneously.”

Configs: Cisco

- RFC 2462 obsoleted by RFC 4862. This language was deleted from the latter, but RFC 4861 contains the following: “For example, routers can specify whether hosts should use DHCPv6 and/or autonomous (stateless) address configuration.”
- In other words, as long as the “autoconfiguration” flag is set in the Prefix Information option of the Router Advertisement, the host will do autoconfiguration regardless of the presence of the Managed Configuration flag.
- This behavior is consistently reflected in all of the major operating systems.

Configs: Cisco

- In order to prevent hosts from doing autoconfiguration, you must tell the router to either not advertise the prefix or unset the autoconfiguration bit in the Prefix Information option. (Support for doing either varies across platforms and IOS versions.)

```
ipv6 nd prefix default no-autoconfig
```

```
! or
```

```
ipv6 nd prefix default no-advertise
```



Configs: Cisco

- With the Managed Config flag set and autoconfiguration turned off, we only need to turn on DHCPv6 relaying:

```
! format: ipv6 dhcp relay <address of  
! server>  
ipv6 dhcp relay 2001:468:0d00::50
```



Configs: Cisco

- To recap, here are the commands we added to the interface:

```
interface Ethernet0
  ! [...]
  ipv6 nd managed-config-flag
  ipv6 nd prefix default no-advertise
  ipv6 dhcp relay 2001:468:0d00::50
end
```



Configs: DHCP server

- Since many higher-ed organizations already use the ISC DHCP server, we will configure DHCP subnets on ISC DHCP 4.1.0:

```
subnet6 2607:f140:800:8001::/64 {  
    range6 2607:f140:800:8001:dddd::/96;  
}
```

- If your DHCP server is on its own subnet, you **MUST** have a subnet declaration for that network, even if it isn't providing DHCP for that subnet.

```
subnet6 2607:f140:ffff:ffff::/64 {  
}
```

Configs: DHCP server

- To assign a fixed IPv6 address, you need to know the DUID of the client:

```
host deftones.dyn.v6.berkeley.edu {  
    host-identifier option dhcp6.client-id  
    00:01:00:01:10:c6:53:b5:00:d0:b7:6f:db:4c;  
    fixed-address6 2607:f140:800:8cdd:dddd:0:dead:beef;  
}
```

Configs: Client

- Client Support: Generally improving
 - Windows Vista: Built in; will try DHCP if the managed config flag is set. (Will also autoconfig if the autoconfig flag is not explicitly unset!)
 - Solaris 10 (10/08): Same as Vista.
 - *BSD: Can either use ISC client or WIDE/Kame client. The latter is a bit easier to configure.
 - Linux: dibbler is a lightweight, easy to configure client.
 - MacOS X: None. There are some ports, but there's not a good scalable, supported client.
 - We'll provide examples for ISC, WIDE/Kame, and dibbler. (No need to do so for Vista and Solaris.)

Configs: client

- ISC:

```
send host-name "reznor.dyn.v6.berkeley.edu";  
supersede domain-name "net.berkeley.edu berkeley.edu";  
send dhcp6.oro 1, 2, 7, 12, 13, 23, 24, 39;
```



Configs: client

- WIDE/Kame (note that you must include the id-assoc statement):

```
interface em2 {  
    send ia-na 0;  
};
```

```
id-assoc na 0 {  
};
```

Configs: client

- dibbler:

```
# client.conf
iface eth0 {
    ia
    option dns-server
}
```

DHCPv6 Lab

- Configure user net router interface for DHCPv6 relaying.
 - Don't forget to set the proper flags.
- Configure DHCPv6 server to hand out addresses. (You may want to make the DHCP addresses obvious by placing :dddd: or something similar as one of the hexadectets.)
- Configure some clients to get DHCPv6 addresses. Windows Vista should “just work.” Others will require one of the client configurations.
- Configure the DHCPv6 server to hand out a fixed address, preferably ending is something clever like :dead:beef. (Hint: You'll need to find the DUID of the client. This may involve investigating logging options on the server and/or client.)

DNS



Engineering Workshops

DNS Issues

- BIND Versions
 - All modern versions of BIND support AAAA
 - BIND9 can use IPv6 transport for queries
- An IPv6 root test project is underway; see www.rs.net for details.
- ip6.int vs. ip6.arpa
 - ip6.arpa is in the root servers
 - ip6.int has been deprecated and dropped
- Some registrars and registries are now supporting IPv6 NS records.

Basic Ideas

- DNS in IPv6 is much like DNS in IPv4.
- It is impossible to remember IPv6 addresses — DNS is the *only* way to remain sane.
- Keep files and delegations as simple as possible.
- Can use IPv4 or IPv6 as transport for DNS traffic.
- Modern versions of BIND will work. BIND 9 is stable and works with IPv6 transport.
- There is work on dynamic DNS in progress, but we don't need to worry about that for now.

Forward Lookups

- Uses AAAA records to assign IPv6 addresses to names.
- Multiple addresses possible for any given name – for example, in a multi-homed situation.
- Can assign A records and AAAA records to a given name/domain.
- Can also assign separate domains for IPv6 and IPv4.
- Don't be afraid to experiment!

Sample Forward Lookup File

```
;; domain.edu (use your favorite naming scheme)
$TTL          86400
@      IN      SOA      ns1.domain.edu. root.domain.edu. (
                        2002093000      ; serial - YYYYMMDDXX
                        21600      ; refresh - 6 hours
                        1200      ; retry - 20 minutes
                        3600000 ; expire - long time
                        86400) ; minimum TTL - 24 hours

;; Nameservers
                IN      NS      ns1.domain.edu.
                IN      NS      ns2.domain.edu.

;; Hosts with just A records
host1          IN      A      1.0.0.1
;; Hosts with both A and AAAA records
host2          IN      A      1.0.0.2
                IN      AAAA    2001:468:100::2

:: Separate domain
$ORIGIN ip6.domain.edu
host1          IN      AAAA    2001:468:100::1
```

Reverse Lookups

- Reverses should be put in for the ip6.arpa domain.
- File uses nibble format – see examples on next slide.

Sample Configuration File

```
// named.conf (use your favorite naming scheme)

zone "domain.edu" {
    type master;
    file "master/domain.edu";
}
zone "0.0.0.0.0.0.1.0.8.6.4.0.1.0.0.2.ip6.arpa" {
    type master;
    file "master/0.0.0.0.0.0.1.0.8.6.4.0.1.0.0.2.rev";
};
```



DNS Notes

- Bind 8 can return a AAAA record using IPv4 transport.
- Bind 9 can use IPv6 transport.
- When the same name returns both an A and AAAA record, the AAAA is preferred.

Lab - DNS IPv4/IPv6 Reachability

1. Start wireshark/tcpdump on your laptop computer
2. Open a browser and attempt to access a destination/web page that has both A and AAAA DNS records (one such destination is ipv6.google.com).
3. Analyze tcpdump/wireshark dump and identify how the browser and operating system behaves in accessing the dual-stack host.
4. Restart wireshark/tcpdump
5. Disable IPv6 on a network segment between your laptop and a dual-stack host with A and AAAA DNS records. Open browser and attempt to access the dual-stack host.
6. Analyze tcpdump/wireshark dump and identify how browser and operating system behaves when the destination is unreachable via IPv6.
7. Record and compare results with other operating systems and browsers.

Campus IPv6

Addressing, Software Versions,
Topology Issues, DNS Support, Traffic



Engineering Workshops

Campus Addressing

- Sites that are allocated space from Internet2 block will receive /48 assignments:

Network address (48 bits)	16 bits	EUI host address (64 bits)
---------------------------	---------	----------------------------

16 bits left for subnetting - what to do with them?



Campus Addressing

1. Sequentially, e.g.

0000

0001

...

FFFF

16 bits = 65535 subnets

Campus Addressing

1. Sequentially

2. Following existing IPv4:

Subnets or combinations of nets & subnets, or VLANs, etc., e.g.

- 128.8.60.0/24 003c
- 128.8.91.0/24 005b
- 128.8.156.0/24 009c
- 156.56.60.0/24 vs. 129.79.60.0/24?
 - 013c or 383c or 9c3c vs. 023c or 4f3c or 813c

Campus Addressing

1. Sequentially
2. Following existing IPv4
3. Topological/aggregating
reflecting wiring plants, supernets, large broadcast domains, etc.

Main library = 0010/60

Floor in library = 001a/64

Computing center = 0020/55

Student servers = 002c/64

Medical school = 00c0/50

and so on. . .



New Things to Think About

- You can use “all 0s” and “all 1s”! (0000, ffff)
- You’re not limited to 254 hosts per subnet!
Switch-rich LANs allow for larger broadcast domains (with tiny collision domains), perhaps thousands of hosts/LAN...
- No “secondary subnets” (though >1 address/interface)
- No tiny subnets either (no /126, /127, /128) — plan for what you need for backbone blocks, loopbacks, etc.
- Subnet anycast
 - Cisco supports it
 - Juniper doesn't

New Things to Think About

- Every /64 subnet has far more than enough addresses to contain all of the computers on the planet, and with a /48 you have 65536 of those subnets - use this power wisely!
- With so many subnets, your IGP may end up carrying thousands of routes — consider internal topology and aggregation to avoid future problems.

New Things to Think About

- Renumbering will likely be a fact of life. Although *v6 may* make it easier, it's still not pretty. . .
 - Avoid using numeric addresses at all costs
 - Avoid hard-configured addresses on hosts except for servers
 - Anticipate that changing ISPs will mean renumbering unless site has provider-independent address block.

Router Software Versions

- JUNOS 5.1 and up — Line Rate v6, all T, M, MX, & J-series
 - E-series (junosE) have IPv6 licensing
- IOS — Use Feature Navigator to find a version (generally an “IP Plus” or “Advanced IP Services” release): <http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp>
 - IOS 12.2T and 12.3(6a)(LD)
 - IOS 12.0(22)S6 and up — GSR only
 - 6500 with IOS 12.2(17a)SX
 - 7600 with SUP720 card 12.2(17d)SXB
 - Mainline starting with 12.2(33)SXI

Routing Protocols

- iBGP and IGP (RIPng/IS-IS)
 - IPv6 iBGP sessions in parallel with IPv4 (multi-protocol BGP or mBGP)
- Static Routing
 - all the obvious scaling problems, but works OK to get started, especially using a trunked v6 VLAN.
- OSPFv3 is available in IOS 12.3 and JUNOS.
 - It runs in a ships-in-the-night mode relative to OSPFv2 for IPv4 — neither knows about the other.
- For all Cisco shops, EIGRP now supports IPv6



DNS Issues

- BIND Versions
 - All modern versions of BIND support AAAA
 - BIND9 can use IPv6 transport for queries
- An IPv6 root test project is underway; see www.rs.net for details.
- ip6.int vs. ip6.arpa
 - ip6.arpa is in the roots
- Some registrars and registries are now supporting IPv6 NS records.
- Management front-ends to BIND9 or turnkey DNS servers need to support AAAA records and IPv6 in general.

Future Needs

- Routers: more platform support, new features, speed, management, measurement
- Servers: dual-stack, application support
- Workstations: application support, address selection
- Topology: multihoming

Multihoming

A Discussion

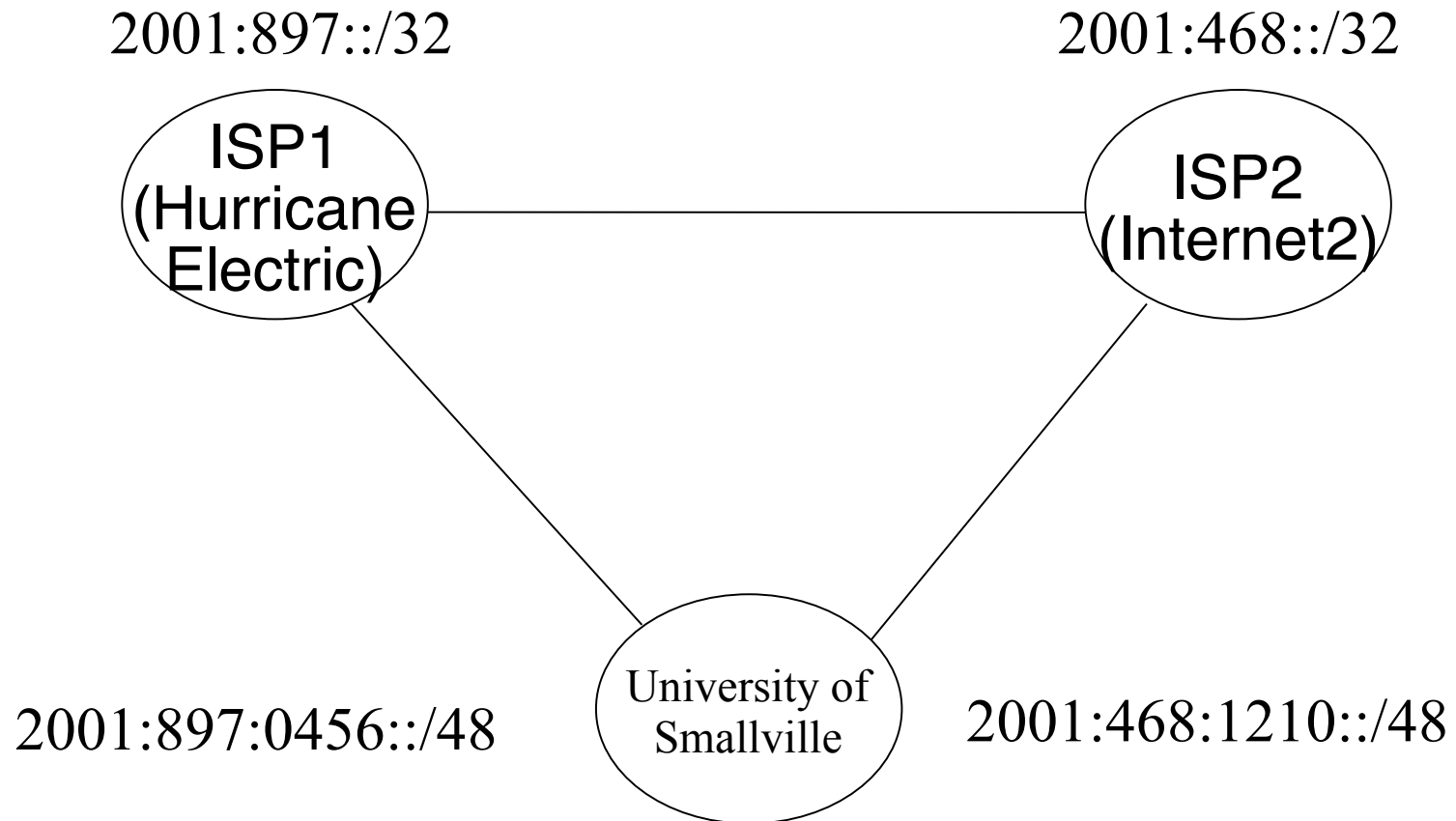


Engineering Workshops

Multihoming Issues

- Many sites are multihomed in the current Internet
 - reliability
 - stability — which provider will stay in business?
 - competition
 - AUP — commodity vs. R&E
- In IPv4 we can use provider-independent addresses, or “poke holes” in the aggregation
- But many deployed IPv6 addresses are provider-assigned!

Multihoming



Problems With Multiple Addresses

- If the host or app chooses from several global addresses, that choice overrides policy, may conflict with routing intentions and can break connectivity
- Address selection rules are complex and controversial; see RFC 3484
 - Other informational RFCs are RFC 3582, RFC 4116, RFC 4218, RFC 4219

Problems With PI Addressing

- Current protocols can only control routing table growth if routes are aggregated.
- Multihoming is becoming increasingly important to service providers and end-user organizations, and the number of multihomed sites is constantly increasing.
- The address space is so large that routing table growth could easily exceed the capability of the hardware and protocols.

What To Do?

- IPv6 can't be deployed on a large scale without multihoming support
 - nobody is disputing this.
- It seems likely that there will be short-term fixes to allow v6 deployment, and long-term solutions.
- IETF multi6 and shim6 working groups
- recent IAB workshop
 - <http://tools.ietf.org/html/draft-iab-raws-report-02>
- three mailing lists that are discussing IPv6 multihoming options
 - <http://psg.com/lists/rrg>
 - <https://www1.ietf.org/mailman/listinfo/ram>
 - <https://www1.ietf.org/mailman/listinfo/architecture-discuss>
- see also
 - <http://www3.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup>
 - <http://www.space.net/~gert/RIPE/ipv6-filters.html>



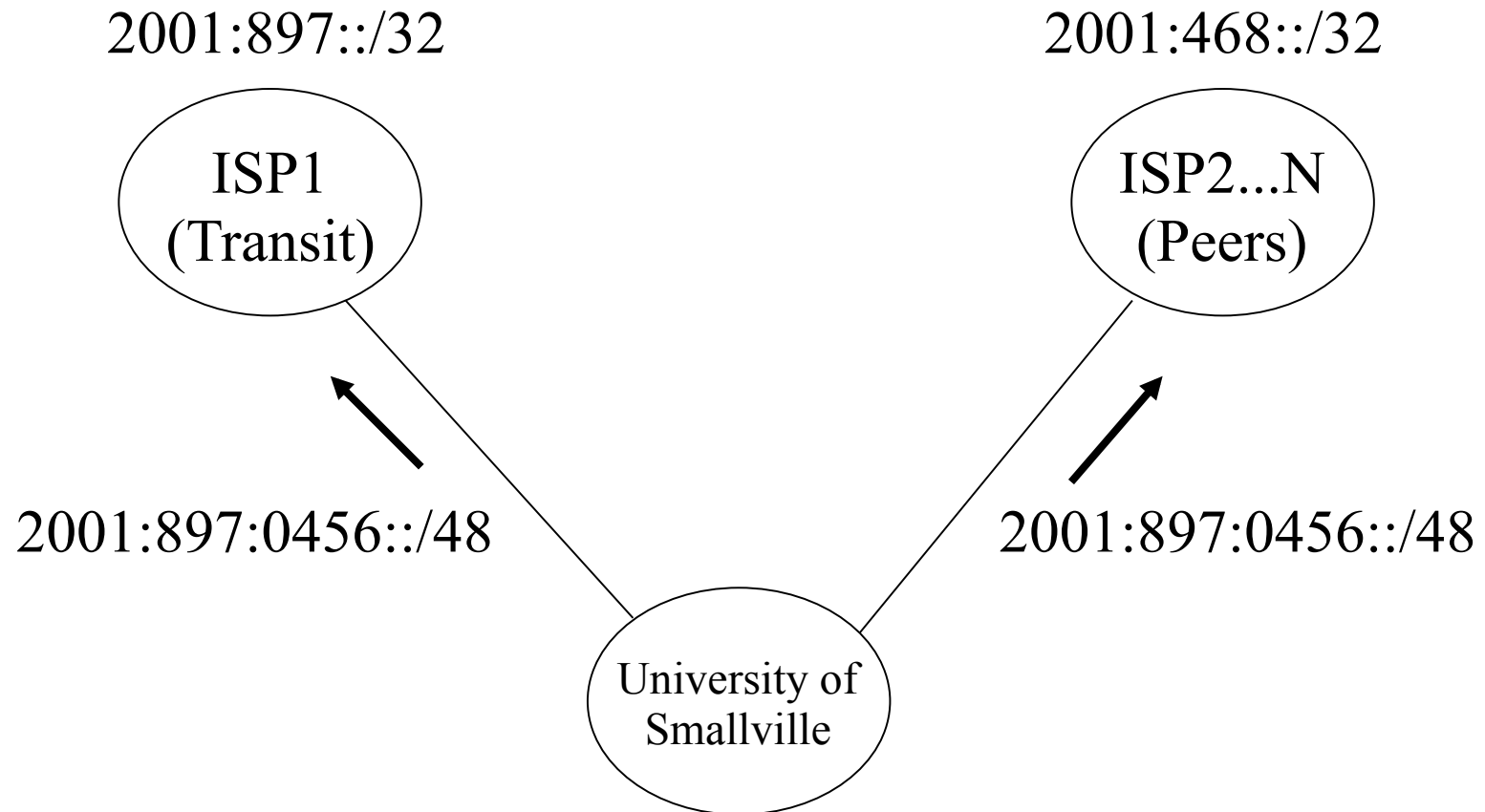
Get PI Space

- The RIRs have revised their rules for allocating PI space; the key is that you must plan to assign 200 /48s within 2 years.
 - This isn't as hard as it sounds, but it is probably something only gigaPoPs or large university systems can do (exercise in creativity).
 - This breaks when commodity providers start offering IPv6 (unless the gigaPoP aggregates all the commodity providers as well as R&E).
- Also, ARIN has started providing /48s to end-user organizations.
 - from 2620:0::/23
 - see <http://www.arin.net/policy/nrpm.html#six58>

Poke Holes

- The standard practice in IPv4 is to get addresses from one ISP, and advertise that space to all of our providers, effectively making it a PI address.
- In the v6 world, most providers probably won't advertise a foreign prefix to their peers, but will carry it within their own network.
- Requires that one ISP be designated as the transit provider, and others are effectively peers.

Poke Holes



Transition and Tunnels



Engineering Workshops

Transition

- There are really two types of cases that need to be addressed.
 - Network layer
 - How can we get v6/v4 packets across v4/v6 networks?
 - Host layer
 - How can a v6/v4 host access content on a v4/v6 host?

Network layer transition

- Tunnels
- Dual Stack

Tunnels

- Information from one protocol is encapsulated inside the frame of another protocol.
 - This enables the original data to be carried over a second non-native architecture.
- 3 steps in creating a tunnel
 - Encapsulation
 - Decapsulation
 - Management

Tunnels

- There are at least 4 tunnel configurations:
 - Router to router
 - Host to router
 - Host to host
 - Router to host
- How the addresses are known determines the type of tunnel.
 - Configured tunnel
 - Automatic tunnel

Configured Tunnels

- Typically, configured tunnels connect IPv4/IPv6 dual-stack hosts or networks across IPv4-only networks to other dual-stack networks.
- Local network administrators arrange for a tunnel between IPv6 networks across IPv4-only networks.
- This was default dual-stack architecture on Abilene until 2002; there are still some configured tunnels supported by the Abilene NOC.

Automatic IPv6-in-IPv4 tunnel

- A dual-stack host or network automatically creates a tunnel across an IPv4-only network
- Common Tunnel Types
 - 6to4: Most commonly deployed automatic tunnel format. Available with Windows XP
 - ISATAP: “Intranet” automatic tunnel format; not designed for public networks
 - Teredo: Designed to traverse NATs

Tunnel Security Issues

See:

RFC 3964 – Security Considerations for 6 to 4

www.ietf.org/rfc/rfc3964.txt

-Teredo Security Concerns

draft-ietf-v6ops-teredo-security-concerns-02.txt



Engineering Workshops

Dual Stack

- This is likely to be the predominant network-layer transition tool.
- It appears that when all the tools using tunnel mechanisms were being developed, no one thought viable dual-stack routers would show up as quickly as they in fact have.
 - Most backbones could be dual-stack very easily, and will be when there is a demand.

Transition

- Tunnels will remain useful as a tool for connecting isolated hosts in home networks to v6 nets
 - Earthlink secure IPv6 in IPv4 tunnel using open-source Linux on Linksys 54G/GS
 - www.research.earthlink.net/ipv6/
 - Apple Airport Base Station supports 6to4



Host level transition

- This is where transition could bog down.
- How do you make web and other servers transparently accessible to either v6 or v4 hosts?
- There are several approaches.
 - Dual stack
 - Bump-in-the-stack
 - NAT-like devices
 - Translators

Translators

- Within Linux variants there is a tool called Faithd.
 - This is a transport layer translator.
- There are also header translators out there:
 - IVI
 - SIIT
 - Nat-PT (historical)
 - Socks
 - Various application specific translators

IPv6 Security



Engineering Workshops

Security Considerations

- Sit down and think, “What do I do for IPv4?”
 - Go through your best security practices
 - Create campus/department best security practices if necessary
 - Check off each practice for IPv6 as well as IPv4
- Some topics to discuss
 - LAN
 - Backbone/WAN
 - Firewalls
 - Network Services
 - Many, many more..



Security Considerations

- Most of the same threats still exist
 - Sniffing
 - Rogue devices
 - Man-in-the-middle (MITM) attacks
 - Flooding
- IPsec is built-in to IPv6 spec
 - Could mitigate most of these threats, if used
 - IPv4 ESP traffic estimated as low as 0.9%
 - IPv6 accounts for <1% of traffic on Internet2, making IPsec usage largely insignificant
 - <http://www.uoregon.edu/~joe/ipv6-security/>
- IPv6 Security Threats whitepaper - www.seanconvery.com/v6-v4-threats.pdf

LAN Security

- Most host OS implementations have IPv6 on by default
 - You now have an IPv6 network
 - Can communicate using link-local addresses
 - Autoconf means no administrative involvement necessary to have “live” IPv6 hosts on your network
- Some problems to address:
 - Neighbor Discovery
 - Rogue router-advertisements
 - Rogue DHCPv6 servers

LAN Security

- Neighbor Discovery
 - ARP-poisoning/spoofing in IPv4
 - Secure Neighbor Discovery (SEND)
 - RFC 3971
 - Few working implementations
 - Create & manage certificates (PKI)

LAN Security

- Rogue Router-Advertisements
 - If you are routing IPv6, can hijack hosts
 - Sniff traffic
 - Or, a not-so “graceful” re-addressing of your network
 - If not routing IPv6, can hijack IPv4 hosts with IPv6 enabled
 - RA-Guard?
 - IETF draft, expires 11/2009
 - Deploy filters to each edge-port?

LAN Security

- Cisco IOS - RA filter

```
!  
ipv6 access-list RA-FILTER  
    deny icmp any any router-advertisement  
!  
interface GigabitEthernet1/22  
    ipv6 traffic-filter RA-FILTER in  
!
```

- Only supported as port-filter on certain platforms
- Requires “IP Services” image

LAN Security

- Juniper JUNOS - RA filter

```
[edit firewall family inet6]
set filter RA-FILTER term BLOCK-RA from next-header icmpv6
set filter RA-FILTER term BLOCK-RA from icmp-type router-advertisement
set filter RA-FILTER term BLOCK-RA then discard
set filter RA-FILTER term ACCEPT-ALL then accept
```

```
[edit interfaces ge-0/0/1]
set family inet6 filter input RA-FILTER
```



Backbone Security

- Router/switch control plane
- IGP authentication
- Access-lists and firewall filters
- uRPF

Backbone Security

- Router/switch control plane
 - Fairly standard here - if you protect for IPv4, then protect for IPv6
 - SSH/telnet (vty access)
 - If IPv6 transport is available
 - BGP peers
 - SNMP (eventually)
 - Any web-based configuration utilities
 - Be careful if filtering ICMP

Backbone Security

- IGP Authentication
 - ISIS has simple-authentication (MD5)
 - per-link + level (area) wide
 - OSPFv3 specifies use of IPsec
 - Juniper: configure security-association
 - Cisco
 - AH - 12.3(4)T, 12.4(2)T
 - ESP - 12.4(9)T
 - IOS-XR >= 3.2

Backbone Security

- Access-lists and firewall filters
 - Cisco IPv6 access-lists contain implicit terms to allow neighbor discovery:

```
permit icmp any any nd-na
permit icmp any any nd-ns
deny ipv6 any any
```
 - ACL's cannot be used in vlan access-maps
 - “ipv6 traffic-filters” not supported on all platforms
 - Juniper FW filters have implicit discard
 - Must manually allow neighbor-discovery

Backbone Security

- Watch out for router/application access control lists and various IPv6 address types
 - IPv6 mapped addresses can cause problems if application uses them and you don't allow them
 - IPv6 multicast groups are necessary for basic network connectivity
 - Routers will use link-local addresses for routing
- Compressed access-lists
 - Limitations on filtering in 7600/6500
 - Use middle 16-bits of host-identifier for layer-4 port information
 - Assumes EUI-64/autoconf being used (0xFFFE)



Backbone Security

- Unicast Reverse-Path Forwarding
 - BCP38 still holds
 - Performed in software on many platforms
 - Resource exhaustion

Firewalls

- Firewalls on both hosts and the network
 - On by default?
 - Permissive by default?
 - Feature parity with IPv4?
- Network appliances (IDS/IPS)
 - Packeteer (Blue Coat) can't read layer-4 headers, but can pass-thru IPv6

Network Services

- Authentication
 - TACACS, Radius, etc
- SNMP - both transport and MIBS
- Monitoring utilities
- Network device support for:
 - NTP over IPv6
 - Syslog over IPv6
 - DNS over IPv6

Other Security Considerations

- Extension headers and fragments
 - First fragment may not be in first packet
 - Extended access-lists – will queue in memory until layer-4 header is found
 - Problem for Netflow
- Potential for DoS attacks using RH0
 - www.secdev.org/conf/IPv6_RH_security-csw07.pdf
 - www.sixxs.net/faq/connectivity/?faq=filters
 - RH0 deprecated by RFC 5095

Other Security Considerations

- Use of /64's for backbone point-to-point links
 - Convention has been to use /64
 - Conforms with autoconfig
 - Concern that hardware-based lookups were optimized for prefix-lengths upto /64
 - One worry is over high CPU utilization if a /64 used on broadcast medium and is scanned, resulting in router performing neighbor-discovery

Security Thoughts

- Know your services
 - Scan all hosts and routers for IPv6 services
 - Nmap supports IPv6 – does NOT support subnet sweeps for IPv6 (approx. 28 years+ for 1 subnet)
- Don't allow mission critical areas to bring up IPv6 without audit/scan of devices by security group
 - Human resources department
 - Credit card department
 - HIPAA, FERPA, etc.

IPv6 Flow



Engineering Workshops

IPv6 Flow Options

- Netflow v9 (aka cflow/jflow)
- Sflow
- IPFix

Common Netflow versions

- Netflow v5 - Fixed record format, no support for IPv6
 - Supported by Cisco, Juniper, Alcatel
- Netflow v9 - Variable record format/template, supports IPv6
 - Supported by Cisco and Juniper (IPv6 traffic reporting since JUNOS 9.4)

Cisco IPv6 Netflow v9 Configuration

- General Configuration

```
ipv6 flow-export version 9
ipv6 flow-export destination <ip-address> <port-no>
ipv6 flow-export template refresh-rate <rate-value>
ipv6 flow-export template timeout <timeout-value>
```

Cisco IPv6 Netflow

- Interface specific commands

```
ipv6 flow ingress
```

```
ipv6 flow egress
```



Cisco CLI Management Commands

- `show ip cache flow`
- `clear ip flow stats`



Juniper IPv6 Netflow v9 Configuration

- General Configuration

```
[edit services flow-monitoring version9]
set template <template-name> flow-active-timeout 20
set template <template-name> flow-inactive-timeout 120
set template <template-name> ipv6-template
```

```
[edit forwarding-options sampling output]
set cflowd <ip-address> port <port-no>
set cflowd <ip-address> source-address <src-ip-address>
set cflowd <ip-address> version9 template <template-name>
```

Juniper IPv6 Netflow

- Interface specific commands

```
[edit interfaces ge-0/0/0 unit 0]  
set family inet6 sampling input  
set family inet6 sampling output
```

IPFix

- IETF working group effort
- Improves on Cisco's Netflow v9
- See: <http://www.nanog.org/meetings/nanog41/presentations/nanog41-ipfix.pdf>



Sflow

- Includes packet header information
- Used by Extreme, Force10, Foundry



Things to Watch For

- Simultaneous IPv6 and flow support
- Impact of IPv6 flow on router or switch performance
- Sampling limitations
- Corner case behavior:
 - MPLS
 - Multicast

Netflow Lab

- Configure an interface on “D” or “E” router to report IPv6 Netflow v9 traffic to one of the pod laptops (or an attendee laptop)
- Open wireshark/tcpdump
- Send IPv6 traffic across interface that has IPv6 Netflow v9 enabled
- Confirm that Netflow v9 traffic is received on laptop -- examine Netflow v9 packets.



IPv6 Applications



Engineering Workshops

Operating Systems - Windows

- Windows XP – Supported since initial release
 - Type “ipv6 install” on XP (no service pack)
 - Type “**netsh interface ipv6 install**” for SP1 or SP2 or use control panel to add network protocol
- Advanced networking service pack adds support for Teredo
- Internet Explorer and Firefox web browsers IPv6-enabled
- 6to4, ISATAP and Teredo supported
- www.microsoft.com/ipv6/

Operating Systems - Windows

- IPv6 is on by default in Windows Vista and Windows “7”, and will be supported across all Microsoft products eventually
 - Active DNS supports AAAA but not transport
- Firewall in Windows 2003 server with SP1 supports IPv6
- Firewall in Windows XP with SP2 supports IPv6
- XP cannot do DNS queries via IPv6 transport
- Ping, tracert, telnet, ftp, netstat and netsh commands all support IPv6
- In Windows Vista, some P2P and/or collaboration tools are IPv6-only
 - e.g. Windows Meeting Space; see <http://technet.microsoft.com/en-us/windowsvista/aa905083.aspx>
 - If the two hosts communicating with these tools don't have native IPv6 connectivity, the IPv6 traffic will be encapsulated in tunnels

Operating Systems – MacOS X

- IPv6 is enabled by default on all interfaces, and can be manually configured through the “network preferences” panel
- 6to4 can be configured, and will track IPv4 address changes
- The “security” panel configures both v4 and v6 firewalls (ipfw and ip6fw)
- No DHCPv6 support yet; talking about supporting RFC 5006 (IPv6 Router Advertisement Option for DNS)

Operating Systems – MacOS X

- IPv6 support has been added for:
 - AppleShare
 - ssh and sshd
 - ftp and ftpd
 - Safari (uses v6 for sites without v4 addresses)
 - DNS queries
 - multicast DNS
 - many other system utilities (telnet, ping, traceroute, syslog, xinetd, etc.)
 - Firefox (pre 3.0) in MacOS X disabled IPv6 DNS resolution by default

Operating Systems - Linux

- www.linux-ipv6.org – USAGI Project (WIDE)
- www.tldp.org/HOWTO/Linux+IPv6-HOWTO/
- www.deepspace6.net – "the Linux IPv6 Portal"
- Most major open source applications support IPv6
 - Red Hat / Fedora enable IPv6 by default but do NOT install ip6tables by default!
- Debian IPv6 Developer's List: <http://lists.debian.org/debian-ipv6/>

Operating Systems - UNIX

- www.kame.net – WIDE's FreeBSD IPv6 site
- www.sun.com/software/solaris/ipv6/ – IPv6 is standard in Solaris since version 8

IPv6-ready hardware and software

- www.ipv6ready.org
 - Focuses mostly on routers, network equipment and operating systems at present
 - Includes participation by WIDE, IPv6 Forum, University of New Hampshire Interoperability Lab
- www.ipv6-to-standard.org
- Presentations by Ron Broersma of DREN
 - <http://events.internet2.edu/speakers/speakers.php?go=people&id=1141>
 - <http://events.internet2.edu/2009/jt-indy/agenda.cfm?go=session&id=10000667&event=1037>

DVTS

- DVTS – Digital Video Transport System

www.sfc.wide.ad.jp/DVTS/

www.dvts.jp

A product of the WIDE Project, DVTS is openly available software which encapsulates DV video in IPv4 or IPv6 packets.

- Supports IPv4 and IPv6, unicast and multicast
- Good for “smoke testing” networks

Apache v.2

- IPv6 support built-in (no patches or other modifications needed)

```
Listen ::
```



Resources

- <http://www.getipv6.info>
- <http://www.ipv6book.ca>
- <http://www.ipv6book.ca/allocation.html>
- <http://ipv6gate.sixxs.net>
- <http://www.sixxs.net>
- <http://www.ipv6forum.com>
- <http://www.ipv6tf.org>
- <http://go6.net>
- <http://www.hexago.com>
- <http://lists.cluonet.de/mailman/listinfo/ipv6-ops>



Contacts

Internet2 IPv6 Working Group
<http://ipv6.internet2.edu/>

Internet2 Network NOC
noc@net.internet2.edu



Engineering Workshops

Please fill out the workshop survey online.

Thank you!



Engineering Workshops