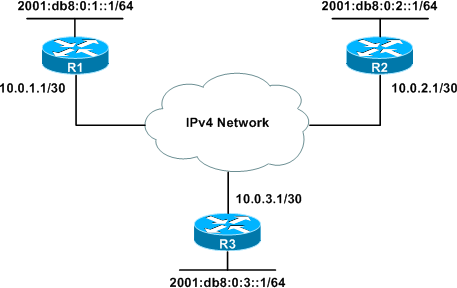
**[6to4 IPv6 Tunneling](http://packetlife.net/blog/2010/mar/15/6to4-ipv6-tunneling/)**

By [stretch](http://packetlife.net/users/stretch/) | Monday, March 15, 2010 at 12:29 a.m. UTC

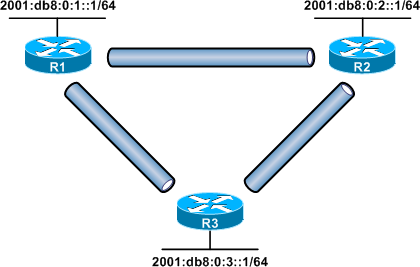
"6to4 tunneling" is an IPv6 transition mechanism described in [RFC 3056](http://tools.ietf.org/html/rfc3056). Like many other transition mechanisms, it enables encapsulation of IPv6 packets into IPv4 for transport across an IPv4 network. What's really neat about 6to4 is that it allows for automatic IPv6-to-IPv4 address translation, and treats the underlying IPv4 network as one big non-broadcast multiaccess (NBMA) network, rather than a collection of independent point-to-point links.

Consider the following topology:



Three sites, each with its own IPv6 LAN, are connected to one another via an IPv4-only backbone.

One way to connect the IPv6 LANs would be to configure individual point-to-point IPv6-in-IPv4 tunnels, so that each site would have an individual tunnel interface to reach every other site. This is a fair solution for our scenario with only three sites, however it obviously does not scale well, requiring n(n-1)/2 tunnels for n sites (e.g. ten sites would require 45 tunnels for full mesh connectivity).



A more efficient solution would be to use automatic 6to4 tunneling. 6to4 works by taking advantage of a reserved IPv6 prefix, **2002::/16**. Aceste adrese sunt rezervate pentru tunelare 6to4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2002:0000::/16** | **6to4** |  |  |  | ALLOCATED | **2002::/16 is reserved for 6to4 [**[**RFC3056**](https://www.iana.org/go/rfc3056)**]. For complete registration details, see [**[**IANA registry *iana-ipv6-special-registry***](https://www.iana.org/assignments/iana-ipv6-special-registry)]. |

A 6to4 tunnel interface automatically converts the 32 bits in its IPv6 address following this prefix to a global unicast IPv4 address for transport across an IPv4 network such as the public Internet. Before digging any deeper into the theory behind 6to4 tunneling, let's look at the configuration involved.

**6to4 Configuration**

To configure 6to4 tunneling, we first need to create a tunnel interface on each dual-stack edge router. There are three key components relevant to 6to4:

* The tunnel mode (**6to4**)
* The tunnel source (IPv4 interface or address)
* The 6to4 IPv6 address (**within 2002::/16**)

On R1, we create the tunnel interface, configure it as 6to4, and specify its cloud-facing IPv4 interface as the tunnel source:

R1(config)# **interface tunnel0**

R1(config-if)# **tunnel mode ipv6ip 6to4**

R1(config-if)# **tunnel source 10.0.1.1**

To the determine the IPv6 address of the tunnel interface, we convert the IPv4 source address to hexadecimal in IPv6 notation, so that 10.0.1.1 becomes 0a00:0101. then we append it to the 6to4 prefix (2002::/16), and fill the remainder of the address with zeros. Host masks (/128) are used for 6to4 addresses in this article, although other prefix lengths (and addresses within the /48) can be used as well.

R1(config-if)# **ipv6 address 2002:a00:101::/128**

This process is repeated to create 6to4 tunnel interfaces for the two other sites as well. Only a single tunnel interface is needed per router.

R2(config)# **interface tunnel0**

R2(config-if)# **tunnel mode ipv6ip 6to4**

R2(config-if)# **tunnel source 10.0.2.1**

R2(config-if)# **ipv6 address 2002:a00:201::/128**

R3(config)# **interface tunnel0**

R3(config-if)# **tunnel mode ipv6ip 6to4**

R3(config-if)# **tunnel source 10.0.3.1**

R3(config-if)# **ipv6 address 2002:a00:301::/128**

Note that, unlike when configuring point-to-point tunnels, we have not designated any tunnel destination addresses.

Now that the 6to4 address of each router is known, we can go back and add the necessary static routes to achieve IPv6 connectivity between all three routers. Three static routes must be added on each router. The first indicates that 2002::/16 is reachable out of interface Tunnel0. The other two are routes for the individual /64 prefixes at either of the other sites, which are to be routed via the 6to4 tunnel.

**R1**

ipv6 route 2002::/16 tunnel0

ipv6 route 2001:db8:0:2::/64 2002:a00:201::

ipv6 route 2001:db8:0:3::/64 2002:a00:301::

**R2**

ipv6 route 2002::/16 tunnel0

ipv6 route 2001:db8:0:1::/64 2002:a00:101::

ipv6 route 2001:db8:0:3::/64 2002:a00:301::

**R3**

ipv6 route 2002::/16 tunnel0

ipv6 route 2001:db8:0:1::/64 2002:a00:101::

ipv6 route 2001:db8:0:2::/64 2002:a00:201::

At this point we should have IPv6 connectivity among the IPv6 LANs at all three sites:

R1# **ping 2001:db8:0:2::1**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2001:DB8:0:2::1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/31/56 ms

R1# **ping 2001:db8:0:3::1**

Type escape sequence to abort.

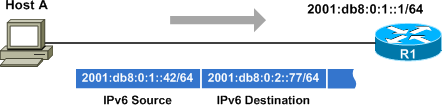
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:3::1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/12/16 ms

**How 6to4 Works**

The logic behind 6to4 tunneling may not be readily apparent. Let's trace the path of an IPv6 packet from a host behind R1 to a host behind R2. The packet begins as a normal IPv6 packet with IPv6 source and destination addresses. These addresses *do not change* for the life of the packet.



R1 receives the IPv6 packet and compares the destination IPv6 address against its IPv6 routing table.

R1# **show ipv6 route**

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route, M - MIPv6

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

C 2001:DB8:0:1::/64 [0/0]

via ::, Loopback0

L 2001:DB8:0:1::1/128 [0/0]

via ::, Loopback0

S 2001:DB8:0:2::/64 [1/0]

via 2002:A00:201::

S 2001:DB8:0:3::/64 [1/0]

via 2002:A00:301::

S 2002::/16 [1/0]

via ::, Tunnel0

LC 2002:A00:101::/128 [0/0]

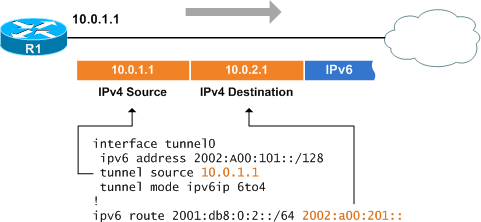
via ::, Tunnel0

L FF00::/8 [0/0]

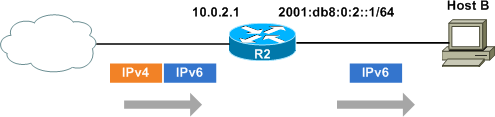
via ::, Null0

The closest match is our static route for 2001:db8:0:2::/64, with a next-hop of 2002:a00:201::. This forces a *recursive lookup* for a route for 2002:a00:201::. The closest (and only) matching route is out interface Tunnel0; note that no next-hop IP address is specified for this second route.

This is where 6to4 comes into play. As the IPv6 packet is routed out of Tunnel0, it is encapsulated into IPv4. The source IPv4 address for the new header is taken from the tunnel source address. The destination IPv4 address is calculated from the next-hop address of our route for 2001:db8:0:2::/64; 2002:a00:201:: is converted to 10.0.2.1 using the reverse of the process we used earlier to decide the tunnel interfaces' IPv6 addresses.



The encapsulated packet is routed through the IPv4 cloud until it reaches R2. The IPv4 header is stripped upon entering R2's 10.0.2.1 interface. The remaining IPv6 packet is routed onto the IPv6 LAN toward Host B.



Our example relies on static routes. Note that internal routing protocols such as EIGRP and OSPFv3 cannot be used across 6to4 tunnels because link-local IPv6 addresses (needed to form router adjacencies) are not supported across 6to4 tunnels. BGP, which supports adjacencies based on global IPv6 addresses, can be used instead.

Also note that despite its NBMA nature, 6to4 tunnels are still *tunnels*, requiring both an ingress and egress IPv6 interface; they cannot be used to communicate directly between IPv4 and IPv6 hosts.