IPv6 Migration Document

Introduction:

This document will discuss about the need and possible smooth ways of migration from IPv4 to IPv6 based network.

IPv4 Exhaustion and need for IPv6 :

Back in 1970s when IPv4 was developed, Internet and network were only used for research and it was not predicted that Internet will evolve as it is today. In early 1990s, experts predicted IPv4 exhaustion based on Internet's drastic growth and there raises a need for a new network layer protocol. Stream Protocol 2 (ST2) also designated as IPv5 was developed around 1993 which was used by IBM, NeXT, Sun Microsystems in their network. In the meantime, technologies like NAT, dynamic address allocation helped keeping IPv4 address exhaustion under control which provides developers enough time for IPng development. IPng was officially designated as IPv6.

In Feb 2011, the last 5 blocks of IPv4 address have been allotted to RIRs and now it is nearly exhausted that businesses need to accelerate the action of migrating Internet from IPv4 to IPv6.

IPv6 Migration Challenges:

Considering the fact being that the whole Internet is currently running on IPv4 network, it is not an easy task to migrate network from IPv4 to IPv6. Below are few challenges currently we face as part of migration,

- While Service provider's current customer base is running on IPv4 network, any new customer may be based on IPv6. This requires that SP core should handle both IPv4 and IPv6 customers without compromising with performance.
- While any new business applications will be developed based on IPv6, current applications are IPv4 based. This requires that at least for few years, IPv4 and IPv6 network should co-exist and inter communicate.

The above mentioned challenges boil down to the below requirement:

- Communication between IPv6 only networks over IPv4 only cloud.
- Communication between IPv4 only networks over IPv6 only cloud.
- Communication between IPv6 only networks and IPv4 only network.

Below is some migration technique which can be used,

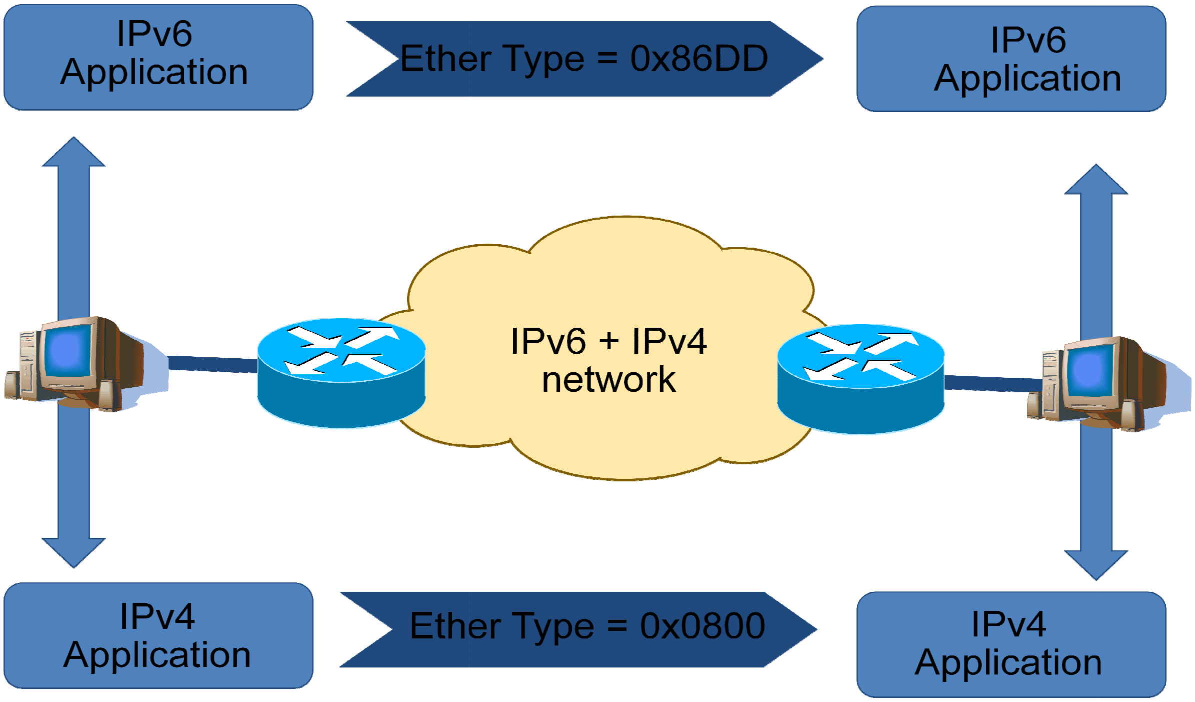
1. Dual Stack technique
2. Tunneling technique
3. Protocol translation technique

1. Dual Stack Technique:

Dual stack is one possible migration technique that involves running both IPv4 and IPv6 at the same time end to end. This involves enabling all application to be aware of both IPv4 and IPv6 protocol stack and end-to-end network running both IPv4 and IPv6 protocol stack.

Any communication data from IPv6 application on end host will be interpreted by IPv6 protocol stack in network layer and will send out with Ether type as 0x86DD. Edge router on receiving it will understand it as IPv6 based on Ether type and will be treated accordingly with IPv6 protocol stack. The same continues end to end and both IPv4 and IPv6 cloud will communicate among themselves. No intercommunication between IPv6 and IPv4 is possible with dual stack technique.

Figure 1.1

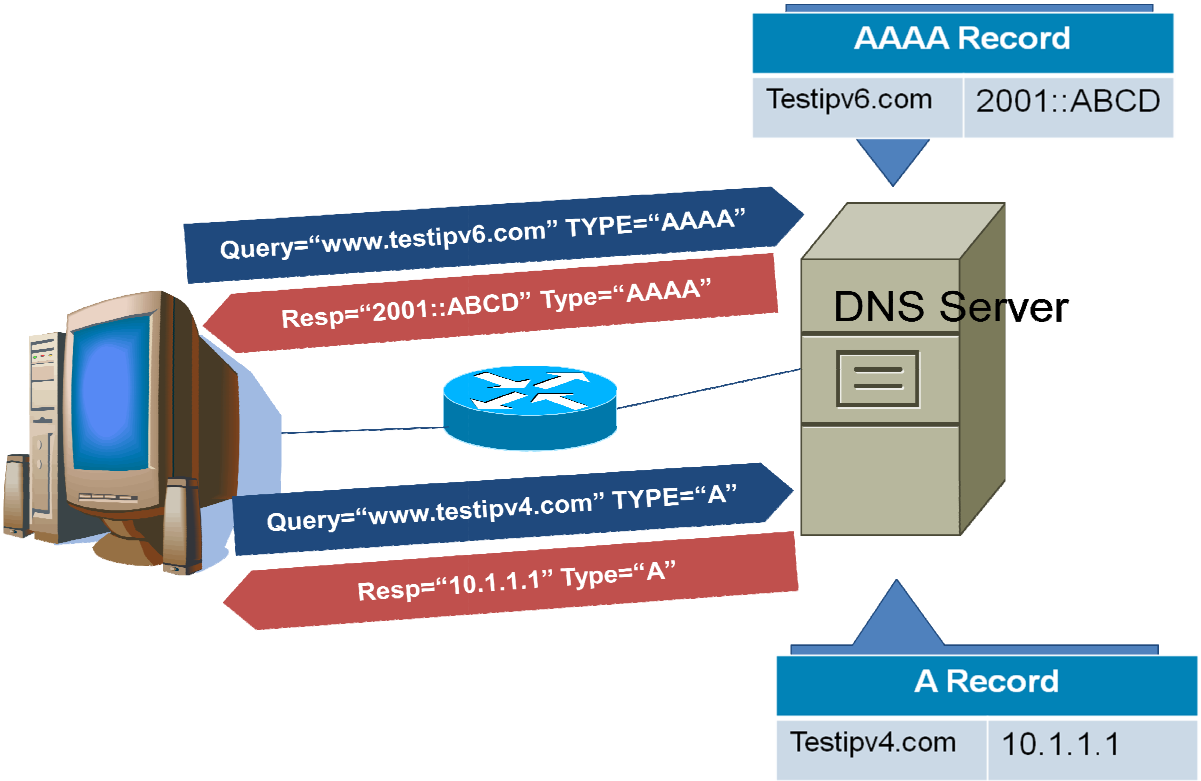


1.2. Dual stack approach with DNS

How a client application does decide if it should use IPv4 or IPv6 protocol stack to communicate to server?

It simply can be hardcoded within the application or can rely on DNS reply. For example, when a client want to establish a connection to server named “test.com”, it will send a DNS query to DNS server to resolve “test.com”. If the test.com server is IPv4 based, a A record will be replied back to querier with some IPv4 address. Client on receiving it will use IPv4 protocol stack for further activities. If test.com server is IPv6 based, AAAA record or a quad-A record will be replied back to querier with IPv6 address.

Figure 1.2


AAAA Record
Testipv6.com
2001::ABCD
Query="www.testipv6.com" TYPE="AAAA"
Resp="2001::ABCD" Type="AAAA"
DNS Server
Query="www.testipv4.com" TYPE="A"
Resp="10.1.1.1" Type="A"
A Record
Testipv4.com
10.1.1.1

2. Tunneling technique:

Tunneling basically provides a way to use an existing network layer infrastructure to carry a different network layer protocol. For example, IPX over IPv4 network, IPv6 over IPv4 network.

IPv6 tunneling is the way of transporting IPv6 packets over IPv4 infrastructure by encapsulating IPv6 packet with IPv4 header. Tunnel establishment can either be manual or automatic. Semi automated tunnels can also be established using Tunnel broker approach. As part of migration, situation may also arise where IPv4 packets may need to be encapsulated over IPv6 infrastructure.

On a broader view tunneling technique can be classified as,

- IPv6 tunnel over IPv4 infrastructure
- IPv4 tunnel over IPv6 infrastructure

2.1. IPv6 tunnel over IPv4 infrastructure

Below are few of the tunneling technique to connect IPv6 network over IPv4 cloud,

- Manual 6to4 Tunnel
- IPv6 over GRE Tunnel
- Automatic 6to4 Tunnel
- ISATAP
- 6RD
- IPv6 over MPLS solution

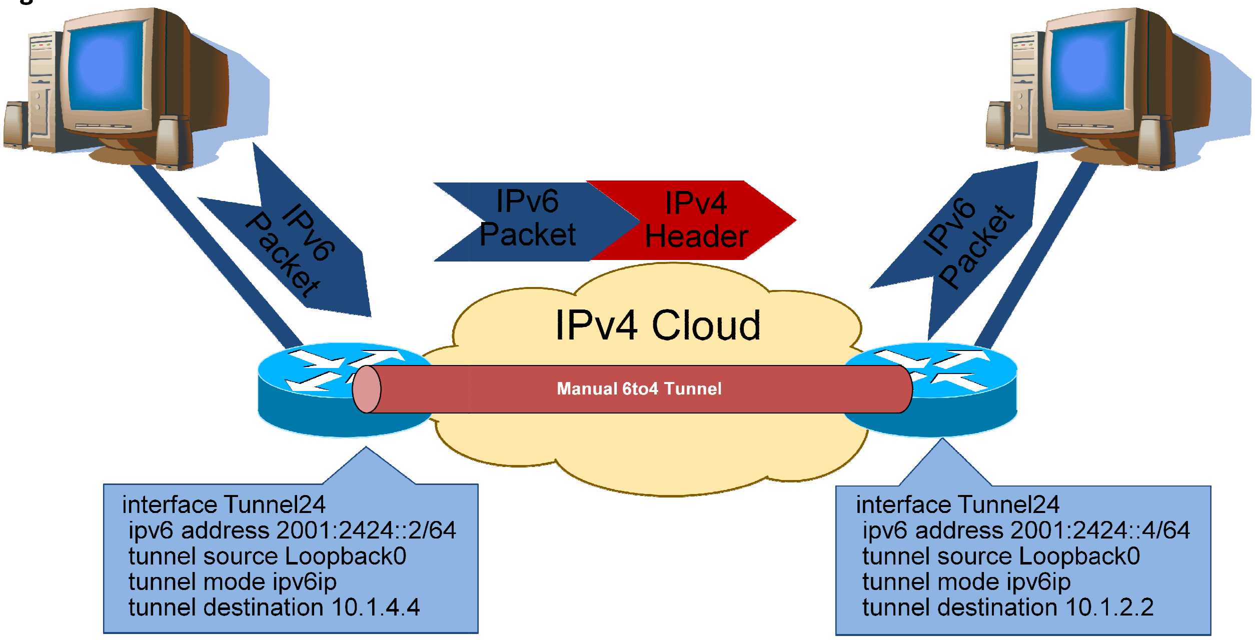
2.1.1. Manual 6to4 Tunneling technique:

This is a simple point-to-point tunnel that needs to be statically configured between two end points which are dual stacked by running IPv4 facing the core and IPv6 facing the CE side. IPv6 packet will be encapsulated with IPv4 header on ingress side with IP protocol number = 41 and no additional header.

There is no special requirement for the IPv6 address configured on tunnel interface except that the both should be on same subnet and the end points should be reachable via IPv4.

This technique which is one of the initially used mechanisms is least preferred these days due to the scalability issue.

Figure 2.1



```
interface Tunnel24
 ipv6 address 2001:2424::2/64
 tunnel source Loopback0
 tunnel mode ipv6ip
 tunnel destination 10.1.4.4
```

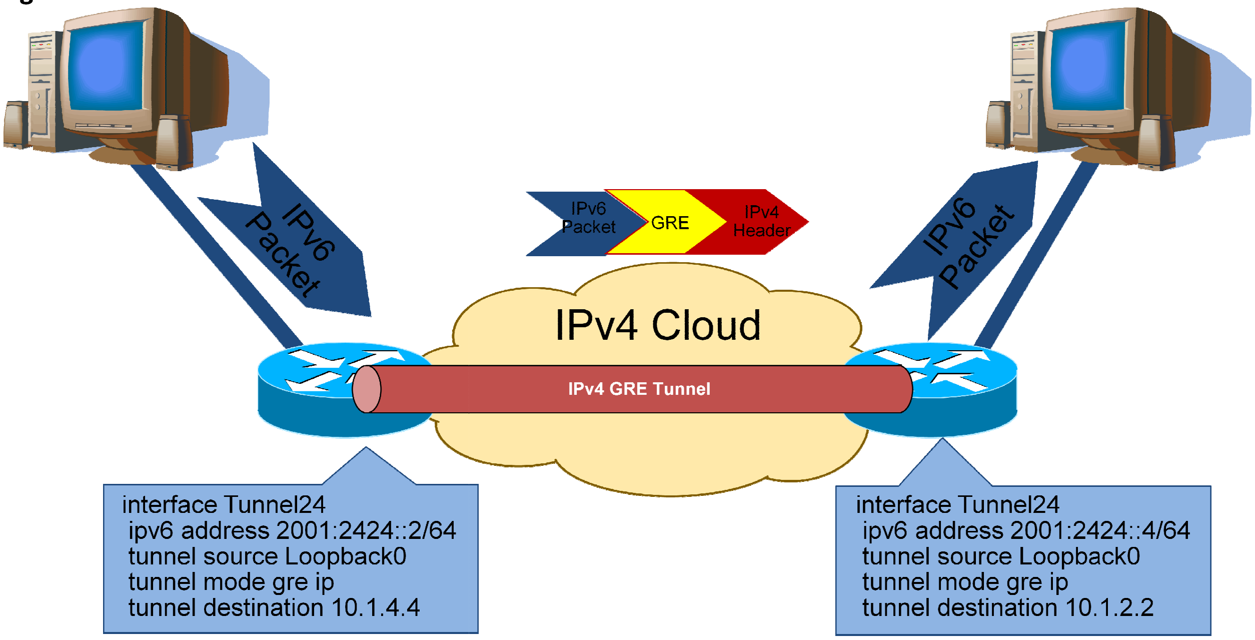
```
interface Tunnel24
 ipv6 address 2001:2424::4/64
 tunnel source Loopback0
 tunnel mode ipv6ip
 tunnel destination 10.1.2.2
```

2.1.2. IPv6 over GRE Tunneling technique:

GRE, Generic Routing Encapsulation protocol is one tunneling protocol that encapsulates one network layer protocol into other. As like Manual 6to4 tunnel, this is a point-to-point static tunnel that needs to be configured between two end points which are dual stacked with IPv4 as core facing and IPv6 as CE facing. In this case, IPv6 is the passenger protocol which traverses over IPv4 as carrier protocol with GRE header added.

IPv6 packets will be encapsulated with GRE header with Protocol type marked as 0x86DD. This in turn will be encapsulated with IPv4 headers with Protocol number as 47 (GRE). As like manual 6to4 technique GRE tunneling doesn’t require any special format IPv6 address except that both side addresses should be from same subnet and end points should be reachable via IPv4.

Figure 2.2



```
interface Tunnel24
 ipv6 address 2001:2424::2/64
 tunnel source Loopback0
 tunnel mode gre ip
 tunnel destination 10.1.4.4
```

```
interface Tunnel24
 ipv6 address 2001:2424::4/64
 tunnel source Loopback0
 tunnel mode gre ip
 tunnel destination 10.1.2.2
```

2.1.3. Automatic 6to4 Tunnel technique:

Manual 6to4 tunnel though had low overhead comparing to GRE tunneling still suffered with scalability issue due to the fact that it is a static point to point tunneling mechanism that requires having tunnel destination statically mentioned. Automatic 6to4 technique tries to fix this limitation by making the tunnel as point to multipoint.

6to4 border router which is dual stacked with IPv4 facing core and IPv6 facing CE treats core facing IPv4 infrastructure as Non broadcast multi-access (NBMA) link. Having said that it treats IPv4 core as NBMA link, we need some automatic way to identify the peer connected to NBMA link. A special format IPv6 address also known as 6to4 address which is IPv4 address embedded IPv6 address with 2002::/16 as prefix is allotted by IANA. This 6to4 address comprises of Format Prefix: 001, TLA Value: 0x0002, NLA Value: IPv4 address to make it as 2002:v4-address::/48

This can be illustrated as below,

Figure 2.3

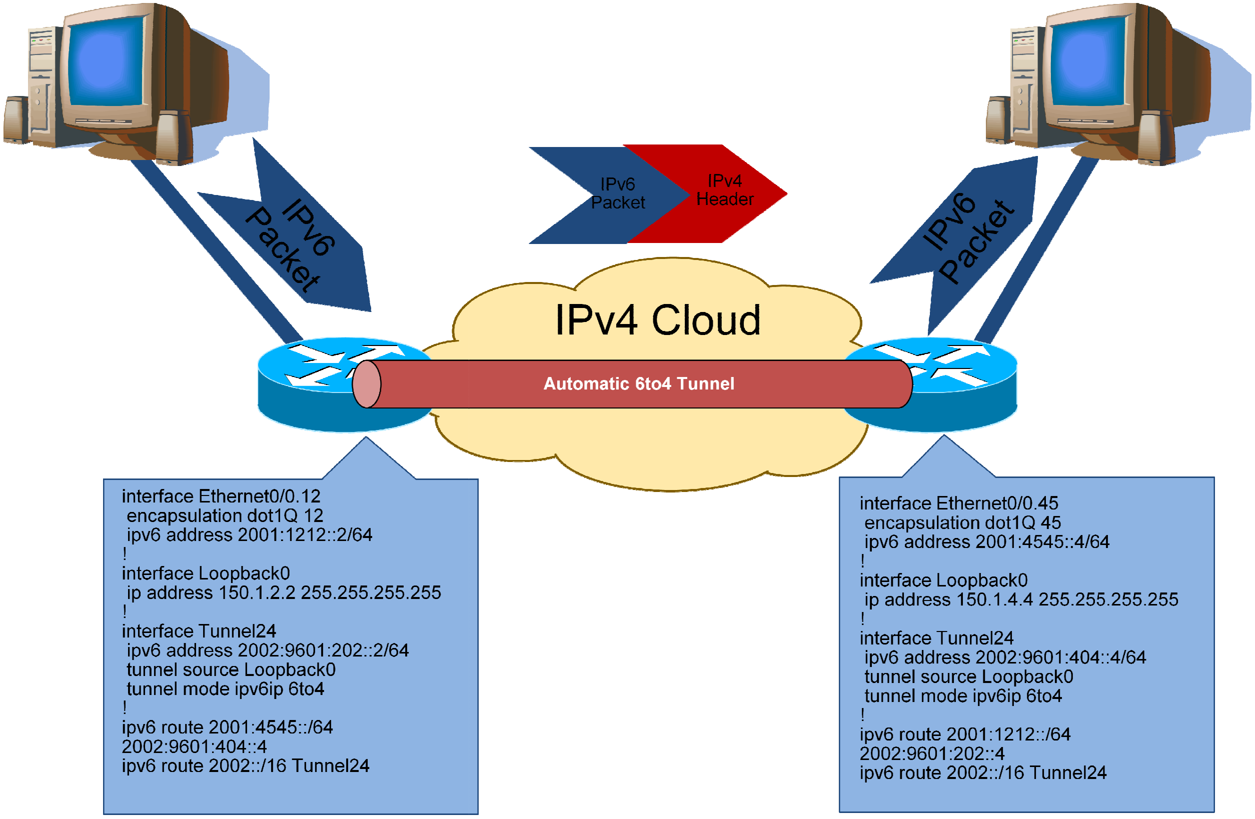
FP=001	TLA=0x0002	IPv4 Address	SLA + Interface ID
2002:		10.1.1.1:	::

6to4 border router can identify other border routers by extracting the IPv4 address from this special IPv6 address (this address needs to be assigned to tunnel interface of any border router).

6to4 border router will look into its IPv6 routing table to identify the next-hop to reach other IPv6 domain. This next hop address will be 6to4 address from which other border routers IPv4 address will be extracted. Now IPv6 packet will be encapsulated with IPv4 header with Protocol number = 41 (which is same as Manual 6to4 tunnel) with destination as IPv4 address extracted from 6to4 address.

Since the tunnel is point to multipoint in nature, IGP is not possible over the tunnel and so have to rely on either static routing or BGP to reach IPv6 sites in other domains.

Figure 2.4



```
interface Ethernet0/0.12
 encapsulation dot1Q 12
 ipv6 address 2001:1212::2/64
!
interface Loopback0
 ip address 150.1.2.2 255.255.255.255
!
interface Tunnel24
 ipv6 address 2002:9601:202::2/64
 tunnel source Loopback0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2001:4545::/64
2002:9601:404::4
ipv6 route 2002::/16 Tunnel24
```

```
interface Ethernet0/0.45
 encapsulation dot1Q 45
 ipv6 address 2001:4545::4/64
!
interface Loopback0
 ip address 150.1.4.4 255.255.255.255
!
interface Tunnel24
 ipv6 address 2002:9601:404::4/64
 tunnel source Loopback0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2001:1212::/64
2002:9601:202::4
ipv6 route 2002::/16 Tunnel24
```

2.1.4. ISATAP Tunneling technique

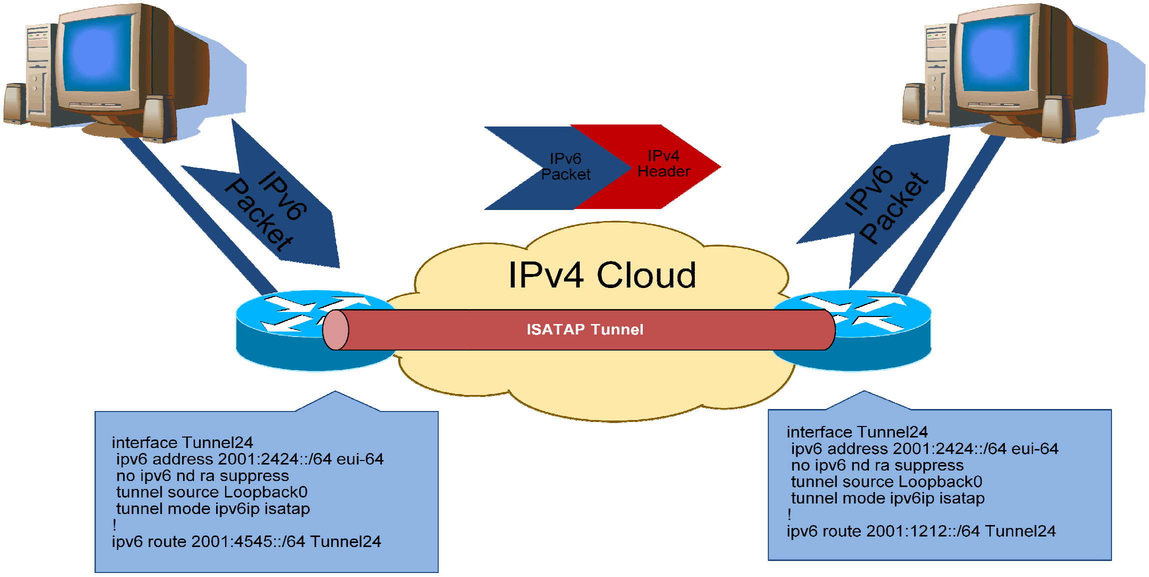
ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) is another mechanism to transport IPv6 traffic over IPv4 infrastructure. As like Automatic 6to4 technique, ISATAP treats core facing IPv4 infrastructure as NBMA link and so have a special format IPv6 address as like Automatic 6to4 technique.

ISATAP router uses special format IPv6 address which comprises of first 64 bits as any Global or site local prefix followed by special ISATAP ID of 32 bits as 0000:5EFE and then by IPv4 address (converted to hex value).

ISATAP tunnel can be either between host and router or between routers to connect IPv6 islands. In case of ISATAP tunnel between host and router, ISATAP host should be populated with PRL (Preferred Router List) either statically or by DNS. ISATAP host will use this list to get the IPv4 address and send ICMPv6 Router Discovery with source and destination as IPv6 link local address encapsulated into IPv4 header and send across to any ISATAP router in the list. ISATAP router in turn will return the ISATAP prefix which will be used for communication further.

In case of ISATAP tunnel between routers, it works as like automatic 6to4, but without the restriction that it should be with 2002::/16 prefix.

Figure 2.5



```
interface Tunnel24
 ipv6 address 2001:2424::/64 eui-64
 no ipv6 nd ra suppress
 tunnel source Loopback0
 tunnel mode ipv6ip isatap
!
ipv6 route 2001:4545::/64 Tunnel24
```

```
interface Tunnel24
 ipv6 address 2001:2424::/64 eui-64
 no ipv6 nd ra suppress
 tunnel source Loopback0
 tunnel mode ipv6ip isatap
!
ipv6 route 2001:1212::/64 Tunnel24
```

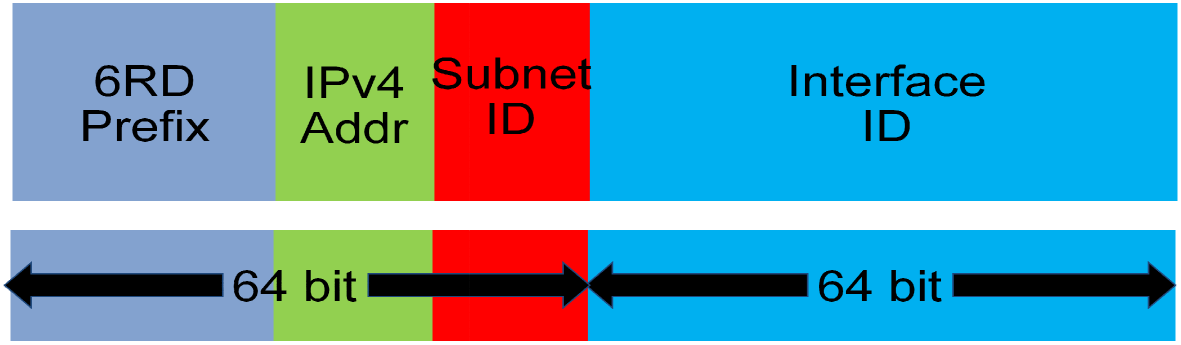
2.1.5. IPv6 Rapid Deployment (6RD) Technique:

6RD (IPv6 Rapid Deployment) is the latest tunneling technique used to transport IPv6 traffic over IPv4 SP network. This is an extension of automatic 6to4 tunneling mechanism with key differentiator that it utilizes SP’s own prefix instead of reserved prefix like 2002::/16. 6RD domain consists of 2 components as 6RD CE and 6RD BR (Border Router).

Both 6RD CE and 6RD BR must be dual stacked with IPv4 and IPv6 protocols. 6RD BR will be connected to external IPv6 domain while 6RD CE will connect IPv6 domains within same autonomous control.

Below is the 6RD address format,

Figure 2.6



6RD prefix is a variable length prefix that can be of Service Providers choice from the prefix they received from registry. This is a key differentiator from 6to4 tunnel where we need to use 2002::/16 as the prefix. Since the prefix belongs to SP, it will be always unique.

Embedded IPv4 address is a variable length field where normally the IPv4 address of the remote side CE or BR will be embedded. 6RD supports prefix compression that if the entire service provider network address commonly starts with 10.x.y.z, we can omit 10 and only use 24 bit in Embedded IPv4 field to fit x.y.z.

Subnet ID is another variable length field. ALL 6RD CE and 6RD BR will be given with same 6RD prefix to which it will append/embed its own IPv4 address (after compression if required). Now Subnet ID will be used to differentiate between subnets.

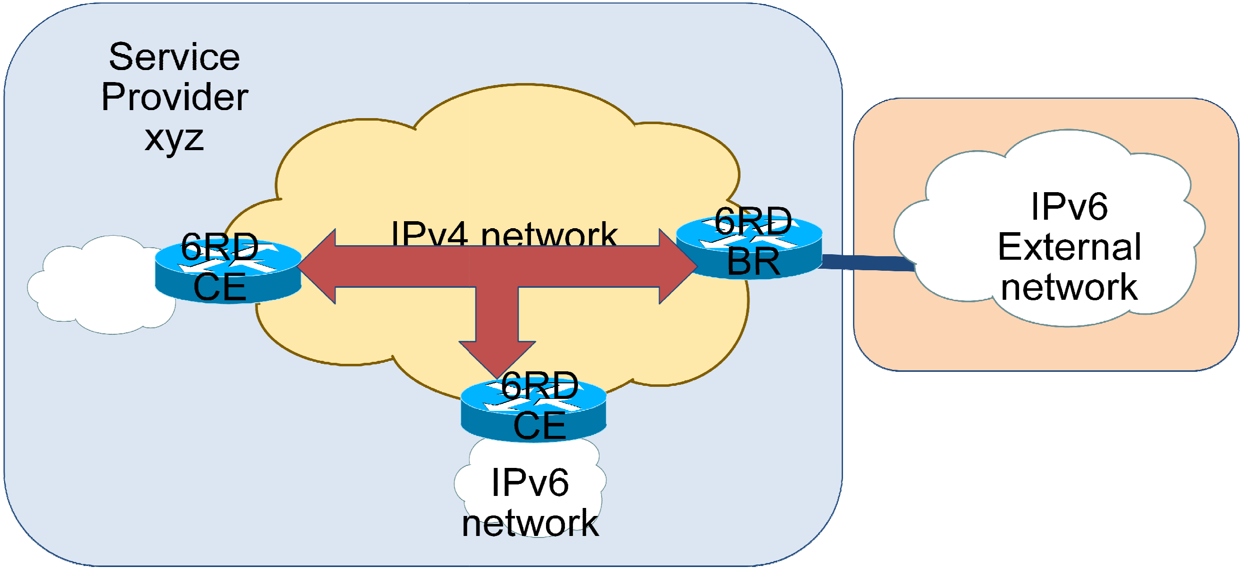
Interface ID is 64 bit field which will be derived from host MAC address.

A 24 bit 6RD prefix as ABCD:AB::/24 with 24 bit of compressed IPv4 address (assuming it from 10.x.y.z network) allows 16 bit for subnet ID and 64 bit for interface ID.

For example, 6RD address on a router with 10.1.1.1 as IPv4 address (entire network is in 10.x.y.z format) in Site ID 100 of 6RD domain allotted with ABCD:AB::/24 will be as below,

ABCD:AB01:0101:0064:<INTERFACE_ID>

Figure 2.7



On CE, if the destination IPv6 address matches local 6RD domain prefix, it will use the embedded IPv4 prefix and communicate directly and if it doesn't match, it will be sent to BR to communicate with external IPv6 domain. Since the IPv6 addresses (6RD address) are unique due to the prefix, they don't need any address translation while communicating with external network.

2.1.6. IPv6 over MPLS Solution:

In current trend, almost all Service providers have MPLS implemented in core to provide service to end customers. MPLS technology performs packet forwarding based on labels in header and not the layer 3 details. This existing MPLS is leveraged to provide IPv6 service between end customers over SP without enabling IPv6 in core.

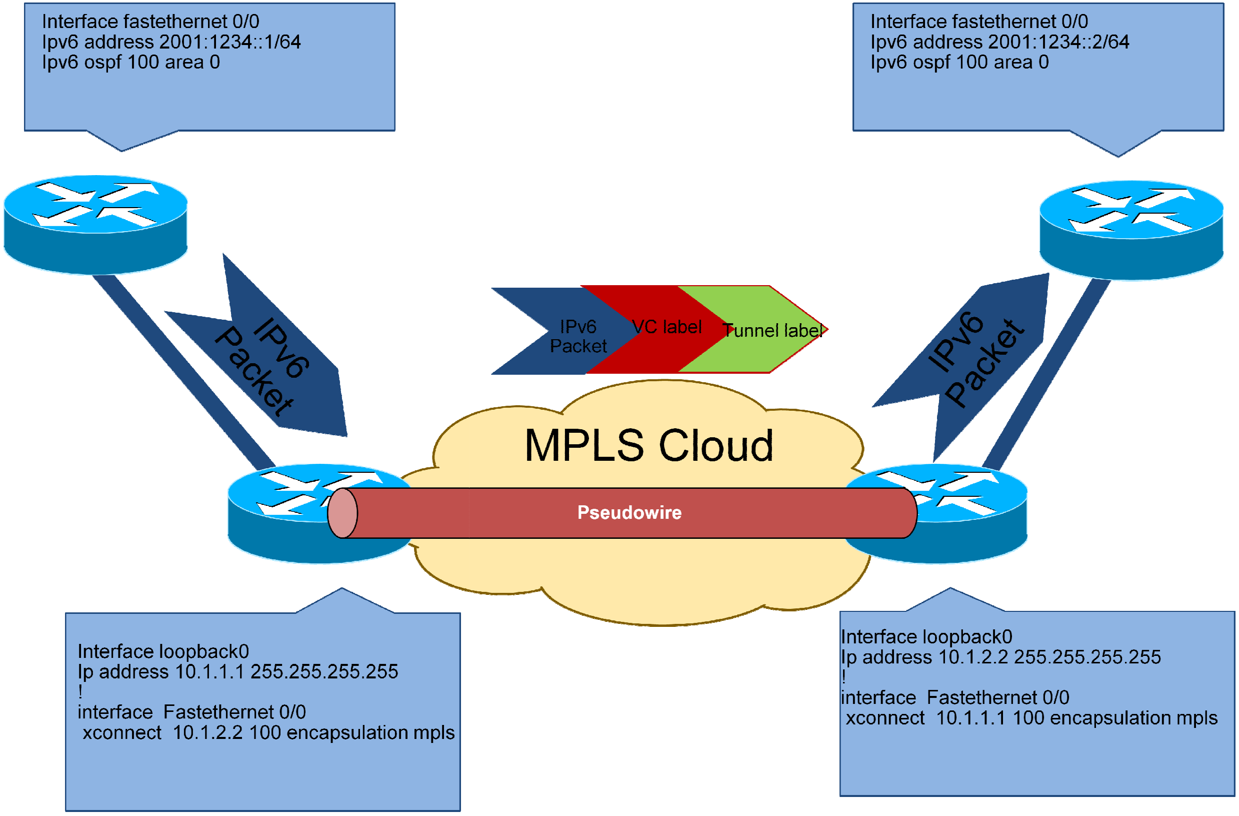
There are 3 different solutions that help providing IPv6 service over MPLS cloud as below,

1. IPv6 over Pseudowire
2. 6PE – IPv6 over MPLS
3. 6VPE – MPLS VPN for IPv6 customer

2.1.6.1. IPv6 over Pseudowire

AToM (Any Transport over MPLS) helps transporting Layer 2 packets over MPLS backbone. Pseudowire will be established between PE devices which act as virtual tunnel over which any Layer 2 packet can be label switched. There will be minimum 2 labels imposed in ingress PE. Top label (Tunnel label) will be used to identify the egress PE and bottom label (VC label) to identify the egress interface in egress PE. No changes required in MPLS backbone to transport IPv6 packets.

Figure 2.8



```
Interface fastethernet 0/0
Ipv6 address 2001:1234::1/64
Ipv6 ospf 100 area 0
```

```
Interface fastethernet 0/0
Ipv6 address 2001:1234::2/64
Ipv6 ospf 100 area 0
```

```
Interface loopback0
Ip address 10.1.1.1 255.255.255.255
!
interface  Fastethernet 0/0
 xconnect  10.1.2.2 100 encapsulation mpls
```

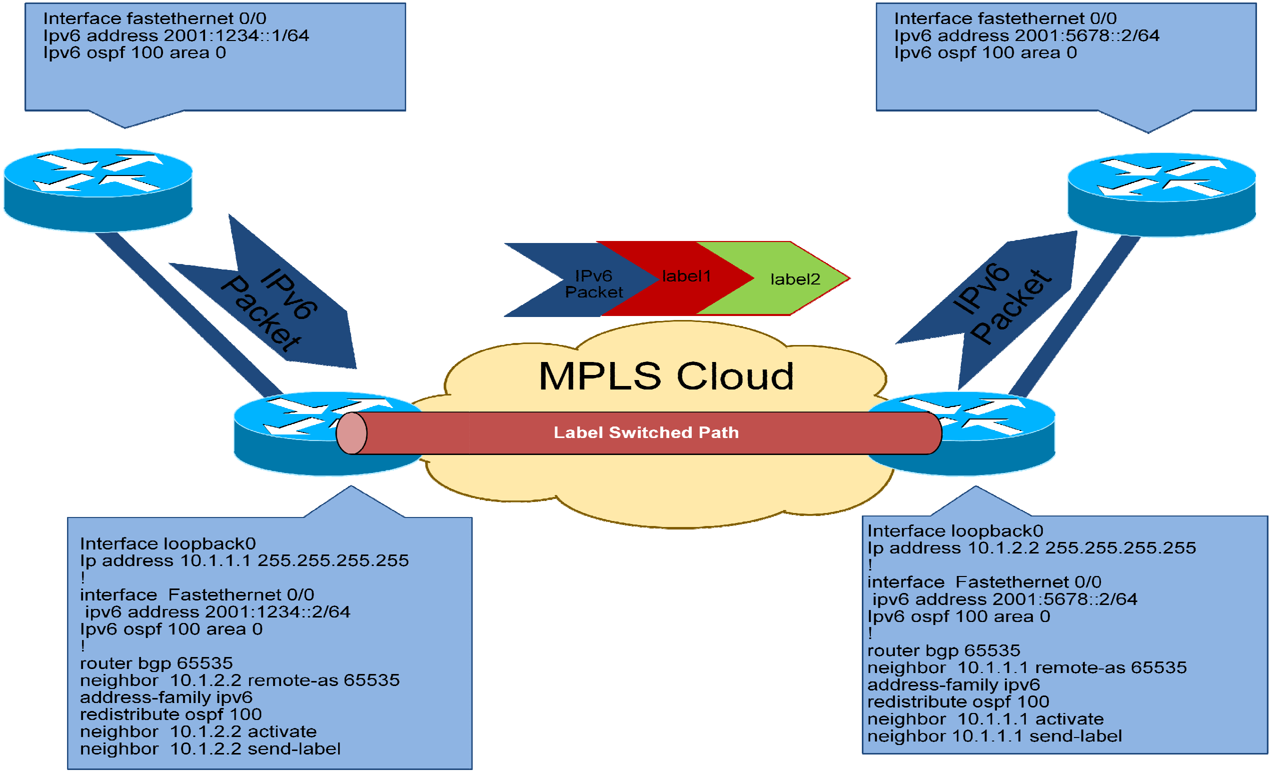
```
Interface loopback0
Ip address 10.1.2.2 255.255.255.255
!
interface  Fastethernet 0/0
 xconnect  10.1.1.1 100 encapsulation mpls
```

2.1.6.2. 6PE solution

6PE is another feature that allows SP to provide IPv6 service without redesigning the existing stable MPLS backbone. PE devices which are connected to both CE and core will be dual stacked with IPv4 as core facing and IPv6 as CE facing. Multi-Protocol BGP will be running between PE devices.

PE device on receiving the IPv6 prefix from CE by routing protocol will assign a label and advertise through BGP to other PE devices which in turn will be advertised to remote CE devices. This label information will be used to switch the packet through MPLS cloud. 6PE implementation will have minimum 2 labels imposed, top label being the one to identify the remote PE while bottom label to identify the IPv6 prefix.

Figure 2.9



```
Interface fastethernet 0/0
Ipv6 address 2001:1234::1/64
Ipv6 ospf 100 area 0
```

```
Interface fastethernet 0/0
Ipv6 address 2001:5678::2/64
Ipv6 ospf 100 area 0
```

```
Interface loopback0
Ip address 10.1.1.1 255.255.255.255
!
interface  Fastethernet 0/0
 ipv6 address 2001:1234::2/64
Ipv6 ospf 100 area 0
!
router bgp 65535
neighbor  10.1.2.2 remote-as 65535
address-family ipv6
redistribute ospf 100
neighbor  10.1.2.2 activate
neighbor  10.1.2.2 send-label
```

```
Interface loopback0
Ip address 10.1.2.2 255.255.255.255
!
interface  Fastethernet 0/0
 ipv6 address 2001:5678::2/64
Ipv6 ospf 100 area 0
!
router bgp 65535
neighbor  10.1.1.1 remote-as 65535
address-family ipv6
redistribute ospf 100
neighbor  10.1.1.1 activate
neighbor 10.1.1.1 send-label
```

2.1.6.3. 6VPE solution

6VPE is similar to 6PE solution where the latter is for IPv6 VPN customers. In other words, 6VPE is MPLS VPN service for IPv6 customers.

All PE devices will be dual stacked with IPv6 as CE facing and IPv4 as core facing with MPLS enabled in core facing interface. Multi protocol BGP will be running between PE devices. PE device on receiving

IPv6 prefix from VRF customer will assign label and will advertise to other PE devices through MP-BGP. These labels will be used to label switch IPv6 traffic over MPLS cloud.

2.2. IPv4 tunnel over IPv6 infrastructure

As part of migration, we may face up with situation where the core/backbone is IPv6 enabled over which we may need to provide IPv4 service.

Below are few of the technique to connect IPv4 over IPv6 cloud,

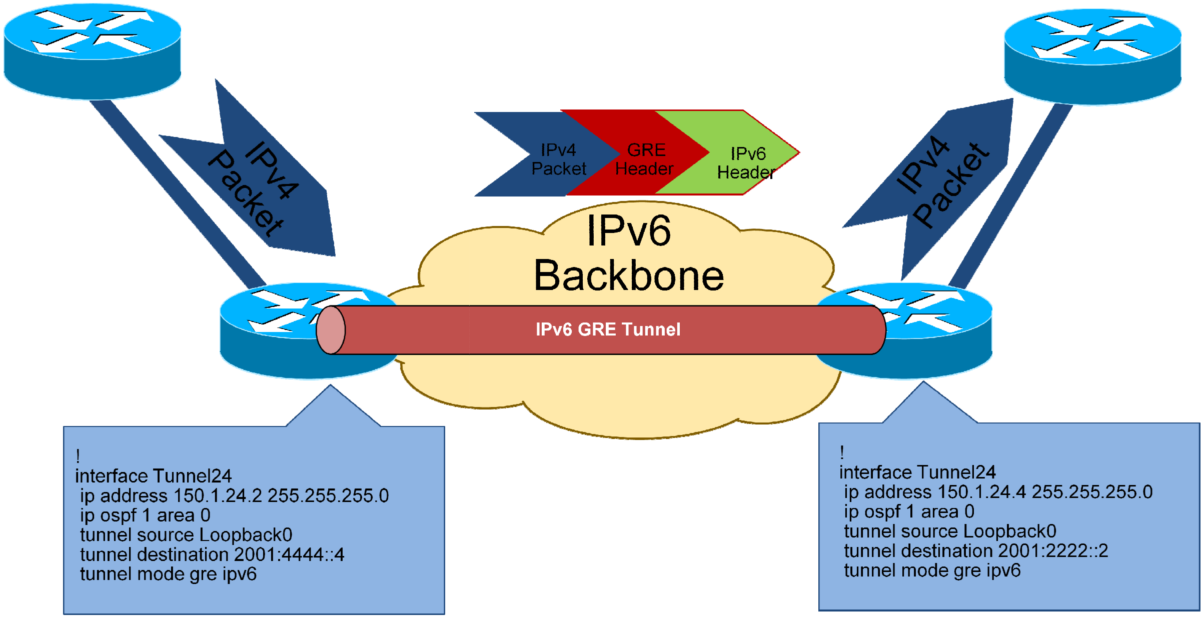
- IPv4 over manual IPv6 GRE tunnel
- Manual 4over6 tunnel
- Automatic 4over6 tunnel
- 4RD (IPv4 Residual Deployment)

2.2.1. IPv4 over IPv6 GRE tunnel technique:

As mentioned earlier, GRE (Generic Routing Encapsulation) is one tunneling technique that encapsulates one protocol over other. In this case, IPv6 is our carrier protocol over which IPv4 traffic will be transported by encapsulating IPv6 GRE header.

PE devices will be dual stacked with IPv6 as core facing protocol and IPv4 as CE facing protocol. PE device on receiving IPv4 packet from CE facing interface will encapsulate with GRE header with Protocol Type as 0x0800 (Ether Type for IPv4). This in turn will be encapsulated with IPv6 header with 47 as protocol number and transported over IPv6 network towards egress PE.

Figure 2.10



```
!
interface Tunnel24
 ip address 150.1.24.2 255.255.255.0
 ip ospf 1 area 0
 tunnel source Loopback0
 tunnel destination 2001:4444::4
 tunnel mode gre ipv6
```

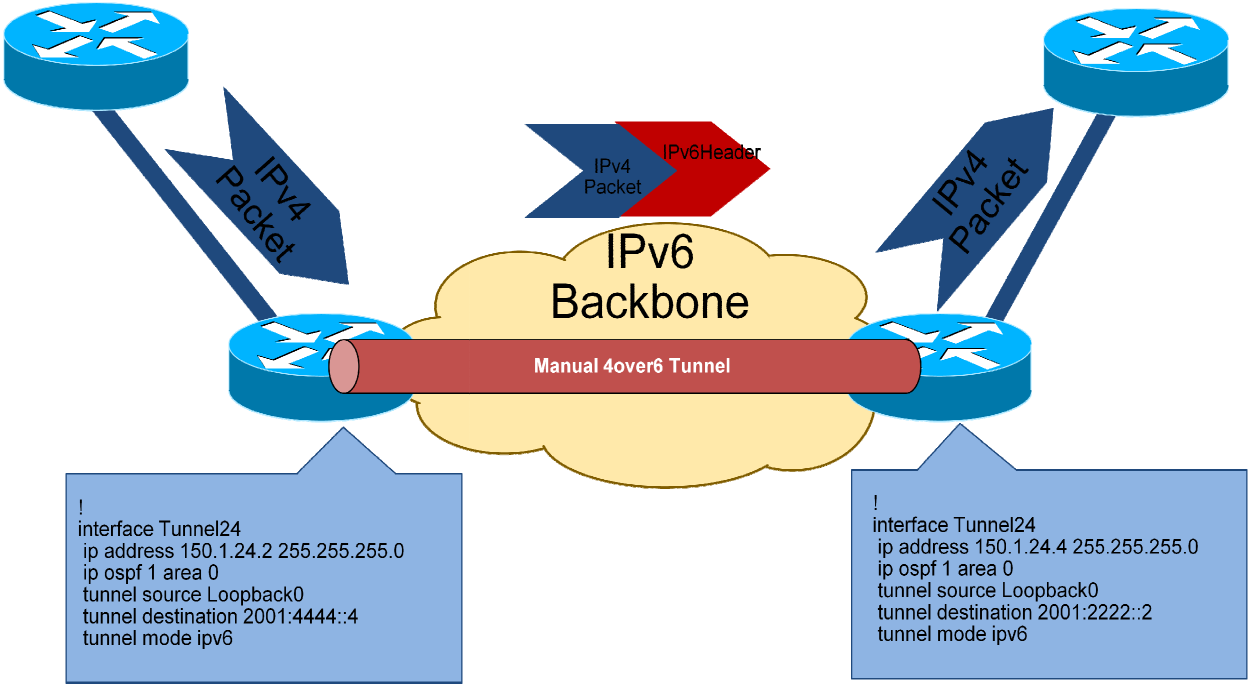
```
!
interface Tunnel24
 ip address 150.1.24.4 255.255.255.0
 ip ospf 1 area 0
 tunnel source Loopback0
 tunnel destination 2001:2222::2
 tunnel mode gre ipv6
```

2.2.2. Manual 4over6 tunnel technique

Manual 4over6 tunnel is another technique which uses static point to point tunnel established between end points which are dual stacked with IPv6 as core facing and IPv4 as CE facing. IP address configured on the tunnel interface on both end points should be in same subnet.

PE device on receiving IPv4 packet from CE facing interface will encapsulate with IPv6 header and mark “Next Header” field as 0x04 (IP in IP encapsulation) and send across over the IPv6 cloud.

Figure 2.11



```
!
interface Tunnel24
 ip address 150.1.24.2 255.255.255.0
 ip ospf 1 area 0
 tunnel source Loopback0
 tunnel destination 2001:4444::4
 tunnel mode ipv6
```

```
!
interface Tunnel24
 ip address 150.1.24.4 255.255.255.0
 ip ospf 1 area 0
 tunnel source Loopback0
 tunnel destination 2001:2222::2
 tunnel mode ipv6
```

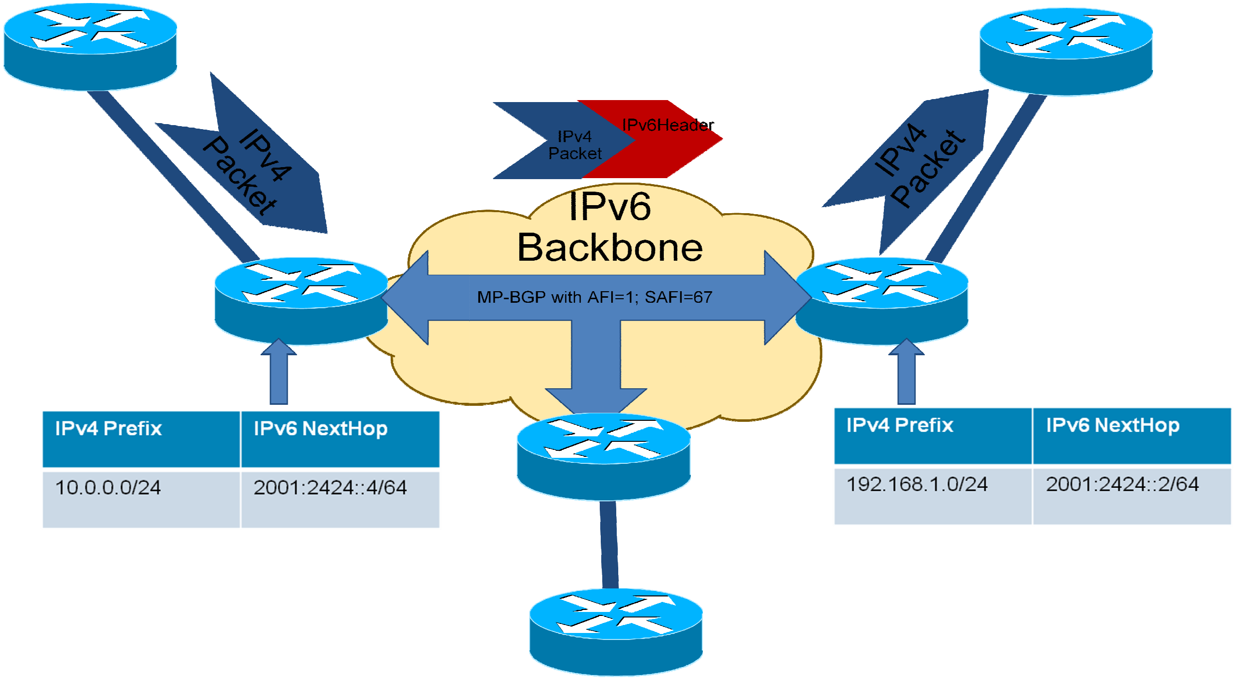
2.2.3. Automatic 4over6 tunnel

Automatic 4over6 solution is defined to address the scalability issue that we will face with static solution. As per RFC 5747, 4over6 framework works in a way that PE devices which are dual stacked with IPv6 as core facing and IPv4 as CE facing will use MP-BGP to exchange IPv4 routing details among PE devices.

MP-BGP with **AFI=1; SAFI=67** will be used to advertise IPv4 routing details from one PE to other with NLRI as IPv4 prefix and next hop as IPv6 address. Other PE devices on receiving the update will populate **Encapsulation table** with IPv4 prefix and IPv6 next hop details.

PE device on receiving IPv4 packet from CE facing interface will perform lookup in Encapsulation table to identify the IPv6 next hop. Now IPv4 packet will be encapsulated with IPv6 header with source as local IPv6 address and destination as the one pulled from the table.

Figure 2.12



IPv4 Prefix	IPv6 NextHop
10.0.0.0/24	2001:2424::4/64

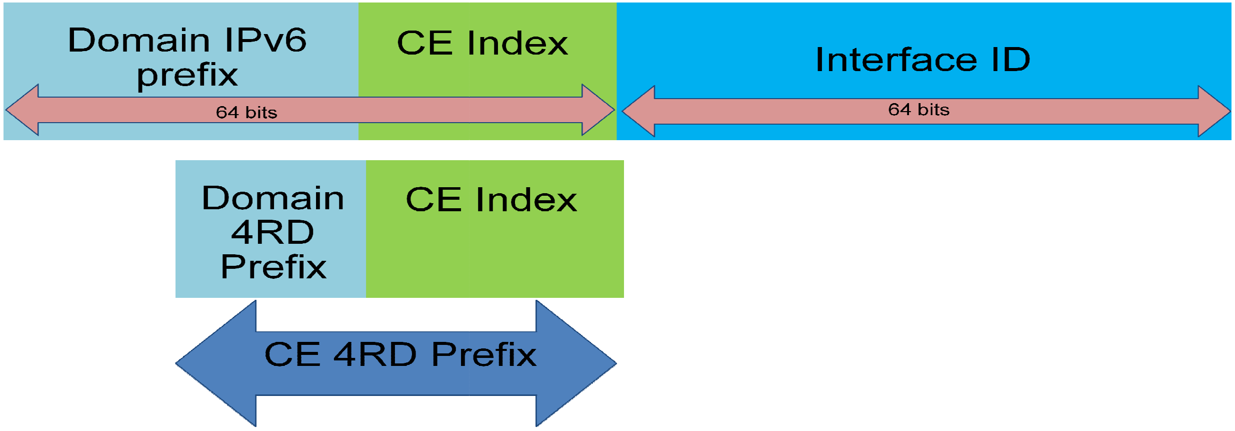
IPv4 Prefix	IPv6 NextHop
192.168.1.0/24	2001:2424::2/64

2.2.4. 4RD (IPv4 Rapid Deployment)

4RD (IPv4 Residual Deployment), which is a reverse of 6RD is a generic lightweight solution providing IPv4 connectivity over IPv6 infrastructure. 4RD domain, an IPv6 network belongs to an ISP comprise of 4RD Border Relay (BR) and 4RD Customer Edge (CE) devices.

4RD CE and BR will be configured with CE IPv6 address which comprise of 64 bit CE IPv6 prefix and 64 bit of Interface ID. CE IPv6 Prefix is made of variable length 4RD Domain IPv6 prefix followed by variable length CE index. Out of 64 bits, the length of the 4RD Domain IPv6 prefix is defined by mapping rule and the remainder will be CE index. Domain IPv6 prefix is made of SP specific IPv6 prefix followed by Domain 4RD prefix. Domain 4RD prefix and CE index is known as CE 4RD prefix”. Below is the visualized format of the IPv6 address

Figure 2.13



CE 4RD prefix is the field which helps identify the egress PE to reach the desired IPv4 destination. CE 4RD prefix can be of length up to 47 bits.

CE device on receiving IPv4 packet will check if the address belongs to 4RD domain. If yes, it will derive IPv6 address from the destination IPv4 address and will encapsulate with IPv6 header and mark Next Header as 4. This will be sent over IPv6 cloud. If the destination IPv4 address doesn't belong to 4RD domain, it will be sent to 4RD BR device for external communication.

P.S : The above technique is still in draft version and so may change later.

3. Protocol Translation technique

As part of migration, we may face up with situation where IPv4 client may have to communicate with IPv6 servers and vice versa. The above discussed solutions may help in migrating from IPv4 to IPv6 network but doesn't address direct communication requirement between IPv4 and IPv6 domains.

Below are few solutions used for translation,

- NAT-PT
- NAT64

3.1. NAT-PT (Network Address Translation – Protocol Translation)

IPv6 NAT-PT is a legacy protocol translator tool that helped having bidirectional communication between IPv4 and IPv6 domains.

NAT-PT router will be enabled with either static or dynamic translation rule programmed in the router. Static translation needs manual configuration of one to one address translation between IPv4 and IPv6 address. For example, if we have IPv6 packet to be sent to IPv4 domain. NAT-PT router will look into the table and get the required IPv4 address which are statically configured and rewrite the same before sending to IPv4 domain. Since translation rule is bidirectional, any reverse traffic can use the same rule and rewrite with IPv6 address while sending into IPv6 domain.

Dynamic translation rule is by allotting 96 bit IPv6 prefix for NAT process. Any IPv4 packet that needs to be sent to IPv6 domain while rewrite with 96 bit IPv6 prefix + embedded IPv4 address making it as 128 bit IPv6 address.

NAT-PT was officially deprecated by RFC 4966 due to operational issues and so is not advised for any new implementation.

3.2. NAT64 – Network Address Translation 6to4

NAT64 is a translator feature that helps enabling communication from IPv6 host to IPv4 internet. NAT64 is said to replacing NAT-PT and is currently recommended for deployment. As part of IPv6 migration, below are the 4 possible situations that requires Address translation,

- Interoperation between IPv6 network and IPv4 Internet
    - ✓ IPv6 Network to IPv4 Internet
    - ✓ IPv4 Internet to IPv6 Network
- Interoperation between IPv4 network and IPv6 Internet
    - ✓ IPv4 Network to IPv6 Internet
    - ✓ IPv6 Internet to IPv4 Network
- Interoperation between IPv6 network and IPv4 network
    - ✓ IPv6 Network to IPv4 Network
    - ✓ IPv4 Network to IPv6 Network
- Interoperation between IPv6 Internet and IPv4 Internet
    - ✓ IPv6 Internet to IPv4 Internet
    - ✓ IPv4 Internet to IPv6 Internet

NAT64 can be used to address the above requirement. NAT64 can be implemented as stateless or stateful based on the requirement to address the above mentioned scenarios.

Stateless NAT64 uses mapping algorithm that derives IPv4 address from IPv6 address. Stateless NAT64 requires IPv6 host to be configured with IPv4-Translatable-IPv6 address which is basically NAT64 prefix +global IPv4 address+suffix. NAT64 router that connects both IPv4 and IPv6 domain will be configured with NAT prefix which is 96 bits or lesser. When IPv6 packet reaches NAT64 router, IPv4 address will be unearthed and will be encapsulated after rewriting the address. When IPv4 packets reaches NAT64 router, it will use the NAT64 prefix to build the IPv6 address.

As the name states, Stateless NAT64 doesn't need to maintain any translation table as IPv4 address will be embedded in IPv6 address. But this comes with its own limitation. Stateless NAT64 requires global IPv4 address from SP domain to be allotted for this IPv6 domain and so this doesn't help addressing IPv4 depletion issue.

Stateful NAT64 doesn't require any specific requirement on addressing for IPv6 host as like Stateless NAT64. This mechanism uses algorithm that translates IPv4 to IPv6 and vice versa from configured stateful prefix. NAT64 router on receiving IPv6 packet will rewrite with IPv4 address from mentioned IPv4 range and will create a table with mapping details. The same works vice versa. Since mapping table is maintained, there is no requirement to have IPv4 embedded into IPv6 address. Stateful NAT64 supports dynamic translation of address/port while initiated from IPv6 to IPv4 network while static binding is required to support communication initiated form IPv4 to IPv6 network.