

IPv6: More than Meets the Eye

Nick Russo, Technical Leader

Cisco Systems

<http://njrusmc.net>

NANOG 82

June 14-16, 2021

Stuff Everyone Knows

- IPv6 is a lot bigger than IPv4
- IPv6 packets cannot be fragmented in transit
- ICMPv6 replaces ARP, IGMP, and IRDP
- Stateful NAT66 and IPv6 MSDP don't exist

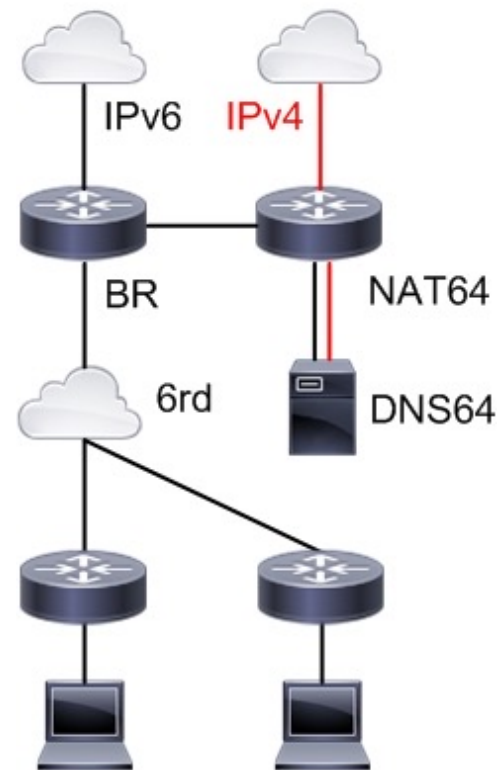
Things IPv4 Can't Do Easily

- Signal hop-limit (TTL) and MTU to clients
- Statelessly generate unique addresses
- Statelessly communicate non-address information
- Dynamically allocate entire prefixes
- Embed in-address information

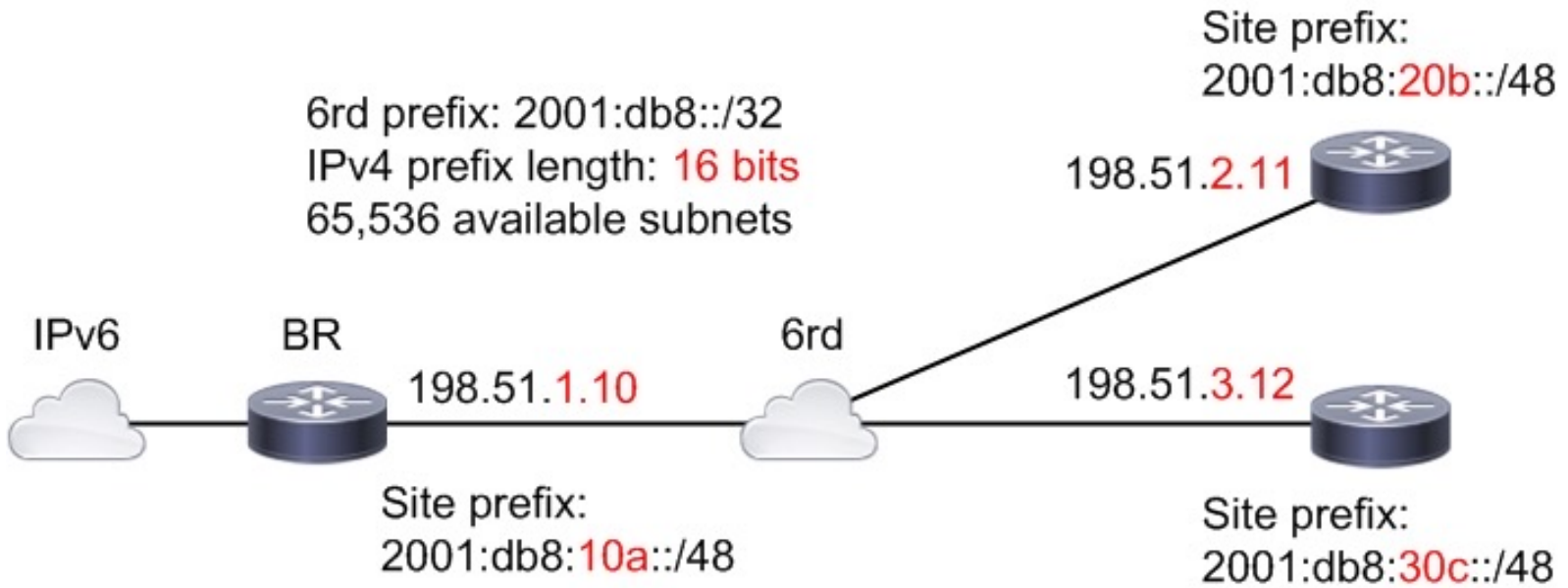
Architecture 1: Unlimited Scale WAN

Unlimited Scale WAN Overview

- Requirements:
 - Any-to-any at scale
 - Minimal OAM
 - Easy client onboarding
 - Optional encryption
 - IPv4 interworking
- Solution
 - 6rd overlay / GETVPN
 - Stateless DHCPv6
 - SLAAC

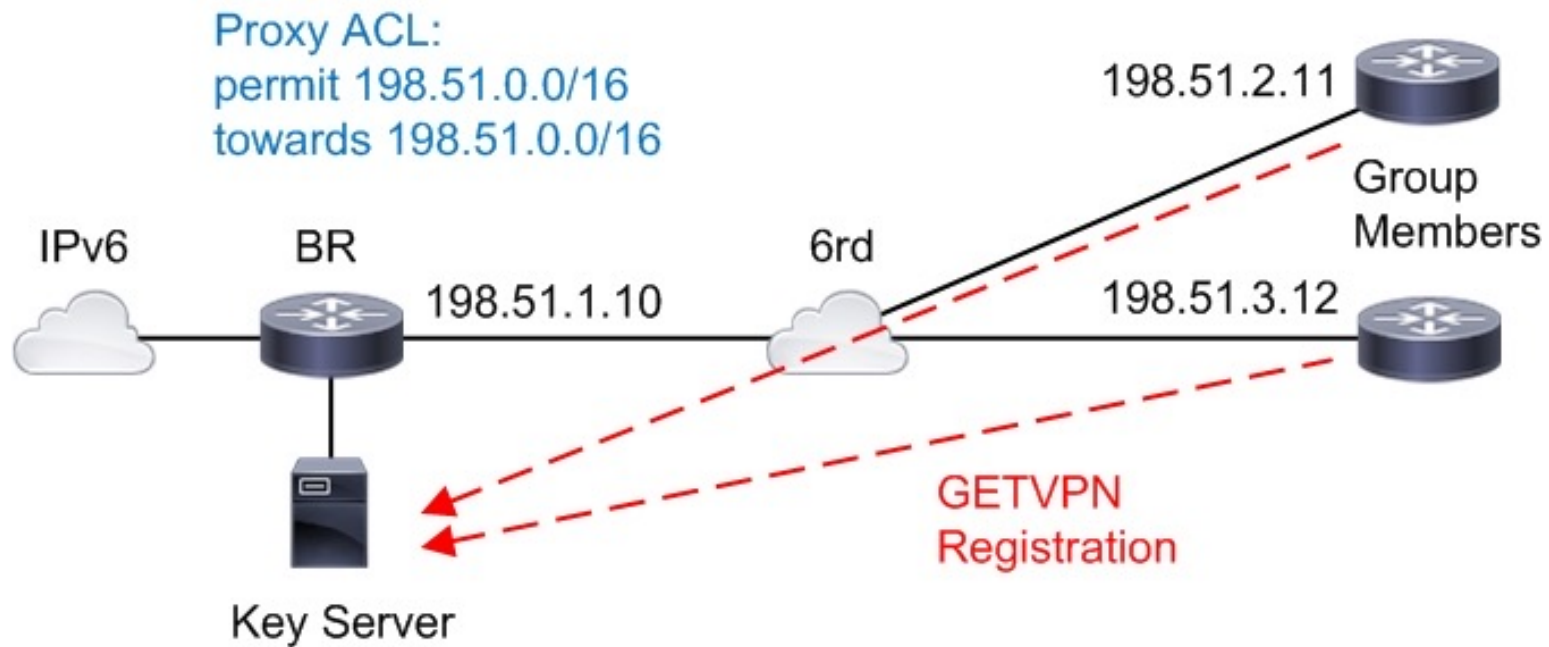


How It Works



IPv6 Inner	IPv4 Outer
S=2001:db8:10a::11	S=198.51.1.10
D=2001:db8:30c::33	D=198.51.3.12

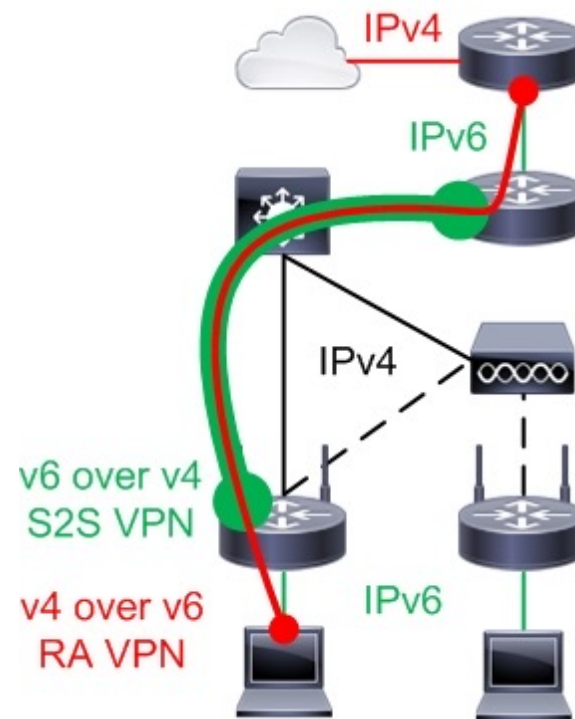
Securing the Transport



Architecture 2: Low-touch Mobile Sites

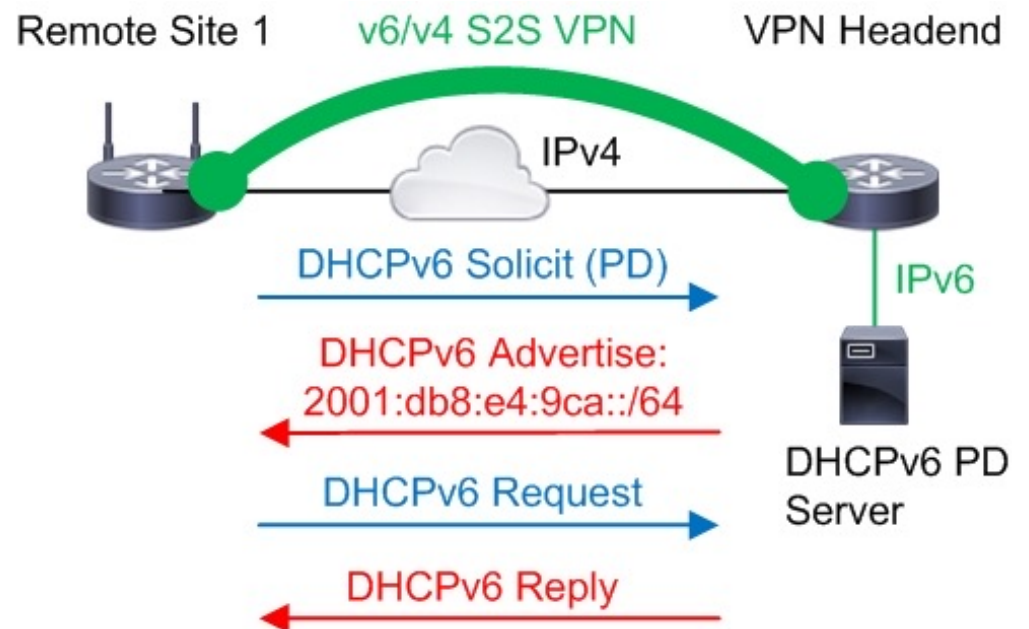
Low-touch Mobile Sites Overview

- Requirements:
 - Hub-to-spoke at scale
 - Multiple transports
 - Fully dynamic sites
 - Top-down automation
- Solution
 - IPv6 BGP over DMVPN
 - DHCPv6 PD
 - Stateless DHCPv6
 - SLAAC



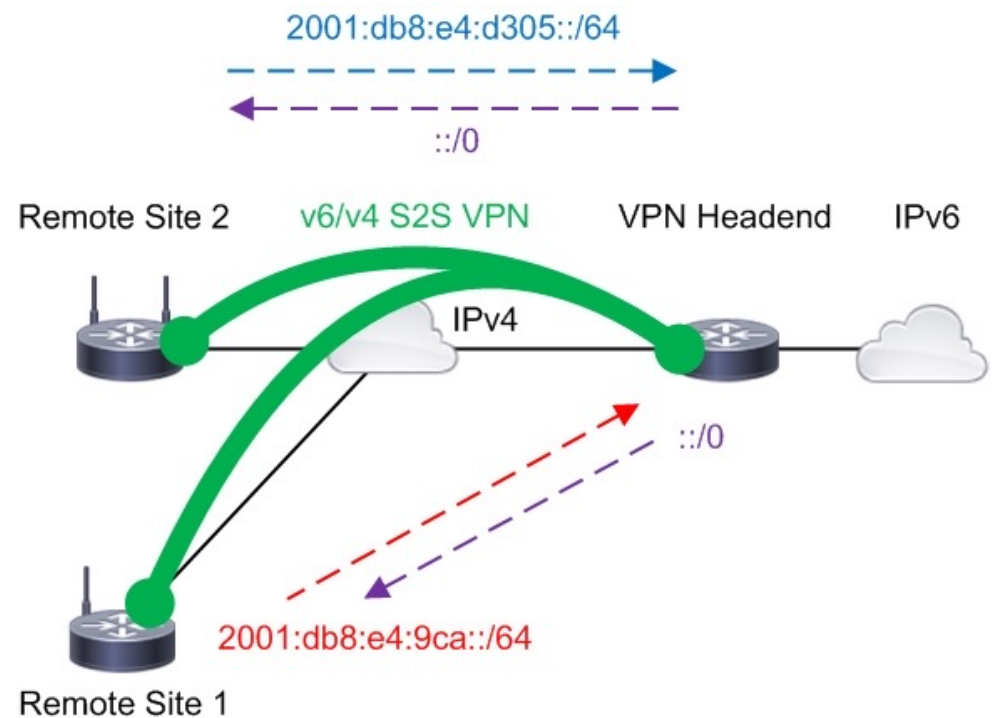
Obtaining an IPv6 Prefix

- Site requests prefix
- Server issues prefixes
 - Stateful mapping
- Can include options:
 - Domain name
 - DNS server
 - SNTP servers

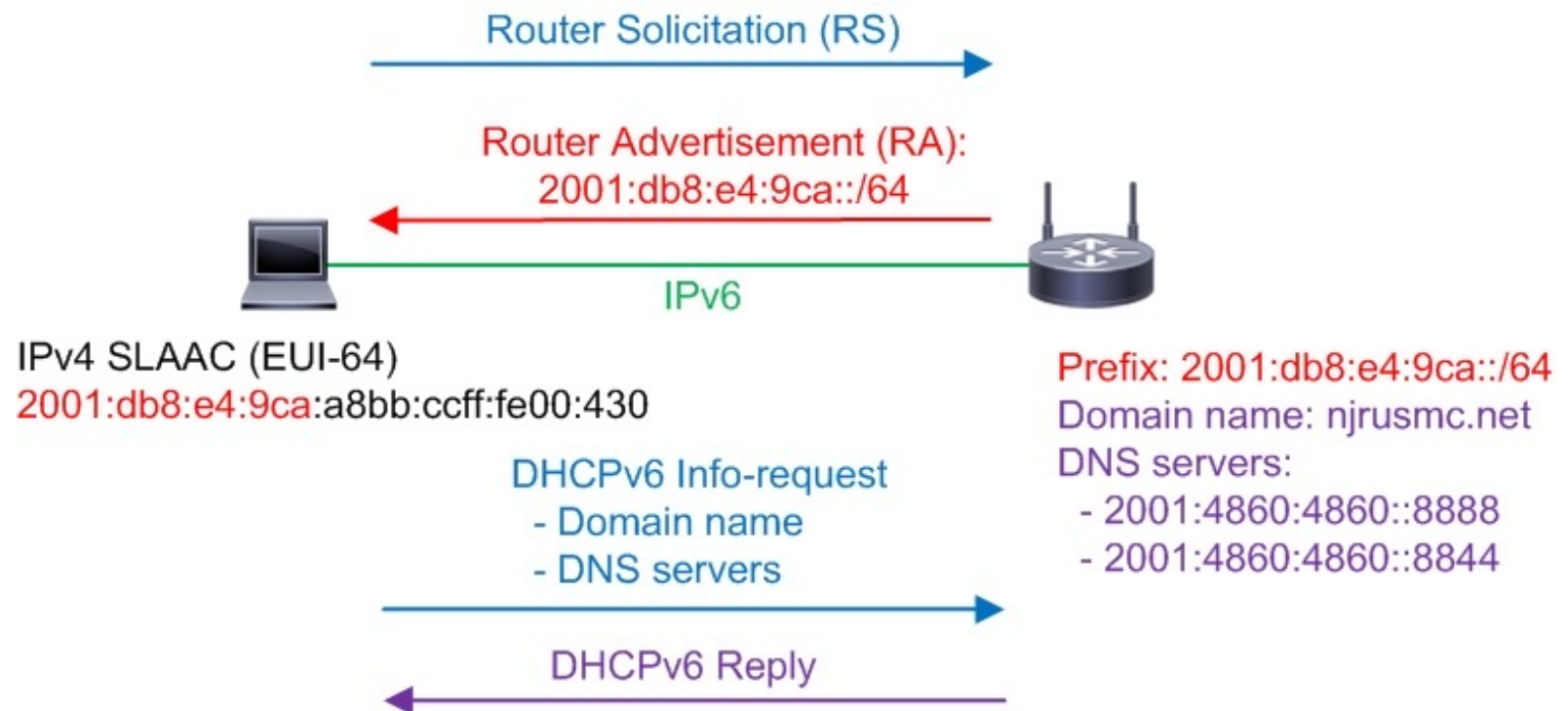


Upstream BGP Routing

- Spokes initiate to hubs
- Advertise PD prefixes
- Receive default route
- Improves resilience:
 - Allows multi-link sites
 - Allows multi-transport



Addressing the Endpoints



What About Automation?

- On-box:
 - Autoconfig hostname
 - Autoconfig EUI-64 /128
 - Send /128 via BGP to hub
- Centralized:
 - Parse /128 from router
 - Build /128 from MACs
 - Do whatever afterwards!

```
resp = conn.send_command("show bgp all", use_genie=True)
v6_rte = resp["address_family"]["ipv6 unicast"]["routes"]

# Initialize Ansible YAML inventory dictionary
ansible_inv = {"all": {"children": {"remotes": {"hosts": {}}}}}

# Iterate over all collected BGP prefixes
for index, prefix in enumerate(v6_rte.keys()):

    # Create an IPv6 network representing the specific prefix
    prefix_net = IPv6Network(prefix.lower())

    # Test for subnet containment and for /128 mask
    if prefix_net.subnet_of(mgmt_net) and prefix.endswith("/128"):

        # Assemble inventory item and update inventory dict
        prefix_str = DoubleQuoted(prefix_net.network_address)
        ansible_inv["all"]["children"]["remotes"]["hosts"].update(
            {f"node_{index + 1}": {"ansible_host": prefix_str}}
        )
```

Some Samples ... It's Cool!

ans_inv_from_eui64.py

```
---
all:
  children:
    remotes:
      hosts:
        node_1:
          ansible_host: "2001:db8::a2bb:ccff:fe00:0300"
          original_mac: "a0bbcc000300"
        node_3:
          ansible_host: "2001:db8::a0bb:ccff:fe00:0300"
          original_mac: "a2bbcc000300"
```

ans_inv_from_bgp.py

```
---
all:
  children:
    remotes:
      hosts:
        node_1:
          ansible_host: "2001:db8:aaaa:aaaa::1"
        node_2:
          ansible_host: "2001:db8:aaaa:aaaa::2"
        node_3:
          ansible_host: "2001:db8:aaaa:aaaa::3"
```



Thank you

Nick Russo <http://njrusmc.net>
Technical Leader, Cisco Systems

References

Architecture 1: Unlimited Scale WAN:

http://njrusmc.net/pub/6rd_getvpn.pdf

Architecture 2: Low-Touch Mobile Sites

http://njrusmc.net/pub/csfc_macp.pdf

Code Samples:

<https://github.com/nickrusso42518/ipv6-tools>