Internet Protocol Version 6: advanced features

The innovative aspects of IPv6
Autoconfiguration

- **Addresses are composed by**
  - Information held by routers (network address)
  - Information locally available (interface identifier)

- **Addressing space: usually received from the provider**
  - Switching provider implies changes in addressing space

- **A protocol is needed for address assignment and propagation of configuration information, and it should be designed for:**
  - making it easier to re-number devices
  - avoiding manual configuration (Plug and Play)
  - avoiding the use of servers, for small sites
  - managing:
    - Link Local Address
    - Site Local Address
    - Global Address

- **What we need to configure**
  - Host, Routers, DNS, applications (e.g. web server, IP-based licenses, access lists, ...)

Stateless autoconfiguration for hosts

- **Basic configuration**
  - Generate a local address
  - Probing to verify its uniqueness

- **The host can communicate with devices within its own LAN, without router intervention**

- **If a router exists**
  - It is possible to listen to Router Advertisement messages
  - (or) to send a Router Solicitation message
  - Configuration and probing the uniqueness of the address

Router advertisement with 2 network prefixes
Stateless autoconfiguration for hosts

- Hosts are always listening to router messages
  - A host can be re-configured at run time
  - Renumbering is easier
    - For example, it is possible to switch from a link local to a global address
    - Or from a global address to another one
- State of addresses
  - Preferred
  - Deprecated
  - (invalid)
Duplicate Address Detection

- **Needed to test the uniqueness of an IPv6 address**
  - For example, it is used to verify the uniqueness of a configured address

- **Procedure**
  - Solicited Node Multicast Address (and corresponding MAC address) corresponding to the IPv6 address to verify is prepared
  - A ICMP Neighbor Discovery message is sent in multicast packet with the address to be verified as a target
  - Wait for a response for at least 1 sec
  - If no answer is received, the address is considered valid

- **Start router discovery phase once done**
Host autoconfiguration: stateful

- DHCPv6: client/server model
- Compatible with stateless autoconfiguration
- Possible messages:
  - Solicit (sent to the all-agents address: FF02::1:2)
  - Advertise
  - Request (sent to the all-agents address: FF02::1:2)
  - Reply
  - Release
  - Reconfigure
Autoconfiguration for routers

- Router Renumbering (RFC 2894)

- Router Renumbering packets
  - they include PCOs (Prefix Control Operations)
    - Match-Prefix: specifies the operation
    - Use-Prefix
  - They are transported in ICMPv6 packets

- Two types of Router Renumbering messages
  - Commands: sent to the routers
  - Results: sent by routers as responses to commands
Autoconfiguration and privacy

- The least significant 64 bits in an IPv6 address never change, when obtained with stateless configuration
  - Privacy problems (es. traceability)
- RFC 4941, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6”
  - Algorithm
    - Stateless address (64 bit) + 64 bit random number (or previous “privacy” address)
    - MD5 is computed over the previous result
    - The most significant 64 bits are selected, and the bit 6 (universal/local) is set to 0 (“local”)
    - The result is used as interface identifier
    - This value is stored to be used in the next address configuration
- Each host may have several different addresses ("default" and "privacy"), used to accept/generate connections
  - Selection of address may be available to the user through suitable API
**DNS**

- **DNS: Domain Name System**
  - Distributed data base
  - It associates IP addresses and names
  - The atomic element in DNS is the resource record (RR)

- **Record:**
  - type A (canonical): used for 32 bit addresses
  - type AAAA: used for 128 bit (RFC1886) addresses
    - IP6.INT domain
  - It is possible to specify the type **ANY** in a query, in order to receive both type A and type AAAA records (if they exist)
  - It is up to the application to select which one to use (modified application)
Scoped addresses

c:\>netsh interface ipv6 show address

<table>
<thead>
<tr>
<th>Interface 1: Loopback Pseudo-Interface 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr Type</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface 10: Wireless Network Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr Type</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface 9: Local Area Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr Type</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface 12: Local Area Connection* 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr Type</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Public</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface 27: Bluetooth Network Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr Type</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface 31: Local Area Connection* 25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr Type</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

c:"
Why a scope is required?

FE80::0237:00FF:FE02:a7FD

socket

application

TCP/UDP

IPv6

DL  DL

PHY  PHY

FE80::0237:00FF:FE02:a7FD
Sintax

- A scoped address is composed by an IPv6 address followed by a % by a number identifying the interface

Example:
  - FE80::0237:00FF:FE02:a7FD%19

The value of the scope is selected according some internal criterion
Security and IPv6 addresses

- **Network scanning**
  - More difficult, from a theoretical point of view, because the larger number of combinations available (64 bits per LAN)
  - In reality, it is possible to use tricks to limit the address space to be scanned
    - Addresses are assigned sequentially (from ::1 on)
    - Stateless address autoconfiguration (48 bits to be scanned)
    - Hosts with sequential MAC addresses (once one is found, all the others have similar MACs)
    - Start scanning with known OUI (NIC manufacturers → 24 bit)
    - IPv6 addresses derived from IPv4 ones
    - Often, an IPv6 host uses dual stack, hence it is possible to scan the IPv4 space
  - Address harvesting, used to find addresses to be used as "seeds"
    - Host published in DNS
    - Analysis of log files of an host (e.g. tracker P2P, web server)

- **DDoS**
  - An attacker may use several different addresses from the same machine (potentially, a whole /64)
**IPv6 routing**

- **Two aspects have to be considered:**
  - How to handle the forwarding of the data packets
    - I.e., how to structure the routing table
  - How to distribute routes across the network
    - I.e., how to engineer routing protocols

- **Additional issue**
  - We expect IPv6 not to be enabled alone (at least in the short term), hence we need to handle both IPv4 and IPv6 routing tables at the same time
Enabling IPv6 routing

- Often, IPv6 routing is turned off by default
  - Although the device may be IPv6-capable
- Cisco basic configuration
  - The `ipv6 unicast-routing` command has to be applied to enable IPv6
    - Without this, even IPv6 addresses cannot be configured
  - If the IPv6 unicast routing is enabled, the router start forwarding IPv6 packets between its interfaces using the IPv6 routing table

```
Router(config)#ipv6 unicast-routing
Router(config)#exit
```
IPv6 routing table

- IPv6 routing uses the longest-match prefix for route selection
  - The same as in IPv4
  - In case a packet matches two routes, the one with the longest prefix is selected
- IPv6 routing table is handled and managed separately from the IPv4 routing table
  - IPv6 and IPv4 are two independent protocols
Options for routing protocols

- Two alternative approaches
  - Integrated Routing
    - A single protocol transports multiple address families
  - Ships in the night
    - Each address family uses a distinct protocol
      - Protocols are completely independent one from the other

- The choice is done by the specific routing protocol
  - Some choose the *integrated routing* approach, other prefer the *Ships in the night*
Integrated Routing

- A single protocol transports multiple address families

Advantages

- More optimized: if fault/network change occurs, the protocol discovers the fault for both address families

Questions

- Which address family (IPv4, IPv6) do we use to transport protocol messages?
  - We cannot use both
- What about if the protocol we use fails (or has bugs, etc.)?
- What about if the two topologies (IPv4 and IPv6) are different?
  - In the migration path, it hard to have both topologies that look the same
Ships in the night

- **Each address family uses a distinct protocol**
  - Protocols are completely independent one from the other

- **Advantages**
  - Supports different routing protocols / topologies in IPv4 and IPv6
  - A problem/fault/bug in a routing protocol does not affect the routing in the other
  - Easier migration (each routing protocol generates messages of the address family it belongs to)

- **Questions**
  - If fault/network change occurs, both protocols have to discover the fault, each one with its timetable
    - Duplicated messages, hence overhead
IPv6 routing protocols: available options

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Routing approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>Ships in the night</td>
</tr>
<tr>
<td>RIPng</td>
<td>Ships in the night</td>
</tr>
<tr>
<td>EIGRP for IPv6</td>
<td>Ships in the night</td>
</tr>
<tr>
<td>OSPFv3</td>
<td>Ships in the night (although Integrated routing can be supported through Instance_ID)</td>
</tr>
<tr>
<td>IS-IS for IPv6</td>
<td>Integrated routing</td>
</tr>
<tr>
<td>MP-BGP4</td>
<td>Both (depends on the configuration), although the most common deployment follows the “Integrated Routing” approach because of the necessity to use the AS number (which is the same for both IPv4 and IPv6) for the BGP process</td>
</tr>
</tbody>
</table>

Diagram:

- AS 1
- BGP
- AS 2
- RIPng
- OSPFv6
- IS-IS
- EIGRP
Administrative distance

- Administrative distance remains the same as in IPv4

<table>
<thead>
<tr>
<th>Route Source</th>
<th>Administrative Distance (Default)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected interface</td>
<td>0</td>
</tr>
<tr>
<td>Static route</td>
<td>1</td>
</tr>
<tr>
<td>Enhanced Interior Gateway Routing Protocol (EIGRP)</td>
<td>5</td>
</tr>
<tr>
<td>summary route</td>
<td></td>
</tr>
<tr>
<td>External Border Gateway Protocol (BGP)</td>
<td>20</td>
</tr>
<tr>
<td>Internal EIGRP</td>
<td>90</td>
</tr>
<tr>
<td>IGRP</td>
<td>100</td>
</tr>
<tr>
<td>OSPF</td>
<td>110</td>
</tr>
<tr>
<td>Intermediate System-to-Intermediate System (IS-IS)</td>
<td>115</td>
</tr>
<tr>
<td>Routing Information Protocol (RIP)</td>
<td>120</td>
</tr>
<tr>
<td>Exterior Gateway Protocol (EGP)</td>
<td>140</td>
</tr>
<tr>
<td>On Demand Routing (ODR)</td>
<td>160</td>
</tr>
<tr>
<td>External EIGRP</td>
<td>170</td>
</tr>
<tr>
<td>Internal BGP</td>
<td>200</td>
</tr>
<tr>
<td>Unknown</td>
<td>255</td>
</tr>
</tbody>
</table>
Showing IPv6 routing table

C2800#sh ipv6 route
IPv6 Routing Table - 15 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
U - Per-user Static route
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

O 2013::/112 [110/65]
   via FE80::20F:34FF:FEE7:ABDE, FastEthernet1
C 2015::/112 [0/0]
   via ::, Serial0/2/0
L 2015::2/128 [0/0]
   via ::, Serial0/2/0
O 2016::/112 [110/65]
   via FE80::223:EBFF:FE44:C6EE, FastEthernet0
   via FE80::20F:34FF:FEE7:ABDE, FastEthernet1
C 2017::/64 [0/0]
   via ::, FastEthernet0/1
L 2017::2/128 [0/0]
   via ::, FastEthernet0/1
L FE80::/10 [0/0]
   via ::, Null10
L FF00::/8 [0/0]
   via ::, Null10
S ::/0 [1/0]
   via FE80::20D:BCFF:FE89:29A3, FastEthernet2

Following addresses are automatically inserted in the routing table:

- Connected networks
- Interface addresses
- Link-local prefix
- Multicast prefix
- Default ipv6 route
Routing protocols

- Inside the same autonomous system
  - RIPng
  - OSPFv3

- Inter-AS
  - MP-BGP