All about IPv6

Josh Wilcox (jw14g24)

April 26, 2025

Table of Contents

1. NDP - Neighbour Discovery Protocol
   - Functions
   - More on Router Advertisement
2. SLAAC
   - RFC4862 Method of SLAAC Autogeneration
   - RFC7217 Method

NDP - Neighbour Discovery Protocol

Overview of NDP

- Responsible for gathering information required for network communication between **IPv6** devices
- Operates on the **internet layer**
- Includes the configuration of local connection and DNS servers and **gateways** within a local network

Subsections of NDP - Neighbour Discovery Protocol

- Functions
- More on Router Advertisement

Functions

NDP defines five *ICMP* packets

- **Router Solicitation** - Type 133
  - Hosts requesting for **routers** to reveal themselves so they can be located
  - Leads routers to generate **Router Advertisements** as soon as they see this
    - Routers would usually just advertise at scheduled times
    - Router soliciation requests would override this schedule
- **Router Advertisement** - Type 134
  - Advertisement of the presence and details of routers on a local network
  - Either sent periodically or on instance of a Router Solicitation from a host on the network
- **Neighbour Solicitation** - Type 135
  - Used by nodes in a network to determine the **link-layer** address of a neighbour
  - Also can verify that neighbours are rechable via cached-addresses
- **Neighbour Advertisement** - Type 136
  - Directly responds to Neighbour Solicitation messages giving the link-layer address of the advertising host
  - Advertisement can also be unsolicited - specifically if the link-layer address of the advertiser changes

These Packets afford the following functions:

- Router Discovery
  - Uses Router Solicitation and Advertisement messages
  - Helps hosts find and select default routers
  - Provides prefix information for the network
- Address Autoconfiguration - See **SLAAC**
  - Allows hosts to automatically configure IPv6 addresses
  - Can work with or without DHCPv6
  - Uses prefix information from Router Advertisements
- Address Resolution
  - Maps IPv6 addresses to MAC addresses
  - Replaces ARP from IPv4
  - Uses Neighbor Solicitation and Advertisement messages
- Next-Hop Determination
  - Finds the best path to reach a destination
  - Determines if destination is on-link or requires a router
  - Uses prefix information from Router Advertisements
- DNS server assignment
  - Enables automatic discovery of DNS servers
  - Can work alongside DHCPv6
  - Provides DNS recursive server addresses to hosts

Process of Router Advertisement

Router Advertisements function as follows:

- A host sees or solicites a Router Advertisement
- The RA message carries the **IPv6 Prefix** to use for this network
- The RA's rouce address provides the default router address - obviously as it has come from the router!
- Flags in the RA message define how addresses are resolved (either through SLAAC or DHCP)
- DNS server information can be included in an RA

SLAAC

Overview of Stateless Address Autoconfiguration

- Mechanism that enables hosts on the network to configure a **unique** IPv6 address without the need for other devices on the network keeping track of addresses already assigned
- There is no central server that keeps track of what addresses are assigned and what are still available
- Nodes themselves are responsible to resolve any duplicate **address conflicts**
  - Literally the simplest way to deal with SLAAC address conflicts - If duplicate detected, just generate a new one

RFC4862 Method of SLAAC Autoconfiguration

Description of the RFC4862 Method

- Node receives Router Advertisement containing network prefix
- Node generates Interface ID using a Modified EUI-64 (based on MAC address):
  - Splits 48-bit MAC address in half
  - Inserts FFFE in the middle
  - Flips 7th bit (Universal/Local bit)
  - Results in a 64-bit interface identifier
- Combines network prefix with Interface ID to form full IPv6 address
- Performs Duplicate Address Detection (DAD):
  - Sends Neighbor Solicitation for tentative address
  - If no response, address is unique and usable
  - If duplicate detected, generates new Interface ID
- Once verified unique, node assigns address to interface
- Process repeats periodically or when network changes

Example of the RFC4862 Method

RFC4862 Example

1. Router advertises network prefix: 2001:db8::/64
2. Node with MAC address 00:1A:2B:3C:4D:5E:
   - Converts MAC to EUI-64: 021A:2BFF:FE3C:4D5E
3. Forms complete IPv6 address:
   - 2001:db8::021A:2BFF:FE3C:4D5E
4. Sends NS for DAD check
5. If unique, assigns address to interface

RFC7217 Method of SLAAC Autoconfiguration

The Need for RFC7217

- Very similar to the RFC4862 Method - But **does not use the EUI-64**
- **Embedding a MAC address** in a global address is a **privacy nightmare**
  - MAC addresses are unique and persistent, allowing devices to be tracked across networks
  - Attackers can derive your device's MAC address from your IPv6 address
  - Makes it possible to track user movement and behavior across different networks
  - Creates a permanent identifier that stays with the device even when changing networks
- However, networks require **stable IPv6 Addresses** for each host based on its hardware
  - Stable addresses are needed for access control lists and security policies
  - Required for maintaining persistent connections and services
  - Allows network administrators to track legitimate network issues

The RFC7217 Method

- Uses a pseudorandom function (PRF) to generate a Random but stable Identifier (RID):
  - `RID = F(Prefix, Net_Iface, Network_ID, DAD_Counter, secret_key)`
  - `F()`: Cryptographic hash function (e.g., SHA-1, SHA-256, but NOT MD5)
  - `Prefix`: IPv6 prefix from Router Advertisement or link-local prefix
  - `Net_Iface`: Stable identifier for the network interface
  - `Network_ID`: Optional subnet identifier (e.g., Wi-Fi SSID)
  - `DAD_Counter`: Starts at 0, increments on address conflicts
  - `secret_key`: $\geq$ 128-bit random key set at OS install/network stack bootstrap
- Key requirements of the function F():
  - Must not be computable without knowing the secret key
  - Must be cryptographically difficult to reverse
  - Should produce at least 64-bit output
  - Can be implemented as hash of concatenated parameters

Example of the RFC7217 Method

RFC7217 Example

1. Router advertises prefix: `2001:db8::/64`
2. Node computes RID using SHA-256:
   - Parameters:
     - Prefix: `2001:db8::/64`
     - Net_Iface: `eth0`
     - Network_ID: `HomeWiFi`
     - DAD_Counter: `0`
     - secret_key: `<128-bit random value>`
   - Resulting RID: `8e9b:1c2d:3a4b:5c6d`
3. Forms complete address: `2001:db8::8e9b:1c2d:3a4b:5c6d`
4. If DAD fails, increments DAD_Counter and recomputes