IPv6 Tutorial

IPv6 Camp

IPv6 associated Protocols

Mohsen.Souissi@nic.fr

30 June 2008
Rennes, France

Agenda

- IPv6 Associated Protocols
- IPv6 support in the DNS

New Protocols

- New features specified in IPv6 Protocol (RFC 2460 DS)
- Neighbor Discovery (ND) (RFC 4861 DS)
- Auto-configuration :
  - Stateless Address Auto-configuration (RFC 4862 DS)
  - DHCPv6: Dynamic Host Configuration Protocol for IPv6 (RFC 3315 PS)
  - Path MTU discovery (pMTU) (RFC 1981 PS)

New Protocols (2)

- MLD (Multicast Listener Discovery) (RFC 2710 PS)
  - Multicast group management over an IPv6 link
  - Based on IGMPv2
  - MLDv2 RFC 3810 (equivalent to IGMPv3 in IPv4)
- ICMPv6 (RFC 4433 DS) "Super" Protocol that :
  - Covers ICMP (v4) features (Error control, Administration, …)
  - Transports ND messages
  - Transports MLD messages (Queries, Reports, …)

Neighbor Discovery

- Protocol features:
  - Router discovery
  - Prefix(es) discovery
  - Parameters discovery (link MTU, Max Hop Limit, ...)
  - Address auto-configuration
  - Address resolution
  - Next Hop determination
  - Neighbor Unreachability Detection
  - Duplicate Address Detection
  - Redirect

Neighbor Discovery (2)

- ND specifies 5 types of ICMP packets:
  - **Router Advertisement** (RA):
    - Periodic advertisement (list of prefixes, MTU, Hop Limit)…
    - Triggered by a *RS*
  - **Router Solicitation** (RS):
    - the host needs RA immediately (at boot time)
  - **Neighbor Solicitation** (NS):
    - link-layer @ resolution, unreachability / duplication detection
  - **Neighbor Advertisement** (NA):
    - answer to a NS packet, advertise change of physical address
  - **Redirect** :
    - Used by a router to inform a host of a better route to a given destination
    - Hardly used nowadays in production environment (security!)

Path MTU discovery (RFC 1981)

- Fragmentation not allowed by IPv6 intermediate systems

- **link MTU** : maximum packet length (bytes) that can be transmitted on a given link without fragmentation
- **Path MTU** (or pMTU) = min { link MTUs } for a given path

- Protocol operation
  - makes assumption that pMTU = link MTU to reach a neighbor (first hop)
  - if there is an intermediate router such that link MTU < pMTU ➔ it sends an ICMPv6 message: "Packet size Too Large"
  - source reduces pMTU by using information found in the ICMPv6 message

- RFC 4821 (A new method): Packetization Layer Path MTU Discovery

Auto-configuration: Stateless vs Stateful

- Hosts should be plug & play

- Stateless auto-configuration (RFC 4862 DS)
  - No servers => useful in LANs
  - Does not apply to routers: they require manual configuration

- Stateful auto-configuration
  - Use of DHCPv6 RFC 3315 => rather an ISP architecture
  - Client/Server/Relay architecture
  - Can be used to complement stateless auto-configuration

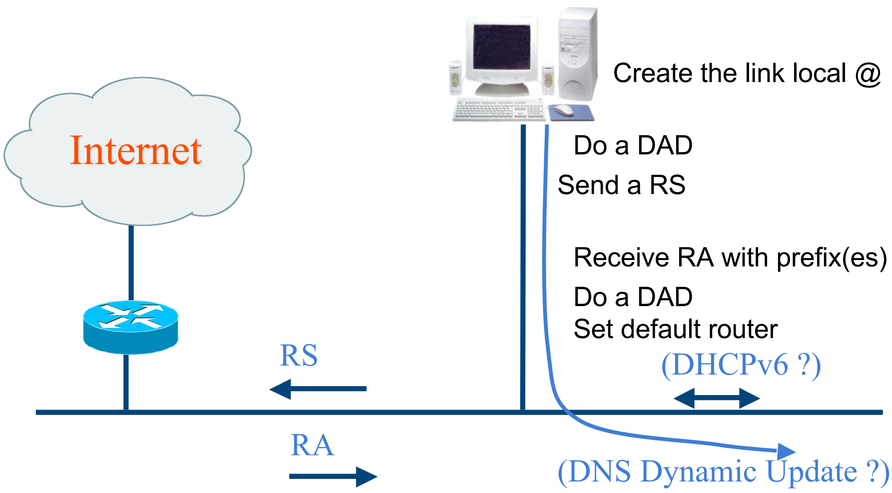
Stateless Auto-configuration

- Allows a host to build its own IPv6 addresses from:
  - its MAC @ (for both link-local and global addresses)
  - prefixes sent in router advertisements (RA) by routers on the link (only for global addresses)
- Addresses are not automatically registered in the DNS
  - Need for DNS Dynamic Update (RFC 2136 and RFC 3007)
- Several steps:
  - Link-local addresses creation
  - Duplicate addresses detection (DAD)
  - Discover the routers on-link (RS/RA)
  - Configure hosts global addresses
  - Configure other parameters: default router, link MTU, …
- Recursive DNS server (cache server) info is not provided in RAs

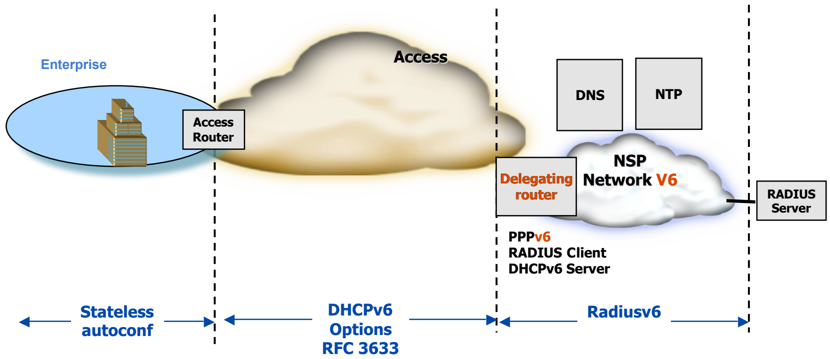
Stateful Auto-configuration

- *Dynamic Host Configuration Protocol for IPv6*
  - RFC 3315 (PS)
  - IPv4 version of DHCP (RFC 2131, DS)
    - based on BOOTP (RFC 951)
- Server
  - Memorizes client's state
  - Optionally provides the client with IPv6 addresses and configuration parameters
- Client
  - Sends requests and acknowledgements in accordance with the protocol (DHCP)

Auto-configuration: Summary



Prefix Delegation



Routing

Routing Protocols

- RFC 2080 (PS) & 2081 (INFO) : RIPng
- RFC 2740 (PS) : OSPF v3
- `draft-ietf-isis-ipv6-07.txt`: IS-IS (RFC XXXX)
- RFC 2545 (PS) : based on MBGP (RFC 2848)
  - Multi-extension protocol for BGP-4

⇒ No major differences with IPv4

- RFC 3031 : MPLS : MultiProtocol Label Switching
- RFC 4798 : 6PE (MPLS Provider Edge IPv6 routing)

IPv6 & DNS: DNSv6

Reminder: Two approaches to the DNS

- The DNS seen as a database
  - Stores different types of resource records (RR), including those related to IPv4 and IPv6 addresses: SOA, NS, A, AAAA, MX, PTR, TXT, …
  - ***DNS database is IP transport version agnostic!***
- The DNS seen as a TCP/IP application
  - The service is accessible in either transport modes (UDP/TCP) and over either IP versions (v4/v6)
  - ***Information given over either IP version MUST BE CONSISTENT!***

DNS Extensions for IPv6 Support
RFC 3596 (DS)

❑ ***Forward lookup*** (‘Name à IPv6 Address’):

- A **new** Resource Record (RR) : **‘AAAA’**
- The **‘AAAA’** RR is for IPv6 what the **‘A’** RR ‘is for IPv4
- Example:

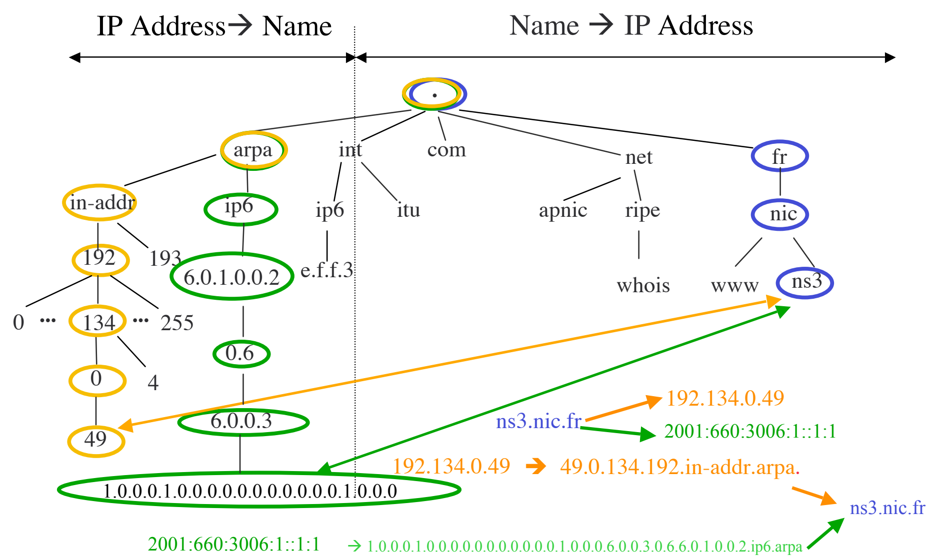
www.afnic.fr.	IN	**A**	192.134.4.20
	IN	**AAAA**	2001:660:3003:2::4:20

❑ ***Reverse lookup*** (‘IPv6 Address à Name’):

- **A new and dedicated** reverse tree: **ip6.arpa**
- The IPv6 equivalent to the IPv4 dedicated in-addr.arpa tree
- Populated with **PTR** and NS RRs with ***nibble*** (4 bits) ***boundaries***
- Example:

0.2.0.0.4.0.0.0.0.0.0.0.0.0.0.0. 2.0.0.0.6.0.0.3.0.6.6.0.1.0.0.2.ip6.arpa. PTR www.afnic.fr.

Lookups in an IPv6-aware DNS Tree



Recursive Name Servers Information Discovery

- A ***Stub Resolver*** needs a *Recursive Name Server* **address** to which it sends **name resolution** queries

- In the IPv4 world, this DNS information is:
  - Either configured manually in the ***stub resolver*** (e.g. /etc/resolv.conf for Unix stations)
  - Or discovered via DHCPv4

- In the IPv6 world: RFC4339 (IPv6 Host Configuration of DNS Server Information Approaches)
  - Via stateful DHCPv6 (RFC 3315)
  - Via stateless DHCPv6 (RFC 3736, “DHCPv6-light”) → best preferred
  - RA-based: (RFC 5006 , EXP, “IPv6 Router Advertisement Option for DNS Configuration”)
  - Well-known address (anycast or unicast)
  - Manual configuration as for IPv4
  - If IPv4 is supported, than run a DHCPv4 client

IPv6 support by Root and TLD Servers

- 13 “logical” root servers in the world:
  - *[A-M]*.root-servers.net
  - In fact: ~100 physical instances due to ***anycast*** deployment

- Since February 2008, some root-servers have been IPv6-enabled
  - A, F, H, J, K, M (‘dig @a.root-server . ns +bufsize=1200’)

- AAAA Glue records already present in the root zone for TLD delegation
  - Who puts them?
    - ICANN/IANA
  - When started?
    - 21 July 2004 with: FR, JP & KR
    - Today: more than 50 TLDs
  - How to proceed for a TLD?
    - http://www.iana.org/procedures/delegation-data.html

APIs

- **getaddrinfo()** for *forward* lookup
  - *hostname* ➔ *addresses*
  - Replacement for **gethostbyname()**
  - With AF_UNSPEC, applications become protocol-independent
- **getnameinfo()** for *reverse* lookup
  - *address* ➔ *hostname*
  - Replacement for **gethostbyaddr()**