Internet Protocol Version 6 (IPv6) Basics cheat sheet – 20130711

by Jens Roesen – email – www – twitter

IPv6 quick facts

successor of IPv4 • 128-bit long addresses • that's 2^{96} times the IPv4 address space • that's 2^{128} or $3.4x10^{38}$ or over 340 undecillion IPs overall • a customer usually gets a /64 subnet, which yields 4 billion times the IPs available by IPv4 • no need for network address translation (NAT) any more • no broadcasts any more • no ARP • stateless address configuration without DHCP • improved multicast • easy IP renumbering • minimum MTU size 1280 • mobile IPv6 • mandatory IPsec support • extension headers • jumbograms up to 4 GiB

IPv6 & ICMPv6 Headers

IPv6 header

0 | 8 | 16 | 24 | 32

version	traffic class	flow label	
payload length		next header	hop limit
source IPv6 address			
destination IPv6 address			

Version (4 bits): IP version. Always 6.
Traffic class (8 bits): Used for QoS. Like the TOS field in IPv4. RFC 2474.
Flow label (20 bits): Used for packet labelling, End-to-end QoS. RFC 6437.
Payload length (16 bits): Length of the payload following the header in bytes. Limits packet size to 64 KB.
Next header (8 bits): Code for the following extension header or UL protocol. Like protocol type field in IPv4.
Hop limit (8 bits): Number of hops until the packet gets discarded. TTL in IPv4.
Source address (128 bit): IPv6 source address.
Destination address (128 bits): IPv6 destination address.

ICMPv6 header

0 | 8 | 16 | 24 | 32

ICMPv6 type	ICMPv6 code	ICMPv6 checksum
ICMPv6 data		

ICMP type (8 bits): Error messages have a 0 high-order-bit (types 0 to 127), info messages have a 1 high-order-bit (types 128 to 255).
ICMP code (8 bits): Further specifies the kind of message along with the type. F.i. type 1 code 4 is "destination port unreachable".
ICMP checksum (16 bits): Checksum to prevent data corruption.

IPv6 Extension Headers (RFC 2460 and it's updates)

Because of the IPv6 header simplification and fixed size of 40 bytes (compared to the IPv4 header with more fields and options and 20 to 60 bytes in size) additional IP options were moved from the main IPv6 header into additional headers. These extension headers (EH) will be appended to the main header as needed. The first 8 bit of each EH identify the next header (another EH or upper layer protocol) following. Only the hop-by-hop header must be examined by every node on the path and, if present, it must be the first header following the main IPv6 header. Every EH must only occur once, only the destination options EH may occur twice - before a routing EH and before the upper layer header.

order suggested in RFC 2460

IPv6 Header	NH 0
Hop-by-Hop Options (0)	NH 60
Destination Options (60)	NH 43
Routing Header(43)	NH 44
Fragment Header(44)	NH 51
Authentication Header (51)	NH 50
ESP Header (50)	NH 60
Destination Options (60)	NH 6
TCP Header (6)	

IPv6 Addresses

/64 – lan segment, 18,446,744,073,709,551,616 v6 IPs
/48 – subscriber site, 65536 /64 lan segments
/32 – minimum allocation size, 65536 /48 subscriber sites, allocated to ISPs

`2001:0db8:0f61:a1ff:0000:0000:0000:0080`

global routing prefix (2001:0db8:0f61) – subnet ID (a1ff) – interface ID (0000:0000:0000:0080)
subnet prefix /64

IPv6 addresses are written in hexadecimal and divided into eight pairs of two byte blocks, each containing four hex digits. Addresses can be shortened by skipping leading zeros in each block. This would shorten our example address to `2001:db8:f61:a1ff:0:0:0:80`.
Additionally, once per IPv6 IP, we can replace consecutive blocks of zeros with a double colon: `2001:db8:f61:a1ff::80`.
The 64-bit interface ID can/should be in **modified EUI-64** format. A 48-bit MAC can be transformed to an 64-bit interface ID by inverting the 7th (universal) bit and inserting a `ff` and `fe` byte after the 3rd byte. So the MAC `00:03:ba:24:a9:c6` becomes `0203:baff:fe24:a9c6`. See RFC 4291 Appendix A and RFC 4941.

MAC	00	03	ba			24	a9	6c
EUI-64	02	03	ba	ff	fe	24	a9	6c

IPv6 Address Scopes

::/128	unspecified address
::1/128	localhost
fe80::/10	link local scope
fec0::/10	site local scope, intended as RFC 1918 successor, deprecated in RFC 3879
fc00::/7	unique local unicast scope, RFC 4193, divided into:
fc00::/8	centrally assigned by *unknown* (see http://bit.ly/IETFfc00), routed within a site
fd00::/8	free for all, global ID must be generated randomly, routed within a site
ff00::/8	multicast scope, after the prefix ff there are 4 bits for flags (0RPT) and 4 bits for the scope
::/96	IPv4-compatible IPv6 address, example: ::192.168.1.2, deprecated with RFC 4291
::ffff:0:0/96	IPv4-mapped IPv6 address, example: ::ffff:192.168.2.1, see RFC 4038
2000::/3	global unicast scope, divided into:
2001::/16	/32 subnets assigned to providers, they assign /48, /56 or /64 to the customer
2001:db8::/32	reserved for use in documentation
2001:678::/29	Provider Independent (PI) addresses and anycasting TLD nameservers
2002::/16	6to4 scope, 2002:c058:6301:: is the 6to4 public router anycast (RFC 3068)
3ffe::/16	6Bone scope, returned to IANA with RFC 3701, you should not see these
64:ff9b::/96	prefix used for representing IPv4 addresses in the IPv6 address space, see RFC 6052

Well Known Multicast Addresses (T-Flag = 0)

ff0X::1	all nodes address (scopes 1 and 2)
ff0X::2	all routers address (scopes 1, 2 and 5)
ff05::1:3	all site-local DHCP servers
ff02::9	all link-local RIP routers
ff02::1:ff/104	solicited-node address, the 24 low-order bits are equal to the interfaces IP 24 low-order bits
ff02::1:2	all link-local DCHP relay agents and servers
ff0X::fb	Multicast Domain Name Service v6 (all scopes)
ff0X::101	Network Time Protocol (all scopes)

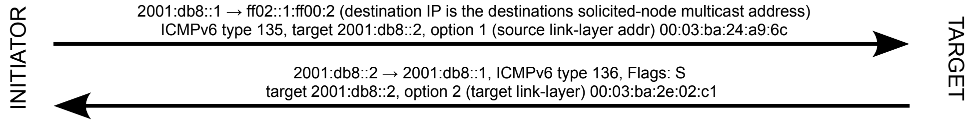
Multicast Scopes

1	Interface-local	5	Site-local
2	Link-local	8	Organization-Local
3	Admin-local	e	Global

← A "X" in the prefix is a place holder for the scope ↑

Neighbor Discovery (ND): Neighbor Solicitation (NS) and Neighbor Advertisement (NA)

Neighbor Solicitation (ICMPv6 type 135) messages are sent to determine the link-layer address of a neighbor (multicasts) or to verify that a neighbor is still reachable (unicasts).

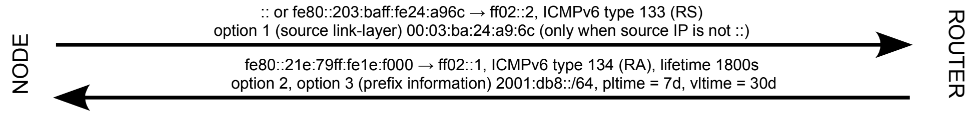


In the example above node `2001:db8::1` wants to reach `2001:db8::2` but doesn't know the link-layer address of `2001:db8::2`. So it sends a NS packet to the solicited-node multicast address of `2001:db8::2` (`ff02::1:ff00:0/104` followed by the last 24 bits of the interface ID) along with its own link-layer address and receives a NA (ICMPv6 type 136) packet with the targets link-layer address.

Duplicate Address Detection (DAD): To perform DAD the NS message is sent with the unspecified source IP `::` and to the solicited-node multicast address of the IP which should be configured. If there is already a node using this desired IP it will answer with a NA packet sent to the all-node multicast address `ff02::1`.

Neighbor Discovery (ND): Router Solicitation (RS) and Router Advertisement (RA)

Router Solicitation (RS) packets are sent in order to receive a Router Advertisement (RA) message independently from the periodically sent RAs. This is typical during stateless address autoconfiguration after successful DAD. The source IP used for the RS message can be :: or the link-local IP for this interface.



After receiving the RS message a router sends a RA message to the all-nodes multicast address. The RA message contains, amongst others, information about the router lifetime (time in seconds the router expects to be a default router), all available prefixes and their preferred (pltime) and valid (vltime) lifetimes. When pltime reaches zero the address becomes deprecated and should not be used for new connections. When the vltime reaches zero the address becomes invalid.

Stateless Address Autoconfiguration (RFC 4862) and Stateful Autoconfiguration DHCPv6 (RFC 3315)

Stateless Address Autoconfiguration (SLAAC) comes in handy when it's not important which exact address a node uses as long as it's properly routable. SLAAC uses mechanisms of Neighbor Discovery. Steps taken during SLAAC presuming there were no DAD errors along the way: forming a link-local address → DAD for the link-local address → activating the link-local address and sending RS message(s) to ff02::2 → forming a global address for each received prefix within an RA message with set "autonomous address-configuration flag" → DAD for each tentative global address → addresses become valid and preferred (for pltime > 0). See RFC 6106 for DNS configuration options advertising via RAs.

DHCPv6 can assign IPs and additional information like DNS/NTP Servers. A client sends a SOLICIT message (type 1) to the All_DHCP_Relay_Agents_and_Servers multicast IP FF02::1:2. Servers answer with a ADVERTISE message (2). The client chooses a server, sends a REQUEST message (3) and receives a REPLY message (7) with configuration options. DAD has to be performed for every address received! Alternatively, and in coexistence with SLAAC, DHCPv6 can only provide clients with additional information like DNS and NTP servers. The client sends a INFORMATION-REQUEST message (11) and receives the options in a REPLY message (7). See RFC 3315 for detailed description of DHCPv6 messages and options.

Connect to IPv6 IPs on the Command Line or in a Browser

CLI	# ssh '2001:db8:dead:f00d:203:baff:fe24:a9c6'
	# lynx http://[2001:db8:dead:f00d:203:baff:fe24:a9c6]
	# wget ftp://[2001:db8:dead:f00d:203:baff:fe24:a9c6]
Browser	http://[2001:db8:dead:f00d:203:baff:fe24:a9c6]

IPv6 and DNS (RFC 3596)

The IPv6 equivalent to the IPv4 A Resource Record is the AAAA RR. No big difference there. The A6 RR with additional fields for prefix length and prefix name defined in RFC 2874 was declared experimental in favour of AAAA RRs. See RFC 3363 and 3364 for more information and discussion.

Reverse delegation is written in nibble format with the new domain `IP6.ARPA`. So `2001:db8::2` becomes `2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.`. The shorter bitstring/bitlabel format (RFC 2672 and 2673) was later discarded but may be still used by older resolvers (BIND > 8.3 and < 9.2.3). `ipv6calc` helps with the conversion!

The `host` command will look for both A and AAAA records, using `dig` you have to explicitly ask for AAAA records (`dig host.example.com aaaa`). Reverse lookups as usual can be done with `host` without further switches (`host 2001:db8::1`) or with `dig` using the `-x` switch (`dig -x 2001:db8::1`).

Linux IPv6 Interface Configuration examples (Steps might slightly differ between distributions)

Manual configuration: You can temporarily configure an IPv6 address with the `ifconfig` or `ip` command:

```
# ifconfig eth0 inet6 add 2001:db8::2/64        or
# ip addr add 2001:db8::2/64 dev eth0
```

Add a default route

```
# route -A inet6 add default 2001:db8::1        or
# ip -6 route add default via 2001:db8::1
```

To check the configuration use `ifconfig eth0` or `ip -6 addr show eth0` respectively `route -A inet6` or `ip route show`. For making the changes permanent you'll have to put them in the appropriate config files.

Automatic configuration using SLAAC: Just having IPv6 enabled and IPv4 configured on the interface should normally do the trick.

SLAAC with privacy extensions: To deal with security and privacy concerns regarding EUI-64 interface IDs enable and prefer temporary addresses over other public addresses with:

```
# sudo sysctl net.ipv6.conf.eth0.use_tempaddr = 2
# sudo sysctl net.ipv6.conf.default.use_tempaddr=2
```

To make these settings boot proof put them into `/etc/sysctl.conf`. Change valid and preferred lifetime of temporary addresses by editing `temp_valid_lft` and `temp_prefered_lft` values (defaults are 604800 (7d) and 86400 (1d) seconds) for the interface.

*NIX IPv6 Console Tools

`ping6`	IPv6 version of ping. Solaris ping supports IPv6 out of the box.
`traceroute6` `tracepath6`	IPv6 versions of traceroute and tracepath. Also try `mtr -6`.
`ip -6`	Configure or view interfaces, routes, ND, list neighbors, multicasts.... on linux
`ipv6calc`	Powerful tool for all sorts of conversions and information gathering. See `http://www.deepspace6.net/projects/ipv6calc.html`
`tcpdump ip6` `snoop inet6`	Packet sniffing tools with IPv6 options. Also works with options like `icmp6`.

IPv6 RFCs (available at http://tools.ietf.org/html/rfc<*RFC number*>)

RFC 2460	IPv6 Specifications	RFC 4193	Unique Local IPv6 Unicast Addresses
RFC 4291	IPv6 Addressing Architectures	RFC 2375	IPv6 Multicast Address Assignments
RFC 4861	IPv6 Neighbor Discovery	RFC 3849	IPv6 Address Prefix For Documentation
RFC 4862	IPv6 Stateless Address Configuration	RFC 4941	Privacy Extensions for SLAAC in IPv6
RFC 1981	Path MTU Discovery for IPv6	RFC 6147	DNS64 – DNS Extensions for NAT64
RFC 3596	DNS Extensions to Support IP Version 6	RFC 6146	Stateful NAT64
RFC 4443	ICMPv6 for IPv6	RFC 6434	IPv6 Node Requirements
RFC 3587	IPv6 Global Unicast Address Format	RFC 6540	IPv6 Support Required for All IP-Capable Nodes