



DNS and DHCP in Dual Stack Networks

Lawrence E. Hughes

Chairman, InfoWeapons Inc.

Your Speaker

- 35+ years in IT field, including:
 - 2 years with VeriSign (PKI, crypto)
 - Co-founder CipherTrust (e-mail security)
 - Founder InfoWeapons (IPv6 Infrastructure)
- I've personally invested US\$5M+ in IPv6
- Built team with large number of IPv6 and security experts (development, testing, Q/A, etc.)
- Member of US IPv6 Consortium
- My company (InfoWeapons) has been running production dual stack in-house networks for 4 years (mostly using our own designs and products)
- Many future IPv6 products planned (VoIP, translator, etc.)



What is DNS?

- DNS = Domain Name System (RFC1034/5 with updates)
- Invented by Paul Mockapetris, November 1987
- Basically maps Fully Qualified Domain Names (FQDNS) to and from numeric IP addresses





What is DNS?

- Also publishes name of key servers for a given domain (MX record for e-mail, SRV record for LDAP, SIP, etc.)
- Implemented as a globally distributed database and database engine, managed by thousands of admins
- The Internet today could not exist without it
- Even more critical for IPv6 networks than IPv4 (ever try typing in 32 hex digit numeric address in a browser?)



How is DNS used?

- Every node on the Internet that can use FQDNs (e.g. www.xyz.com) must have a DNS client (“resolver”) that can query a DNS server to translate any node’s FQDN into a numeric IP address (which is what is actually used as destination in packets)
- Implements tree-like hierarchical name space
- If the DNS server you query doesn’t have the answer (is “authoritative” for that domain, or has “cached” that data from a previous query), that server can either point you to the right server (referral), or obtain the data on your behalf (recursive)
- No single database could accomplish this, or keep data current. Only a distributed database could hope to perform this function. There are parts of this database scattered all over the Internet.

DNS and IPv6

- 'A' records map FQDN to IPv4 address, 'AAAA' for IPv6
- 'A6' records now deprecated, not needed
- PTR records map IPv4 and IPv6 address to FQDN

AAAA record fields

NAME	Domain name
TYPE	AAAA (28)
CLASS	Internet (1)
TTL	Time to live in seconds
RDLENGTH	Length of RDATA field
RDATA	String form of the IPV6 address as described in RFC 3513

DNS and IPv6

- Client->server (query) and server->server (zone transfer) connections can be over IPv4 or IPv6
- This has been in BIND since version 9 (good since 9.3.2)
- In a dual stack appliance, web management, NTP, SNMP and all other protocols should also be dual stack
- A given node may have zero or more IPv4 and zero or more IPv6 addresses defined (but at least one address of some kind)
- Clients should be ready to accept multiple addresses for a given node query, and if the client should try any returned IPv6 addresses first, then fall back to IPv4

DNS Root Server Issues with IPv6

- Root servers are the top of the global DNS pyramid – there are “13” around the world
- As of 2/4/2008 at least six of them support IPv6 for pure IPv6 DNS queries to work
- Currently the K root server is measuring 100 IPv6 queries per second (IPv4 typical 9000/sec)
- Note that due to longer IPv6 addresses, a single 512 byte UDP packet will no longer hold 13 root server addresses
 - One solution is EDNS0 (RFC 2671)
 - Can also fail over to TCP (may cause FW issues)
 - *Names* of all 13 root servers can fit with no problem

What is DHCP?

- DHCP = Dynamic Host Configuration Protocol (RFC 2131, plus updates)
- A way for clients to obtain an IP address and other network info (DNS server address, gateway, netmask, etc.) at power-up time
- DHCPv4 depends on broadcast, can only supply IPv4 data
- Addresses can be assigned first come first servers from an address pool
- Often used to serve more users than you have valid IPv4 addresses (not simultaneously) as “dynamic addresses”
- Can assign fixed addresses to specific nodes (e.g. servers) keyed to their MAC address





What is DHCPv6?

- Replacement for DHCPv4 that supports IPv6 – RFC 3315 (with updates)
 - Does not use broadcast (uses multicast)
 - Can supply IPv6 address info
 - “Stateless” and “Stateful” operating modes
 - Relay agents not needed in each subnet



Stateless DHCPv6

- Works in conjunction with IPv6 stateless autoconfiguration (which supplies IP addresses to client nodes)
 - Client gets prefix from Router Advertisement Daemon
 - Low 64 bits typically derived from client MAC address
 - No way to “cluster” addresses by group (e.g. acctng, devel, support) which complicates FW rules
 - Limited number of MAC addresses make scans possible
- DHCPv6 server only supplies “stateless” data (things that are the same for every node), such as DNS server addresses, gateway address, etc.
- Currently DHCPv6 is the only way for a client to learn IPv6 addresses for DNS servers



Stateful DHCPv6

- Client nodes get stateless data *and* a unique IPv6 address from the DHCPv6 server
 - Possible to assign address keyed to MAC address, so clustering of addresses by group is possible, vastly simplifying firewall rules, Network Access Control
 - Possible to generate low 64 bits randomly, which makes subnet scans impossible
 - Can be integrated with DNS to publish assigned addresses
- Will probably be used in most medium or large networks (>50 nodes), versus stateless autoconfig and DHCP
- Many early DHCPv6 servers don't yet support stateful operation (IW SolidDNS™ 2.0 does)



Client Support for DHCPv6

- Microsoft Vista includes client support
- Clients available for Windows XP (SolidDNS™ includes GUI DHCPv6 client)
- Most *NIX derivatives have clients available (FreeBSD, Linux, Solaris, etc.)



Automated Network Renumbering

- Requires integrated dual stack DNS/DHCPv6 server
- Possible to renumber entire networks periodically (once a day?), transparently to rest of world (so long as DNS is used)
- Depends on clever use of “deprecated but still functional” attribute of IPv6 addresses and DNS “Time to Live” values
- If low 64-bit generated randomly, can lead to very strong security (resistance to scanning, hackers have to rediscover nodes constantly, etc.)
- Planned for future release of SolidDNS™



END