SWINBURNE UNIVERSITY OF TECHNOLOGY

IPv6 @CAIA and @Home

Mattia Rossi

mrossi@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

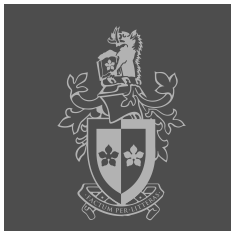

Table of contents

IPv6 vs. IPv4 – Important Differences

- Important differences
  - Address length (address space)
    - IPv6: 128 bit – IPv4: 32 bit
  - Notation
    - IPv6: Hex with colons – IPv4: Dotted decimal
  - Fragmentation
    - IPv6: no fragmentation – IPv4: Fragmentation
    - Affects MTU

IPv6 vs. IPv4 – Important Differences cont.

- Address scope
  - IPv6: Scopes by design – IPv4: Scopes created on the way
- Broadcast and Multicast
  - IPv6: No Broadcast, only Multicast – IPv4: Broadcast and Multicast
- Dynamic Host configuration
  - IPv6: Multiple methods – IPv4: DHCP
- Prefixes vs. Addresses
  - IPv6: Prefixes and Addresses – IPv4: Addresses

IPv6 numbers

- IPv6 address space:
  - 2^{128} addresses = $2^{32} * 2^{32} * 2^{32} * 2^{32}$ addresses
  - IPv4 Internet * IPv4 Internet * IPv4 Internet *IPv4 Internet
  - (IPv4 Internet)^4
- IPv6 prefix space:
  - 2^{64} /64 prefixes (subnets) = $2^{32} * 2^{32}$ /64 prefixes (subnets)
  - IPv4 Internet * IPv4 Internet = (IPv4 Internet)^2 /64 prefixes

IPv6 numbers cont.

- IPv6 prefix space detail:
  - 2^{48} /48 prefixes = $2^{32} * 65536$ /48 prefixes
  - IPv4 Internet * 65536 /48 prefixes = 65536 /48 in one /32
  - Swinburne has a /48 prefix, AARNet has a /32, APNIC a /23
  - Swinburne: 65536 /64 subnets or 4096 /60 or 256 /56
- Plenty of prefixes/subnets available and even more addresses

IPv6 addresses and notation

- IPv6 addresses in hexadecimal notation:
  - 2001:388:e000:a00::1
  - 2001:388:e000:a00:20f:feff:fe8c:ab5 — global unicast
  - fe80::20f:feff:fe8c:ab5%em0 — link-local unicast
  - ff02::c
  - ff02::1 — link-local multicast
  - fdfe:dcba:9876:: — unique local address - ULA
  - ::1 — Loopback address
  - :: — Unspecified address

IPv6 addresses and notation

- Different notations for the same address:
  - 2001:388:e000:a00::1
    - 2001:0388:e000:0a00:0000:0000:0000:0001
      collapse sequence of zeros to ::
    - 2001:0388:e000:0a00::0001
      strip leading zeros
    - 2001:388:e000:a00::1

IPv6 addresses and notation

- Basic rules for the notation
  - ❑ Only one sequence of zeros can be reduced to ::
  - ❑ Always reduce the longer sequence to ::
  - ❑ A single 16 bit field of zeros can not be reduced to :: but must be written as 0
  - ❑ Striping leading zeros is allowed, striping trailing zeros is not!
  - ❑ Rules apply to address and prefix notation
  - ❑ RFC 5952

Construction of an IPv6 address

- IPv6 address consists of LOC (locator) and ID (identifier)
  - ❑ 64 bits LOC (network ID) and 64 bits ID (interface ID)
  - ❑ maximum length of global routable prefix is 64 bit (/64)
  - ❑ ID can be created manually or can be generated
    - ❑ from MAC address, EUI-64, CGA, pseudo-random ID
    - ❑ generated ID is always 64 bit!!
  - ❑ 2001:388:e000:a00:20f:feff:fe8c:ab5

    LOC ID
  - ❑ LOC for ULAs can be generated using an algorithm

IPv6 prefix notation

- IPv6 prefix notation examples
  - 2001:388:e000:a00::/64
  - 2001:388:e000:a00::/56

  } global unicast
  - fe80::/10 } link-local unicast
  - ff00::0/12 } multicast
  - fc00::/7 } ULA
  - 2001:db8::/32 } documentation range

- More information in RFC 4291 and IANA IPv6 registry

IPv6 subnet calculation

- /64 from documentation range as example
- 2001:db8::/32 - 2001:db8::/64

documentation range 1st /64 subnet

  - 2001:db8::/64 ⟶ 2001:db8:0:0::/64
  - or 2001:0db8:0000:0000::/64

/8 /16 /24 /32 /40 /48 /56 /64

  - Trailing zeros can't be omitted!
  - 2001:db8:0:a00::/64 – 2001:db8:0:a00::/56

IPv6 subnet calculation cont.

- Splitting up 2001:db8:0:a00::/60 [1]

  - P:a00::/64, P:a00::/63, P:a00::/62, P:a00::/61
  - P:a01::/64,
  - P:a02::/64, P:a02::/63
  - P:a03::/64
  - P:a04::/64, P:a04::/63, P:a04::/62
  - P:a05::/64
  - P:a06::/64, P:a06::/63
  - P:a07::/64
  - P:a08::/64, P:a08::/63, P:a08::/62, P:a08::/61

[1] IPv6 Prefix Primer – IPv6 Now Pty Ltd.

Getting IPv6

- Multiple ways to get IPv6
  - Native / DHCPv6 Prefix Delegation (PD) (RFC 3633)
    - Clients for FreeBSD, Linux, Windows XP, MacOSX
  - IPv6 in IPv4 tunnel (gateway6 - RFC 3053)
    - Clients for FreeBSD, Linux, Windows, MacOSX
  - DS-Lite (Cisco, Linksys – RFC 6333)
    - Clients for Linux
  - Some CPE (Customer premises equipment) support some of the methods

Getting IPv6

- Getting IPv6 means:
  - getting an IPv6 prefix assigned
  - getting an IPv6 default route
  - getting an IPv6 enabled name server
- Commonly IPv6 is an addition to the existing IPv4 network: IPv6/IPv4 Dual-Stack
- Depending on the method of delivery, IPv6 enabled name servers are not available:
  - IPv4 only DNS servers can resolve AAAA as well
  - Problem: IPv6 only networks and hosts are left out

Getting IPv6 @ CAIA

- AARNet runs IPv6 tunnel broker (RFC3053)
  - Need to register at http://broker.aarnet.net.au
- IPv6/IPv4 tunnel using ***gateway6*** client (gogo6.net) running on a FreeBSD 8.2-RELEASE host
- /56 prefix from AARNet (2001:388:e000:a00::/56)
- Addresses from a /64 prefix (2001:388:e000:a00::/64) are distributed on the 229 VLAN using SLAAC
  - **S**tate**L**ess **A**ddress **A**uto**C**onfiguration (RFC 4862)
  - Every IPv6 enabled client on the 229 VLAN will get an IPv6 address

Gateway6 configuration

- ***net/gateway6*** port on FreeBSD:
  - Installs in /usr/local
  - Config file: /usr/local/etc/gw6c.conf
  - Only a few changes to the default config necessary:

```
userid=<your_userid>
passwd=<your_password>
server=broker.aarnet.net.au
host_type=router
prefixlen=56
if_prefix=em0
tunnel_mode=v6udpv4
```


Running gateway6

- Executing gateway6 on FreeBSD:
  - /etc/rc.d/gateway6 start
  - /etc/rc.conf entry for autostart:

```
gateway6_enable="YES“
```

  - If host_type is set to router, gateway6 automatically runs SLAAC on if_prefix interface, sending router advertisements (RA)
  - Gateway6 sets up an address (PREFIX::1) on the if_prefix interface - gateway address
  - Depending on the tunnel_mode gateway6 creates a gif or tun interface to set up a tunnel to the broker

ifconfig with running gateway6 client

- ifconfig command on FreeBSD:

```
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:8c:0d:2a
        inet 136.186.229.112 netmask 0xffffff00 broadcast 136.186.229.255
        inet6 fe80::a00:27ff:fe8c:d2a%em0 prefixlen 64 scopeid 0x1
        inet6 2001:388:e000:a00::1 prefixlen 64
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
ipfw0: flags=8801<UP,SIMPLEX,MULTICAST> metric 0 mtu 65536
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1280
        options=80000<LINKSTATE>
        inet6 fe80::a00:27ff:fe8c:d2a%tun0 prefixlen 64 scopeid 0x4
        inet6 2001:388:f000::a79 --> 2001:388:f000::a78 prefixlen 128
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        Opened by PID 843
```


sysctl changes on IPv6 router

- sysctl settings are changed if host_type=router:
  - Relevant FreeBSD sysctl variables:

```
net.inet6.ip6.forwarding: 1
net.inet6.ip6.redirect: 1
net.inet6.ip6.accept_rtadv: 0
```

  - A router configured for SLAAC can not accept router advertisements itself

SLAAC

- SLAAC daemon on FreeBSD systems is called ***rtadvd***
- Installed as part of the base system
- Provides /64 prefix (or smaller if configured so) and a route to the host (RA)
- Host automatically generates an address using the prefix (all modern OS support SLAAC)
- Address conflicts are resolved using DAD (duplicate address detection
- Makes use of NDP (neighbor discovery protocol)

Neighbor Discovery Protocol (NDP)

- NDP protocol (RFC 4861):
  - ❑ Improved ARP
    - ❑ Works on all multicast capable links
    - ❑ Allows Neighbor Unreachability Detection (NUD)
    - ❑ Allows Duplicate Address Detection (DAD)
    - ❑ Allows parameter discovery, like MTU
    - ❑ ...
  - ❑ uses ICMP for communication, ICMP must be allowed on the network!
  - ❑ Uses link-local addresses and multicast addresses
    - ❑ link-local addresses are generated locally from MAC or EUI-64

NDP example output

- Control/troubleshoot utility on FreeBSD – ***ndp*** [2]***:***

```
# ndp  -an
Neighbor                                Linklayer Address   Netif Expire      S Flags
2001:388:e000:a00:8880:1784:8bc3:56f    0:2:b3:eb:27:d1     fxp0 permanent    R
2001:388:e000:a00:202:b3ff:feeb:27d1    0:2:b3:eb:27:d1     fxp0 permanent    R
fe80::20f:feff:fe8c:ab5%fxp0            0:2:b3:eb:27:d1     fxp0
fe80::a00:27ff:fe8c:d2a%fxp0            8:0:27:8c:d:2a      fxp0 1h32m59s     S R
2001:388:e000:a00::                     0:f:fe:8c:a:b5      em0 permanent     R
2001:388:e000:a00::1                    8:0:27:8c:d:2a      em0 20s           R R
2001:388:e000:a00:54e0:3a31:877f:80de 0:f:fe:8c:a:b5        em0 permanent     R
2001:388:e000:a00:2951:55b0:7562:9d0c 0:f:fe:8c:a:b5        em0 permanent     R
2001:388:e000:a00:bc42:ceed:57a2:f4e6   0:f:fe:8c:a:b5      em0 permanent     R
2001:388:e000:a00:f9ac:dbcf:fcd7:7dff   0:f:fe:8c:a:b5      em0 permanent     R
2001:388:e000:a00:c03f:ca6:cf27:86e3    0:f:fe:8c:a:b5      em0 permanent     R
2001:388:e000:a00:20f:feff:fe8c:ab5     0:f:fe:8c:a:b5      em0 permanent     R
fe80::20f:feff:fe8c:ab5%em0             0:f:fe:8c:a:b5      em0 permanent     R
2001:388:e000:a00:7870:c6f3:2881:372c 0:f:fe:8c:a:b5        em0 permanent     R
2001:388:e000:a00:8dec:2ef:ff00:8925    0:f:fe:8c:a:b5      em0 permanent     R
fe80::a00:27ff:fe8c:d2a%em0             8:0:27:8c:d:2a      em0 15s           R R
```

[2] ndp (8)

SLAAC cont.

NDP

- A Router runs ***rtadvd*** to send periodic router advertisements (RA)
- A host runs ***rtsold*** [3] to send periodic router solicitations (RS)
- On FreeBSD router solicitations can be forced from a client using ***rtsol <if>***
- ***rtsol*** – part of the base system (world)
- FreeBSD client needs to be enabled to accept router advertisements:
  - ❑FreeBSD 7 and 8 - /etc/rc.conf: ipv6_enable="YES“
  - ❑FreeBSD 9 - /etc/rc.conf [4]:

    ifconfig_em0_ipv6="inet6 accept_rtadv –ifdisabled defaultif“

[3] rtsold (8)
[4] ifconfig (8)

SLAAC cont.

- rtadvd configured/executed by gateway6 platform specific script
- Does not necessary need configuration, calculates the /64 prefix(es) from the address(es) configured on an interface
- If rtadvd needs to run on multiple interfaces a configuration file is needed
- Invoke as ***rtadvd –c configfile <if>***
- more info: rtadvd (8)

Interfacing gateway6 and DHCPv6

- DHCPv6 possibly wanted to create multiple subnets
- Needs manual intervention (not used at CAIA)
- On FreeBSD install ports ***net/isc-dhcp42-server*** or ***net/dhcp6*** (don't use net-mgmt/wide-dhcp, it's broken)
- Run gateway6 in host mode (not router)
- Configure DHCPv6-PD (Prefix delegation) based on prefix assigned by the broker (RFC 3633)
- Needs manual intervention if renumbering occurs
- Broker usually does not renumber

Configuring isc-dhcpd42 server for PD

- ***net/isc-dhcp42-server*** installs into /usr/local/sbin/dhcpd (v4 and v6 version)
- Configuration file for DHCPv6: /usr/local/etc/dhcpd6.conf
- Execute as /usr/local/sbin/dhcpd -6 -cf /usr/local/etc/dhcpd6.conf
- or /usr/local/etc/rc.d/isc-dhcpd6 start and in rc.conf: dhcpd6_enable="YES"

Configuring isc-dhcpd42 server for PD

- /usr/local/etc/dhcpd6.conf [5]:

```
options dhcp6.name-servers 2001:db8::ca11;
options dhcp6.domain-search "caia.swin.edu.au","swin.edu.au";
subnet6 2001:db8::/56 {
   prefix6 2001:db8:0:10:: 2001:db8:0:f0:: /60;
}
```

[5] dhcpd.conf(5), dhcp-options(5)

Configuring isc-dhcpd42 client for PD

- Client at /usr/local/sbin/dhclient
- In /etc/rc.conf :

```
dhclient_program= "/usr/local/sbin/dhclient"
dhclient_flags= "-cf /usr/local/etc/dhclient.conf "
```

- /usr/local/etc/dhclient.conf [6]:

```
interface "em0" {
    request dhcp6.ia-pd;
    also request dhcp6.name-servers, dhcp6.domain-search;
}
```

[6] dhclient.conf(5): man –M /usr/local/man dhclient.conf

What to do with a PD

- Delegated prefix can be /64 or smaller (if globally routable)
- ISC dhclient does not assign addresses or subprefixes to an interface
- Additional scripting needed – dhclient-script, dhclient-enter-hooks
- Additional information can be passed using –e flag (interface information, rtadvd path etc.)
- dhclient-enter-hooks is shell script, can execute rtadvd.
- Example available @ CAIA

What to do with a PD – KAME/WIDE dhcp

- KAME/WIDE dhcp supports assignment of subprefix/address to interface
- Multiple prefixes/interfaces supported
- simple configuration file directive
- Suggested for native IPv6 @ Home from Internode (only known provider to offer native IPv6 atm.)
- rtadvd can then be used to send router advertisements to other hosts in the subnet

What to do with a PD – KAME/WIDE dhcp

- WIDE dhclient.conf [7]:

```
interface ppp0 {
    send ia-pd 0;
    script "/etc/wide-dhcpv6/dhcp6c-script";
};
id-assoc pd {
    prefix-interface eth0 {
        sla-id 0;
        sla-len 8;
    };
};
```

sla-id 0; ← Site level aggregator ID – start of the new prefix

sla-len 8; ← Site level aggregator length:
delegated prefixlen + this = new prefixlen
In this case 56 + 8 = 64

[7] http://ipv6.internode.on.net/configuration/adsl-linux-bsd/

Additional DHCPv6 clients

- DHCPv6 – PD capable clients only needed on routers
- DHCPv6 clients for name servers etc. Needed on hosts
- Other PD capable clients: dibbler, RedHat DHCP
  - Dibbler runs on Windows XP and Linux, included in OpenWRT
  - RedHat dhcp is similar to wide-dhcp, runs on Linux
- DHCPv6 clients without PD capability:
  - MacOSX Lion, Windows 7, Linux, FreeBSD

DHCPv6 vs SLAAC

- Why DHCPv6 if we have SLAAC?
- SLAAC only for IPv6 addresses and routes
  - also name servers (RDNSS) and domain search lists (DNSSL) (RFC 6106)
- DHCPv6 for prefixes, addresses, name servers, ntp servers, SIP servers etc.
- SLAAC and DHCPv6 play nicely together
  - M flag: obtain addresses from DHCPv6 server
  - O flag: obtain additional (other) information from DHCPv6 server, like name server
- Windows 7 and MacOSX Lion require both!

Name servers

- If DHCPv6 is not available, how do get nameservers?
- IPv4/IPv6 dual stack: use IPv4 nameserver
- IPv6 only scenario (1): Manual setup of DNS server
  - Windows 7, MacOSX Lion
- IPv6 only scenario (2): (RFC 6106)
  - Shipped with FreeBSD 9, patch for 8 available – backport may be available soon, radns available for Linux [7]
  - Needs to be configured in rtadvd.conf
  - Needs configuration on the client

[7] http://hack.org/c/hacks/radns/

Name servers

- rtadvd.conf example [8]:

```
em0:\
        :addr="2001:db8:ffff:1000::":prefixlen#64:\
        :rdnss="2001:db8:ffff::10,2001:db8:ffff::2:43":\
        :dnssl="foo.com":\
        :raflags="mo"
```

prefix

name servers

domain search list

M and O flags, M obsoletes O, Forces address configuration from DHCPv6

[8] rtadvd.conf (5)

Name servers

- FreeBSD 9 clients use resolvconf tool (new!)
- Configuration file: /etc/resolvconf.conf
- Example:

```
# cat /etc/resolvconf.conf
name_servers_append="136.186.20.9 136.186.1.111 136.186.1.115"
```

- Should interact properly with DHCPv4, DHCPv6 and RDNSS
- Can take care of VPN DNS, and keep per interface domain search lists

Name servers

- Which name server to use?
- ISP, tunnel provider - but what if it's not known?
  - ❑ Run your own!
- Bind is part of FreeBSD and MacOSX, can be easily installed on Linux
- Simple to set up:
  - named.conf: listen-on-v6 { 2001:db8::1; ::1; };
  - rc.conf : named_enable="YES" and /etc/rc.d/named start

Name servers

- CAIA IPv6 nameserver: 2001:388:e000:a00::1
- IPv6 only, does not listen to IPv4
  - bind can run in dual-stack mode
- Advertised using RDNSS
- Uses AARNet forwarders (Google IPv6 whitelist)

```
# dig AAAA www.google.com
;; ANSWER SECTION:
www.google.com.      48915   IN     CNAME   www.l.google.com.
www.l.google.com.    145     IN     AAAA    2001:4860:4001:803::1012
```


Hostnames

- IPv6 address are hard (impossible?) to remember, names are strongly desired, for ssh access etc.
- Simple solution: deploy IPv6 capable DNS server in the network, and add all hosts manually
  - SLAAC configured hosts always generate the same address
- Possible for small home networks, impossible for larger networks
- Not dynamic

Hostnames

- Again DHCPv6 vs. SLAAC
- DHCPv6 interfaces well with DNS (bind)
  - Requires addresses to be distributed via DHCP
- Possible solution in the far future :
  - draft-jiang-dhc-addr-registration-03
  - Registering SLAAC addresses at DHCPv6 server which then registers names to DNS
- SLAAC and DNS do not cooperate yet

Dynamic hostname resolution via mDNS

- Other dynamic solution: multicast Domain Name Service (mDNS)
  - Aka: bonjour, zeroconf, avahi
  - On the way to become an IETF standard, currently draft-cheshire-dnsext-multicastdns-14
- Available on MacOSX, FreeBSD and Linux
- Not available on Windows
  - Uses Link Local Multicast Name Resolution (LLMNR) (RFC4795 – Informational, after 47 draft revisions)

Dynamic hostname resolution via mDNS

- mDNS resolves names and services via UDP multicast messages
  - Works on IPv4 and IPv6, uses port 5353
  - Uses IPv6 multicast address ff02::fb
  - Uses .local domain
- DNS-SD (Service Discovery) provides information about services provided by a host
  - ssh, www, multimedia etc.
  - draft-cheshire-dnsext-dns-sd-10
  - http://www.dns-sd.org/

Dynamic hostname resolution via mDNS

- mDNS on FreeBSD or Linux via ***mDNSResponder*** (Apple) or ***Avahi*** (avahi.org)
- mDNSResponder provides ***mdnsd*** and ***dnsextd*** which need no configuration
  - mdnsd is the mDNS daemon, which advertises services and names
  - dnsextd is supposed to exchange mDNS information with DNS (bind), enabling DNSSEC style security, but the POSIX platform specific files are incomplete!

Dynamic hostname resolution via mDNS

- Avahi provides ***avahi-daemon*** and ***avahi-dnsconfd***
  - avahi-daemon provides the same services as mdnsd.
    - Configuration file avahi-daemon.conf
    - Allows flexibility
  - avahi-dnsconfd allows distribution of name servers (if DHCPv6 or RDNSS are not available)
    - Shell script avahi-dnsconfd.action
    - A bit buggy under FreeBSD, but it's shell script, can be fixed easily
- Avahi is in FreeBSD ports: net/avahi
  - Can pull in X related stuff

Dynamic hostname resolution via MDNS

- avahi-daemon sample config for IPv6 only:

```
[server]
host-name=mrossi2
domain-name=local
browse-domains=0pointer.de, zeroconf.org, caia.swin.edu.au, swin.edu.au
use-ipv4=no
use-ipv6=yes
allow-interfaces=em0, lo0
deny-interfaces=tun0, gif0
#check-response-ttl=no
#use-iff-running=no
enable-dbus=yes
#disallow-other-stacks=no
allow-point-to-point=yes
#cache-entries-max=4096
#clients-max=4096
#objects-per-client-max=1024
#entries-per-entry-group-max=32
ratelimit-interval-usec=1000000
ratelimit-burst=1000
```

Dynamic hostname resolution via MDNS

- avahi-daemon sample config for IPv6 only cont.:

```
[wide-area]
enable-wide-area=yes

[publish]
#disable-publishing=no
#disable-user-service-publishing=no
#add-service-cookie=no
#publish-addresses=yes
#publish-hinfo=yes
#publish-workstation=yes
publish-domain=yes
publish-dns-servers=2001:388:e000:a00::1, 136.186.20.9, 136.186.1.111, 136.186.1.115
publish-resolv-conf-dns-servers=no
publish-aaaa-on-ipv4=no
publish-a-on-ipv6=no
```

- Other domain than .local can be used – using the same domain as DNS causes unpredictable results though

Dynamic hostname resolution via MDNS

- Avahi in action:

```
# avahi-browse -a
+   em0 IPv6 sting1                                  _ntp._udp          local
+   em0 IPv6 mrossi3                                 _sftp-ssh._tcp     local
+   em0 IPv6 sting1                                  _sftp-ssh._tcp     local
+   em0 IPv6 mrossi                                  _sftp-ssh._tcp     local
+   em0 IPv6 mrossi2                                 _sftp-ssh._tcp     local
+   em0 IPv6 SFTP File Transfer on bgp1                    _sftp-ssh._tcp     local
+   em0 IPv6 mrossi3                                 _ssh._tcp          local
+   em0 IPv6 sting1                                  _ssh._tcp          local
+   em0 IPv6 mrossi                                  _ssh._tcp          local
+   em0 IPv6 mrossi2                                 _ssh._tcp          local
+   em0 IPv6 bgp1                                    _ssh._tcp          local
+   em0 IPv6 abolfazl's remote desktop on desktop          _rfb._tcp          local
+   em0 IPv6 desktop                                 _udisks-ssh._tcp   local
+   em0 IPv6 kstoeckigt-desktop                           _udisks-ssh._tcp   local
+   em0 IPv6 mrossi@mrossi                                _presence._tcp     local
```

Dynamic hostname resolution via MDNS

- Avahi in action cont.:

```
+   em0 IPv6 kstoeckigt@kstoeckigt-desktop            _presence._tcp      local
+   em0 IPv6 mrossi3 [78:e7:d1:cc:29:25]              _workstation._tcp   local
+   em0 IPv6 localhost [00:30:48:fd:74:3b]            _workstation._tcp   local
+   em0 IPv6 sting1 [00:24:81:1f:b6:d7]               _workstation._tcp   local
+   em0 IPv6 mrossi [00:0f:fe:8c:0a:b5]               _workstation._tcp   local
+   em0 IPv6 mrossi2 [08:00:27:8c:0d:2a]              _workstation._tcp   local
+   em0 IPv6 bgp1 [00:1d:7d:d1:d3:05]                 _workstation._tcp   local
+   em0 IPv6 desktop [00:0f:fe:d7:a4:1d]              _workstation._tcp   local
+   em0 IPv6 kstoeckigt-desktop [00:0f:fe:a2:52:fd]   _workstation._tcp   local
+   em0 IPv6 HP LaserJet P2055dn [493A66]             _telnet._tcp        local
+   em0 IPv6 HP LaserJet P2055dn [493A66]             _http._tcp          local
+   em0 IPv6 abolfazl's remote desktop on desktop     _http._tcp          local
+   em0 IPv6 HP LaserJet P2055dn [493A66]             _pdl-datastream._tcp local
+   em0 IPv6 HP LaserJet P2055dn [493A66]             _printer._tcp       local
```


Dynamic hostname resolution via mDNS

- Allow glibc to resolve hostnames via mDNS:
- nss_mdns:
  - FreeBSD port dns/nss_mdns
- /etc/nsswitch used for resolution order
- /etc/nsswitch mDSN enabled:

  hosts: files mdsn_minimal [NOTFOUND=return] dns mdns
- Problem: mDNS does not tell when it's finished searching
  - No error like DNS - timeout is used

Dynamic hostname resolution via mDNS

- nss_mdns in action:

```
# ping6 sting1.local
PING6(56=40+8+8 bytes) 2001:388:e000:a00:8dec:2ef:ff00:8925 -->
    2001:388:e000:a00:224:81ff:fe1f:b6d7%12683064
16 bytes from 2001:388:e000:a00:224:81ff:fe1f:b6d7, icmp_seq=0 hlim=64 time=9.097 ms
16 bytes from 2001:388:e000:a00:224:81ff:fe1f:b6d7, icmp_seq=1 hlim=64 time=0.347 ms
```

- and:

```
# ndp -a
Neighbor                Linklayer Address  Netif Expire    S Flags
sting1.local            0:24:81:1f:b6:d7   em0 23h58m29s S
mrossi.local            0:f:fe:8c:a:b5     em0 permanent R
mrossi2.local           8:0:27:8c:d:2a     em0 23h59m16s S R
mrossi.local            0:f:fe:8c:a:b5     em0 permanent R
mrossi.local            0:f:fe:8c:a:b5     em0 permanent R
mrossi.local            0:f:fe:8c:a:b5     em0 permanent R
mrossi.local            0:f:fe:8c:a:b5     em0 permanent R
```


IPv6 Privacy extensions

- What's with all this *mrossi.local* occurrences?
- IPv6 privacy extensions (RFC 4941):
  - Make it harder to trace hosts
  - Keep assigning a new IPv6 address on temporary basis
  - Controlled by two sysctls (FreeBSD):
    - net.inet6.ip6.use_tempaddr: Switch privacy extensions on
    - net.inet6.ip6.prefer_tempaddr: Prefer the temporary address when using
  - For manual DNS update: not a problem, original autoconfigured address is retained

IPv6 Privacy extensions

- Privacy extensions in action:

```
ifconfig em0
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=219b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,TSO4,WOL_MAGI>
        ether 00:0f:fe:8c:0a:b5
        inet 136.186.229.109 netmask 0xffffff00 broadcast 136.186.229.255
        inet6 fe80::20f:feff:fe8c:ab5%em0 prefixlen 64 scopeid 0x1
        inet6 2001:388:e000:a00:20f:feff:fe8c:ab5 prefixlen 64 autoconf
        inet6 2001:388:e000:a00:f9ac:dbcf:fcd7:7dff prefixlen 64 deprecated autoconf temporary
        inet6 2001:388:e000:a00:c03f:ca6:cf27:86e3 prefixlen 64 deprecated autoconf temporary
        inet6 2001:388:e000:a00:bc42:ceed:57a2:f4e6 prefixlen 64 deprecated autoconf temporary
        inet6 2001:388:e000:a00:7870:c6f3:2881:372c prefixlen 64 deprecated autoconf temporary
        inet6 2001:388:e000:a00:: prefixlen 64
        inet6 2001:388:e000:a00:2951:55b0:7562:9d0c prefixlen 64 deprecated autoconf temporary
        inet6 2001:388:e000:a00:54e0:3a31:877f:80de prefixlen 64 deprecated autoconf temporary
        inet6 2001:388:e000:a00:8dec:2ef:ff00:8925 prefixlen 64 autoconf temporary
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
```

Security

- Firewall – ipfw:

```
#ipfw defaults
add 00100 allow ip from any to any via lo0
add 00200 deny ip from any to 127.0.0.0/8
add 00300 deny ip from 127.0.0.0/8 to any
add 00400 deny ip from any to ::1
add 00500 deny ip from ::1 to any
add 00600 allow ipv6-icmp from :: to ff02::/16
add 00700 allow ipv6-icmp from fe80::/10 to fe80::/10
add 00800 allow ipv6-icmp from fe80::/10 to ff02::/16
add 00900 allow ipv6-icmp from any to any ip6 icmp6types 1
add 01000 allow ipv6-icmp from any to any ip6 icmp6types 2,135,136

#allow ping6
add 01100 allow ipv6-icmp from any to any ip6 icmp6types 128,129

#allow ipv4
add 29000 allow ip4 from any to any

#allow traffic of the network to be routed
add 29300 allow ip6 from 2001:388:e000:a00::/56 to any via em0
add 29301 allow ip6 from 2001:388:e000:a00::/56 to any via lo0
```

Security

- Firewall – ipfw cont.:

```
#allow the tunnel endpoint to talk to the gateway address
add 29400 allow ip6 from 2001:388:f000::a79 to 2001:388:e000:a00::1

#allow the tunnel out
add 29500 allow ip6 from 2001:388:f000::a79 to 2001:388:f000::a78
#allow the tunnel in
add 29501 allow ip6 from 2001:388:f000::a78 to 2001:388:f000::a79

#allow all on the subnet out and back in, once initiated
add 30000 allow tcp from 2001:388:e000:a00::/56 to any out via tun0 setup keep-state
add 30001 allow udp from 2001:388:e000:a00::/56 to any out via tun0 keep-state

#allow the tunnel endpoint to initiated outbound connections
#the host uses the tunnel endpoint address as source IPv6 address..
add 30002 allow tcp from 2001:388:f000::a79 to any out via tun0 setup keep-state
add 30003 allow udp from 2001:388:f000::a79 to any out via tun0 keep-state

#allow ssh in
add 35000 allow tcp from any to 2001:388:e000:a00::/64 dst-port 22 setup keep-state

#block everything
add 65535 deny ip from any to any
```


OS specific info

- Windows:
  - Only Windows 7 supports IPv6 properly
  - Supports SLAAC and DHCPv6 (no PD)
  - IPv6 DNS has to be set up manually if DHCPv6 is not present (no RDNSS)
  - No mDNS
- MacOSX
  - MacOSX Lion has DHCPv6 support, older versions don't
  - Supports SLAAC, no RDNSS
  - Supports mDNS, not sure about nss_mdns, no Avahi,
  - IPv6 DNS has to be set up manually if DHCPv6 is not present

OS specific info

- Linux:
  - Similar to FreeBSD
  - Differences between distributions (e.g. Gentoo has no wide-dhcp)
  - No rtsol or ndp commands ☹
  - ifconfig <if> inet6 = ip -6
  - rtadvd = radvd (different syntax! Many examples on the net)
  - DS-Lite client available
  - IPv6 enabled by default
  - mDNS enabled by default on most distributions (enabled by default by GNOME and KDE)

Issues with IPv6

- Problem is missing glue between various tools
- Problem is dual-stack operation
  - can cause racing conditions for name resolution
  - Happy Eyeballs
  - Misconfigured servers cause bad user experience which is attributed to IPv6
- Problem is the complexity of Home Networks
  - IETF HomeNet WG tries to solve the issues (started at IETF-81)
  - http://www.potaroo.net/ispcol/2011-08/home.html
- The more we use IPv6, the quicker we can fix problems!

Conclusions

- IPv6 is easy to deploy for simple networks, specially if run in dual-stack mode
  - Small issues can arise
- Need a variety of tools to get going – no plug and play
- Different solutions available to get IPv6, platform dependent
- Have a try, it's fun!
- Check out http://www.potaroo.net for interesting IPv6 articles