IPv6: Notes on Setup, And Details of the New Protocol in Action

by Allan Bruce

TABLE OF CONTENTS

1. History of IP

The internet is expanding at a phenomenal rate due to lower hardware prices and the introduction of broadband. Many other devices such as PDAs and even mobile phones now have the ability to browse the web. Each device requires a unique address so that communication is successful. The IP protocol allocates these addresses. The current version, IPv4, has several limitations, mainly lack of address space, and security.

IPv4 allocates a 32-bit address to each node in the form of four 8-bit numbers in dotted notation, e.g. 139.133.204.80. This address structure allows for a maximum of approximately 4.3 billion addresses. This seems a lot, however the protocol has some hierarchical structure and also has reserved addresses, e.g. loopback (127.x.x.x), private local networks (192.168.x.x), and multicast (x.x.x.255). There are 'hacks' which increase address space such as NAT (Network Address Translation) so why do we need to modify the protocol? Address space is an issue with IPv4, but as mentioned earlier this is only one limitation. To overcome all of these issues, the Internetworking Group decided that a new protocol would be required. Instead of merely modifying the existing protocol, this newer version was re-designed from scratch keeping in mind the limitations of IPv4.

1.1 IPv6

This new protocol is called IPv6. The name comes from the number assigned to it in the version field in the header. IPv4 was number 4. Number 5 was already used, so 6 was assigned, hence IPv6. The protocol offers many advantages over its predecessor:

- Increased address space
- Auto-configuration
- Simplified header
- Improved extensions/options support
- Built in security options
- Less load for Routers

1.2 IPv6 Addresses

The address space of IPv6 has been increased to 128 bits, which offers approximately 3.4 x 10^{38} addresses. This is a very large number to comprehend so it is maybe easier to think, that this is enough for around 1000 addresses per square meter of the earths surface[1]! Practically, this number of addresses will not be required, so IPv6 offers a better hierarchical structure. An IPv6 address is split into eight 16-bit numbers separated by a colon, e.g. fe80::a00:20ff:fec1:19c9. It is useful to notice that any preceding 0's can be discarded for simplicity. Also, a string of 0's expanding over a whole field can be written using shorthand notation :: This must only be used once, as it would be impossible to evaluate how many 0's had been skipped. From a programming perspective, the address can be extracted by:

1. Copying the address from the start until ::
2. Copying the address from the end until ::

[1] Christian Huitema, "IPv6: The New Internet Protocol (second edition)", Prentice Hall, November 1997

The auto-configuration aspect is apparent in two ways. Each IPv6 node is assigned at least two IPv6 addresses. One of these may be a private address (or Link-local address), used only within a subnet. The address above is an example of this, these always begin fe80:: and have 64 trailing bits for the unique address. Another local address (Site-Local) is also defined but is not commonly used. These begin fec0:: and contain 16 bits for the subnet ID before the trailing 64 bits.
The auto-configurator assigns the last 6 Hex digits (24 bits) of the address to the last 6 of the MAC address of the Ethernet card in the machine. The above machine's MAC address is therefore x:x:x:c1:19:c9.
There is also a global address. This is the address that is advertised when communicating outwith the LAN. This address is only assigned when connected via a router. There is no concept of subnets in IPv6, however the global addresses do follow a hierarchy. My machine has global address 2001:630:241:0:a00:20ff:fec1:19c9. The first 3 fields are fixed to the institute/company which is stored in the IPv6 router to assign to machines. The MAC address, again, forms the latter 6 Hex digits of the IPv6 address (although these can be changed to any unique address if required). At the end of an IPv6 address, there is a / (or % on Microsoft Windows) followed by a number. This number indicates the number of bits in the 'subnet' for each machine to assign a unique address (these are the last bits of the address). This IPv6 auto-configuration supersedes DHCP used in IPv4 which still required a lot of work for system administrators.
The addresses of all IPv6 machines within ERG are shown in Appendix E.
There are issues concerning privacy with the auto-configured addresses. The address need not contain part of the MAC address, this is desired to stop tracing internet access. Privacy Extensions are defined in IPv6 which allow the addresses to be manually configured or allocated via DHCPv6. Another method is to randomly generate part of the address which can also change over time – indeed Microsoft have adopted this approach as default although this can be disabled if necessary.
To allow for backward compatibility, IPv4 addresses may also be implemented in IPv6. One special address which is used for tunnelling v6 packets over an IPv4 infrastructure (known as the IPv4-compatible IPv6 address) is as follows:

::x.x.x.x

This is merely the 32-bit IPv4 addresses preceded by a string of 0s.
IPv4 only nodes can be reached by using a special address (known as the IPv4 mapped IPv6 address). It has the format:

::ffff:x.x.x.x

This is the 32-bit IPv4 address preceded by a string of 0s then ffff. This is the method that must be used if IPv6 is not supported by the machine with IPv4 address specified[2].

1.3 IPv6 Header

The IPv6 header has been simplified greatly in comparison to that of IPv4. The new header is at least 40 Bytes which is larger than the 20 Bytes of IPv4 but considering the addresses are 8 times larger, this is an improvement. The new header now only comprises of:

[2] R. Hinden and S. Deering, "RFC 2373 - IP Version 6 Addressing Architecture", 1998

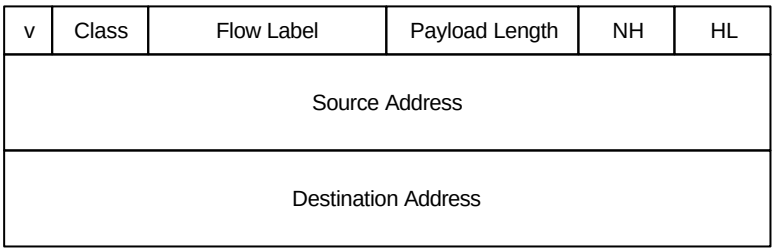


figure 1: IPv6 Header

- The first field, Version (4 bits), is set to 6. This field remains here to maintain backwards compatibility.
- The next field, DSCP+ECN (1 Byte), replaces the type of service field in Ipv4. This is used to distinguish between the priorities and classes of IPv6 packets
- Flow Label (20 bits), is used for packet classification on firewalls/QOS-enabled networks.
- Payload Length (2 Bytes), is the length of the total packet excluding the 20-byte header. Any optional/extension headers are included in this. This allows a maximum packet size of 64KB. This can be extended using Jumbograms (an IP extension header).
- Next Header (1 Byte), indicates the type of header following. This can be used for optional/extension headers or the next layer protocol.
- Hop Limit (1 Byte), replaces the TTL in IPv4. This is decremented by each router. A maximum of 255 hops is dictated, which is a special value to indicate the machine is on the local network.
- Source Address (16 Bytes), is the address of the sending node.
- Destination Address (16 Bytes), is the address of the intended recipient. This may be the ultimate destination as in IPv4 but could be different if a routing header is used.

RFC 2460 – IPv6 Specification, defines 6 Extension headers. An IPv6 packet may contain none or any combination of these headers. These headers are situated between the IPv6 header and the next layer header. The header is identified in the Next Header field.
The extension headers are only processed by the destination in order. One exclusion to this is the use of the Hop-by-Hop Options header. This is processed by every node and it must immediately follow the IPv6 header to reduce processing time for routers. For further information on the IPv6 Extension headers see IPv6 Essentials[3].
To see some extension headers captured, see 5.2 Using IPv6, for ICMPv6 and 5.3 Neighbour Discovery for Routing Headers.

[3] Silvia Hagen, "IPv6 Essentials: Integrating IPv6 into Your IPv4 Network", O'Reilly, July 2002

2. Set up of IPv6

IPv6 has been around for some time now, and as such is implemented in most Operating Systems already. In most, it is merely a case of adding a few commands or enabling a few options to get the kernel to use IPv6.

2.1 Windows XP

Under Windows XP, one command entered into the command prompt installs all necessary IPv6 options. This is:

```
ipv6 install
```

There is one option to be changed which Microsoft included as a security measure. Windows XP, by default, will change the local IPv6 address every 3 minutes which is fine for browsing but is undesired for authentication, e.g. for mail servers. This option can be disabled by:

```
netsh
interface ipv6 set privacy state=disabled
commit
```

A reboot is required to activate the option.

2.2 Linux

Linux has an advantage that the kernel is freely available which enables the distributions to be kept up-to-date and easily modified. This is one of the easiest operating system to setup for IPv6 (Kernel 2.4 and above). All that is required is to modify the /etc/rc.local file to include the line:

```
modprobe ipv6
```

A reboot is required to ensure the ipv6 drivers are loaded.

2.3 MacOS

IPv6 is native under MacOS 10.2 and above and is installed as default.

2.4 Solaris

Solaris 8 is the first version that supports IPv6 natively. When installing, an option needs to be flagged to include IPv6 support. Installing Solaris is a relatively straight-forward task, however, a few extra options need to be changed.
To allow the newly installed machine to get external network connectivity (IPv4), the /etc/defaultrouter should contain a line indicating the IPv4 address of the default gateway or router. To enable DNS lookups the /etc/resolv.conf file should contain a line to a valid DNS server like:

```
nameserver 139.133.204.80
```

To add users, type admintool on a shell. Once this has been done, it is now possible to use Solaris. It is recommended to patch the OS to the latest versions. To get the latest patches, type the following at the shell:

```
ftp sunsolve.sun.com
```

Log in with user:anonymous and a valid email address as password. Now type:

```
cd /pub/patches
bin
hash
get 8_Recommended.zip
quit
```

This will take some time as the file is rather big. Once downloaded the file needs to be extracted, by:

```
unzip 8_Recommended.zip
```

To install the patches type:

```
cd 8_Recommended
install_cluster
```

This will install all the patches in order. Some patches will not install but this should not be a problem. Check the logfile if in doubt.

3. Installing Apache Web Server

The installation of the Apache webserver (v2.0.45) itself is straightforward. Windows has an executable which self extracts and installs the necessary components. All that is needed is to edit the httpd.conf file to allow IPv6 connections, skip to section 3.3. Linux has a source-code release available, skip to section 3.2.

3.1 Solaris Requirements

Solaris requires some other necessary tools are required for installation of any source-code applications. Apache requires gcc to install, which also requires binutils. These applications are all available from sunfreeware or any of its mirrors, at a shell type:

```
ftp mirror.ac.uk
```

Login with user:anonymous pass:<email address>, then type:

```
cd /sites/ftp.sunfreeware.com/pub/freeware/sparc/8
bin
hash
get gcc-3.0.3-sol8-sparc-local.gz
get binutils-2.11.2-sol8-sparc-local.gz
quit
```

Later versions of gcc are available but they are a much bigger download. Each of these files are installed in the same way, by:

```
gunzip filename.gz
pkgadd –d ./filename
```

(The ./ may be removed if the path is set to search the current directory first). Binutils must be installed first, then gcc. Now the system is ready to install Apache.

3.2 Compiling Apache

For Linux and Solaris, Apache should be downloaded and extracted to a directory then the following commands should be typed

```
./configure –prefix=directory
./make
./make install
```

3.3 Apache Configuration

Apache should now be operational, but as is will not be able to accept IPv6 connections. To enable this, type the following commands from the apache directory:

```
cd conf
vi httpd.conf
```

Any text editor can be used here, vi is standard in Solaris/Linux, notepad is the equivalent under Windows. Search for 'Listen', and add the following line just above it:

```
Listen [IPv6address]:8080
```

Port 8080 is the default port for IPv6 web traffic. You should also complete the identification options for support, see the README included with Apache for more details. (You may also like to increase the number of keepaliverequests to enable more efficient management of http connections).

3.4 Starting Apache

To start the server type

```
/directory/apachectl start
```

Under Windows, click the icon to 'Monitor Apache Servers' then click on start.
Now, Apache should be up and running and will accept IPv6 web traffic.

4. Useful Tools

Some useful tools for network analysis have been modified to use IPv6 also. Solaris includes a utility called snoop. To capture only IPv6 traffic the following should be typed:

```
snoop ip6 [options]
```

4.1 Tcpdump

Another capture tool that is often used is tcpdump. This program is available for most Operating Systems therefore is more widely supported than snoop. To install tcpdump, libpcap is required. Libpcap and tcpdump should be downloaded from http://www.tcpdump.org/ for Linux and Solaris or from http://windump.polito.it/ for Windows executables. Windows installation is trivial - skip to 4.3. For Linux installation, skip to section 4.2. Solaris also requires flex, bison and GNU M4. These can be downloaded from sunfreeware as indicated above. Once in the site type:

```
get flex-2.5.4a-sol8-sparc-local.gz
get bison-1.875-sol8-sparc-local.gz
get m4-1.4-sol8-sparc-local.gz
```

These packages are installed as detailed above.

4.2 Compiling tcpdump

The downloaded files will be in the form of gzipped tar files. To install these (v3.7.2) follow the commands shown here:

```
gunzip filename          (not needed in Linux)
tar xvf filename         (use tar xzvf filename in Linux)
cd filename
./configure –enable-ipv6
./make
./make install
```

This should be done for libcap, then tcpdump.
Tcpdump should now be ready for use.

4.3 Running tcpdump

Tcpdump should be run by:

```
tcpdump ip6 [options]
```

It should be noted that some versions of tcpdump use ipv6 as the option instead of ip6.

5. Seeing IPv6 in action

Once these above steps are complete, it is now possible to use IPv6 to browse web pages and to see the protocol in action using one of the monitoring tools.
The University of Aberdeen was given a range of addresses and ERG uses 2001:630:241::/64.

5.1 Is IPv6 installed correctly?

It is a good idea to make sure that the system is currently configured correctly for ipv6.
Under Solaris, this is done using two commands:

```
ifconfig –a

and

netstat –rn
```

Running ifconfig –a should give a similar output to the following:

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
le0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 139.133.210.30 netmask ffffff00 broadcast 139.133.210.255
        ether 8:0:20:1b:2e:ce
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
le0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 8:0:20:1b:2e:ce
        inet6 fe80::a00:20ff:fe1b:2ece/10
le0:1: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        inet6 2001:630:241:4:a00:20ff:fe1b:2ece/64
```

The first line indicates the loopback address for IPv4.
The following line, is the Ethernet device and shows the IPv4 address and MAC address of the card itself.
The next lines are what indicates whether IPv6 is present or not. First, there is the IPv6 loopback interface which is ::1 (the /128 shows that the last 128 bits are considered part of the local address structure, i.e. all of it in this case).
Line 4 shows the Ethernet interface with a Link-local IPv6 address. It can be seen that the last 24 bits of the MAC address correspond to the last 24 bits of the IPv6 Link-local address (auto-configured).
Line 5 shows the alternate IPv6 address for this node. This is the global address configured by the IPv6 router. The first 64-bits (in this case) are assigned by the router, leaving 64-bits for a type of subnet.
It can be seen that the two IPv6 addresses both have a common number of bits (0a00:20ff). This appears to be the same for most Solaris machines. Indeed, most Microsoft Windows XP machines have this similarity too, but the common bits are different (0260:97ff). Mac OS 10.2.5 mostly uses 0230:65ff. This common field is due to the manufacturer of the NIC[4].

[4] M. Crawford, "Transmission of IPv6 Packets over Ethernet Networks", 1998

Most PCs have Intel/3Com cards, most Sun machines use Sun cards and most Macs use the same manufacturer, thus giving an indication into which machine a user may be using.

Running netstat –rn gives:

```
Routing Table: IPv6
  Destination/Mask        Gateway                     Flags     Ref   Use    If
--------------------------- --------------------------- -----  ---  ------ -----
2001:630:241:4::/64         2001:630:241:4:a00:20ff:fe1b:2ece   U       1      2    le0:1
fe80::/10                   fe80::a00:20ff:fe1b:2ece    U       1      1    le0
ff00::/8                    fe80::a00:20ff:fe1b:2ece    U       1      0    le0
default                     2001:630:241:4:207:85ff:fe60:3ba0   UG      1      6
::1                         ::1                         UH      1      0    lo0
```

This shows the routing table for IPv6. The fourth entry is a router acting as a gateway for default traffic.

Under Windows, the command

```
ipconfig
```

is used to determine if IPv6 is installed correctly on the machine. A typical output from this is shown:

Ethernet adapter Local Area Connection:

```
Connection-specific DNS Suffix  . :
IP Address . . . . . . . . . . . . . . . . . . . : 192.168.0.4
Subnet Mask . . . . . . . . . . . . . . . . . . . : 255.255.255.0
IP Address . . . . . . . . . . . . . . . . . . . : fe80::260:97ff:fe27:9283%4
Default Gateway . . . . . . . . . . . . . . . . . : 192.168.0.2
```

Tunnel adapter Automatic Tunneling Pseudo-Interface:

```
Connection-specific DNS Suffix . :
IP Address. . . . . . . . . . . . . . . . . . . . : fe80::5efe:192.168.0.4%2
Default Gateway . . . . . . . . . . . . . . . . . :
```

This machine is standalone and does not have any routing but netstat –rn can also be used in Windows to find the routing tables.

5.2 Using IPv6

Once everything is set up and verified, IPv6 can be used. If an IPv6 route is available then most software will use it over IPv4, however some software requires extra options to use IPv6. Ping uses ICMP to get an echo from a host. This is used to determine if a route to the host exists and both nodes are set up to communicate. Ping also uses ICMPv6 to determine if an IPv6 route exists. If the machine specified with the ping command is an IPv6 address or

IPv6 only name then ping will use IPv6 as default. If, however, the name specified has two host entries (DNS), then IPv6 options needs to be specified, e.g.

```
ping -A inet6 -a hostname
```

This will send echo requests until the user breaks the command with ^C. An example ping request on ICMPv6 was captured using tcpdump[5]. The packets are shown below.

```
2001:630:241:4:a00:20ff:fe1b:2ece > 2001:630:241:0:a00:20ff:fec1:19c9: icmp: echo request
                         6000 0000 0040 3a3c 2001 0630 0241 0004
                          0a00 20ff fe1b 2ece 2001 0630 0241 0000    IPv6 Header
                          0a00 20ff fec1 19c9 8000 2c38 213d 0000    ICMPv6 Header
                          3ea6 be2a 000e 5dd2 0809 0a0b 0c0d 0e0f    ICMPv6 Payload
                          1011 1213 1415 1617 1819 1a1b 1c1d 1e1f
                         2021
2001:630:241:0:a00:20ff:fec1:19c9 > 2001:630:241:4:a00:20ff:fe1b:2ece: icmp: echo reply
                         6000 0000 0040 3afe 2001 0630 0241 0000
                          0a00 20ff fec1 19c9 2001 0630 0241 0004
                         0a00 20ff fe1b 2ece 8100 2b38 213d 0000
                          3ea6 be2a 000e 5dd2 0809 0a0b 0c0d 0e0f
                          1011 1213 1415 1617 1819 1a1b 1c1d 1e1f
                         2021
```

IPv6 header:

- The first 4-bits is 6, therefore we have an IPv6 packet.
- The following byte is 0x0, therefore is uncharacterised traffic (DSCp = 0, and ECN off).
- The next 20 bits are also 0x0 therefore no flow ID. The following 2 Bytes indicate the length of the data after the IPv6 header. This is 0x40 = 84 for this case. This can be verified by counting the total bits and subtracting the IPv6 header bits (total = 164 bits, IPv6 header = 80 bits, leaving 84 bits)
- The next Byte indicates the type of header following, in this case 0x3a =58, therefore ICMPv6 header follows
- The hop limit follows occupying 1 Byte. This is 0x3c for the echo request (60 – the default for unicast under Solaris) and 0xfe for the reply (254). The 254 allows us to see that the target was 2 hops away (1 intermediate node reduced the max of 255 by 1 before reaching this node)
- The next 256 bits are the source address followed by the destination address

IPv6 additional header (ICMPv6)

- The first Byte of the ICMPv6 header is the type, 0x80(128) for the request and 0x81(129) for the reply
- The following Byte is not used by echo/request and is therefore set to 0x0
- The next 2 Bytes contain the checksum for the ICMPv6 header, these are 0x2c38 and 0x2b38 for the example above.
- The remaining 38 Bytes is the data for echo request including timestamp for RTT measurement.

(A table of ICMPv6 codes can be found in Appendix F.)
Using snoop to capture one of the packets looks like:

[5] tcpdump –x ipv6 *host* (v3.7.1)

```
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 254
IPv6:   Source address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:   Destination address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 129 (Echo reply)
ICMPv6:  Code = 0 (ID: 8509 Sequence number: 0)
ICMPv6:  Checksum = 2b38
```

A telnet packet was captured as shown:

```
2001:630:241:4:a00:20ff:fe1b:2ece.35479 > 2001:630:241:0:a00:20ff:fec1:19c9.23: P
52465719:852465725(6) ack 2705672248 win 25920
                         6000 0000 001a 063c 2001 0630 0241 0004
                         0a00 20ff fe1b 2ece 2001 0630 0241 0000
                         0a00 20ff fec1 19c9 8a97 0017 32cf 9837
                         a145 4838 5018 6540 20f9 0000 fffd 01ff
                         fc01
```

The IPv6 header is similar to that analysed above. The length of the IPv6 data is 0x1a which is 26 bytes. The next header this time is 0x06 (6) which is TCP. The hop limit is advertised as 0x3c which is 61 bytes.
The TCP header follows the IPv6 header[6].

- The Source port occupies thee first two Bytes, which is 0x8a97 (35479) in this case.
- The destination port is 0x0017 (23 - Telnet).
- The next 4 Bytes indicate the sequence number – 0x32cf9837 (852465719). This can be compared with tcpdump (the status line shows 852465725(6) which is the next sequence number to expect and the length of the current packet, therefore resulting in what is expected).
- The next 4 Bytes indicate the ACK – 0xa1454838(2705672248) which again can be compared with tcpdump.
- The following 4 bits represent the header length (5). This is in 4-Byte blocks, therefore the header length is 20 Bytes, thus containing no options
- The next 6 bits are reserved
- Then the flags follow – The ACK and PSH flags are present in the example.
- The Window size follows which is 2 Bytes - 0x6540 (25920) which can be compared with tcpdump
- The next 2 Bytes bits are the TCP checksum (0x20f9)
- The last 2 Bytes of the TCP header is the urgent pointer (0x0000)

[6] W. Richard Stevens, “TCP/IP Illustrated, Volume 1: The Protocols”, Addison Wesley, February 1994

The remaining data is for the telnet protocol:

- The first Byte (0xff) indicates that the next Byte is a command. In this case, the next Byte is 0xfd (253) which is a DO negotiation option
- The next Byte (0x01) indicates an echo
- Another command follows (due to the 0xff). This time the command is 0xfc (252) which is a WONT negotiation option
- Finally, another echo completes the last Byte (0x01)

This complete telnet trace can be found in Appendix A – Telnet Trace

5.3 Neighbour Discovery

Neighbour Discovery is used to determine neighbouring routers that can forward their packets, detect which neighbours are reachable and determine the layer 2 addresses for nodes on the same link. Neighbour Discovery uses ICMPv6 and makes use of Neighbour Solicitation and Neighbour Advertisement. The first packets sent out for Neighbour Discovery is shown below:

```
fe80::a00:20ff:fec1:19c9 > ff02::2: icmp: multicast listener ff02::202 done
                  6000 0000 0018 3aff fe80 0000 0000 0000
                  0a00 20ff fec1 19c9 ff02 0000 0000 0000
                  0000 0000 0000 0002 8400 3998 0000 0000
                  ff02 0000 0000 0000 0000 0000 0000 0202
fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: router advertisment maxhlim=64  life=1800
nud=0/0 [(src LLaddr 0:7:85:60:3b:a1) (MTU 1500) |] [dsb 224]
                  6e00 0000 0040 3aff fe80 0000 0000 0000
                  0207 85ff fe60 3ba1 ff02 0000 0000 0000
                  0000 0000 0000 0001 8600 377a 4000 0708
                  0000 0000 0000 0000 0101 0007 8560 3ba1
                  0501 0000 0000 05dc 0304 40c0 0000 012c
                 0000
```

The first packet, gets sent from the machine just rebooted to ff02::2 which is the reserved address for all routers. The second packet gets sent to ff02::1 which is the reserved address for all nodes. Both packets use ICMPv6 (0x3a) which is used for ND. The second packet has its Traffic Class field as 0xe0.
Looking at the ICMPv6 packets, the first has type 0x84 (132) which is "Multicast Listener Done". The second packet has type 0x86 (134) which is a "Router Advertisement". Looking at the second packet, the Hop Limit is 0x40 (64) and the Byte following is flags. The next 2 Bytes indicate the Router Lifetime. This is 0 for a non-default router, but in this case is set to 0x0708 (1800). The next 4 Bytes are the Reachable time, but these are unspecified in the example above. The next 4 Bytes are also unspecified, these indicate the Retransmission Timer. Options follow this which may include MTU or source-link addresses. The rest of this trace may be found in

Appendix B – Neighbour Discovery Trace.

5.4 Web transfers

One of the main reasons for using IPv6 is for increasing the address space. Since the most common use of the internet is for the web, a quick analysis was carried out. IPv6 worked as standard IP. It was noted that IPv6[7] made use of persistent connections which allows TCP to send the multiple data files in one connection increasing the throughput (since no slow-start is required for new data items). Some packets of a web transfer are shown in Appendix C.

5.5 Traceroute v6

A quick look at traceroute shows that it behaves as one would expect it to with IPv4 addresses. An example output is shown below:

```
root @ genesis 146 > traceroute blake
traceroute: Warning: Multiple interfaces found; using 2001:630:241:0:a00:20ff:fe74:940c @
le0:1
traceroute to blake.erg.abdn.ac.uk (2001:630:241:4:a00:20ff:fe1b:2ece), 30 hops max, 60 byte
packets
 1  door-ipv6.erg.abdn.ac.uk (2001:630:241:0:207:85ff:fe60:3ba1)  3.157 ms  2.256 ms  2.274 ms
 2  blake.erg.abdn.ac.uk (2001:630:241:4:a00:20ff:fe1b:2ece)  3.584 ms  3.346 ms  3.548 ms
```

(Under Windows, the command for traceroute is tracert.)

5.6 Neighbor Cache

Ipv6 has an alternative to ARP known as Neighbor Cache. This can be found by either:

```
show ipv6 neig
ndp –a
```

or

```
netstat –p
```

The outputs from these are shown below (consecutively):

```
IPv6 Address                              Age Link-layer Addr State
Interface
FE80::A00:20FF:FE86:ECDF                    0 0800.2086.ecdf  STALE Fa0/1.1
FE80::202:2DFF:FE46:4306                   59 0002.2d46.4306  STALE Fa0/1.3
2001:630:241:1:230:65FF:FE18:81E6           0 0030.6518.81e6  REACH Fa0/1.3
2001:630:241:4:A00:20FF:FE1B:2ECE           0 0800.201b.2ece  REACH Fa0/0
FE80::A00:20FF:FEC1:19C9                  266 0800.20c1.19c9  STALE Fa0/1.1
FE80::A00:20FF:FE1B:2ECE                   57 0800.201b.2ece  STALE Fa0/0
FE80::230:65FF:FE18:81E6                    0 0030.6518.81e6  REACH Fa0/1.3
2001:630:241:1:202:2DFF:FE46:4306         102 0002.2d46.4306  STALE Fa0/1.3
2001:630:241:0:A00:20FF:FE86:ECDF           0 0800.2086.ecdf  STALE Fa0/1.1
```

```
Neighbor                      Linklayer Address  Netif Expire    St Flgs
```

[7] using Apache v2.0.45 and Netscape 7

```
Prbs
inspiration-ipv6.erg.abdn.ac.uk 0:30:65:18:81:e6     en1 permanent R
fe80::1%lo0                     (incomplete)         lo0 permanent R
fe80::207:85ff:fe60:3ba1%en1    0:7:85:60:3b:a1      en1 13s       R  R
fe80::230:65ff:fe18:81e6%en1    0:30:65:18:81:e6     en1 permanent R
```

```
Net to Media Table: IPv4
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
eri0   milliways-erg.erg.abdn.ac.uk 255.255.255.255       00:d0:bb:f7:c6:c1
eri0   mavis.erg.abdn.ac.uk 255.255.255.255       08:00:20:86:ec:df
eri0   gordon.erg.abdn.ac.uk 255.255.255.255       08:00:20:96:10:1a
eri0   shushu.erg.abdn.ac.uk 255.255.255.255       00:10:a4:c0:9c:aa
eri0   maomao.erg.abdn.ac.uk 255.255.255.255       00:00:39:57:3c:91
eri0   blade.erg.abdn.ac.uk 255.255.255.255 SP    00:03:ba:09:08:7c
eri0   globecast.erg.abdn.ac.uk 255.255.255.255       00:60:97:a0:e4:5b
eri0   BASE-ADDRESS.MCAST.NET 240.0.0.0       SM    01:00:5e:00:00:00

Net to Media Table: IPv6
 If   Physical Address    Type      State      Destination/Mask
----- -----------------  ------- ------------ --------------------------
eri0  33:33:00:00:00:00  other   REACHABLE    ff00::
eri0  00:03:ba:09:08:7c  local   REACHABLE    fe80::203:baff:fe09:87c
eri0  00:03:ba:09:08:7c  local   REACHABLE
2001:630:241:0:203:baff:fe09:87c
eri0  00:07:85:60:3b:a1  dynamic REACHABLE    fe80::207:85ff:fe60:3ba1
```

6 Discussion

IPv6 was set out to address problems with the old IPv4. These have succeeded, but there are a few interesting points to note about IPv6.

The first interesting *feature* is that IPv6 auto-configured addresses appear to have several bytes which may identify the Operating System of the node. These bytes are dependant on the manufacturer of the NIC. Since most Windows machines use the same manufacturer and Sun use a different manufacturer etc., it gives an indication as to which OS the user may be running.

IPv6 Web transfers appear slightly quicker than their IPv4 counterparts. This may be due to coherence with HTTP 1.1 standards and better use of Persistent connections.
A few ftp transfers were carried out and the average speed of an IPv4 transfer was 1070KB/s, whereas IPv6 was only 1007KB/s, almost a 1% reduction. This is due to the extended size of the IPv6 header. Most LANs use an MTU of 1500 Bytes, but 84 Bytes of this is used by Ethernet, 20 or 40 by IP (v4 or v6 respectively) and a further 20 Bytes by TCP. This leaves 1376 Bytes of data for IPv4 or 1356 for IPv6 (which is 1.5% less).

It has to be mentioned that *nslookup* has been superseded also with IPv6. The *host* command should now be used to find the address of a host. There are three possible types of entry for the host table:

1. A – IPv4
2. AAAA – IPv6
3. A6 – IPv6

These can be specified when running the command, e.g.

```
Host –t aaaa hostname
```

A dump of one of these requests is shown below:

```
13:50:08.280835 genesis.erg.abdn.ac.uk.37199 > gordon.erg.abdn.ac.uk.domain: 19782+ AAAA?
endeavour-ipv6.erg.abdn.ac.uk. (47) (DF)
                        4500 004b eb61 4000 ff11 e063 8b85 cc80
                        8b85 cc50 914f 0035 0037 43d2 4d46 0100
                        0001 0000 0000 0000 0e65 6e64 6561 766f
                        7572 2d69 7076 3603 6572 6704 6162 646e
                        0261 6302 756b 0000 1c00 01

13:50:08.282966 gordon.erg.abdn.ac.uk.domain > genesis.erg.abdn.ac.uk.37199: 19782* 1/2/2
(153) (DF)
                        4500 00b5 d911 4000 ff11 f249 8b85 cc50
                        8b85 cc80 0035 914f 00a1 cd99 4d46 8580
                        0001 0001 0002 0002 0e65 6e64 6561 766f
                        7572 2d69 7076 3603 6572 6704 6162 646e
                        0261 6302 756b 0000 1c00 01c0 0c00 1c00
                        0100
```

This is also shown using tcpdump's verbose mode (tcpdump –vvv) below:

```
13:50:08.280835 genesis.erg.abdn.ac.uk.37199 > gordon.erg.abdn.ac.uk.domain: 19782+ AAAA?
endeavour-ipv6.erg.abdn.ac.uk. (47) (DF) (ttl 255, id 60257)
```

```
13:50:08.282966 gordon.erg.abdn.ac.uk.domain > genesis.erg.abdn.ac.uk.37199: 19782* q:
endeavour-ipv6.erg.abdn.ac.uk. 1/2/2 endeavour-ipv6.erg.abdn.ac.uk. (153) (DF) (ttl 255, id
55569)
```

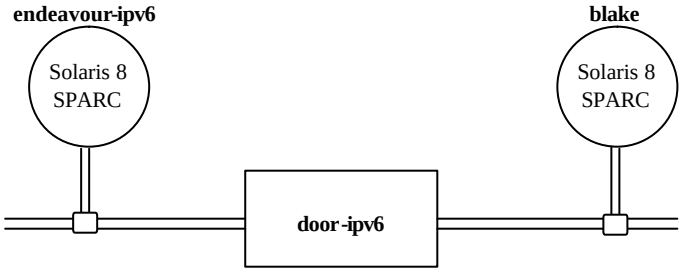
Appendix A – Telnet Trace

This is a snoop trace of a telnet session over IPv6

Host name: endeavour-ipv6
Host address: 2001:630:241:0:a00:20ff:fec1:19c9
Host MAC: 8:0:20:c1:19:c9
Host OS: Solaris 8 (SPARC)

Server name: blake
Server address: 2001:630:241:4:a00:20ff:fe1b:2ece
Server MAC: 8:0:20:1b:2e:ce
Server OS: Solaris 8 (SPARC)

Router name: door-ipv6
Router MACs: 0:7:85:60:3b:a0 (blake side)
0:7:85:60:3b:a1 (endeavour-ipv6 side)




```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 16 arrived at 17:25:35.74
ETHER:  Packet size = 87 bytes
ETHER:  Destination = 8:0:20:1b:2e:ce, Sun
ETHER:  Source      = 0:7:85:60:3b:a0,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 33
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 59
IPv6:   Source address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:   Destination address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 23
TCP:  Destination port = 34432
TCP:  Sequence number = 606986077
TCP:  Acknowledgement number = 1562884143
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
```

```
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0x5d9a
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  "\377\373\1\377\375\1login: "
TELNET:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 17 arrived at 17:25:35.74
ETHER:  Packet size = 80 bytes
ETHER:  Destination = 0:7:85:60:3b:a0,
ETHER:  Source      = 8:0:20:1b:2e:ce, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 26
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:   Destination address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 34432
TCP:  Destination port = 23 (TELNET)
TCP:  Sequence number = 1562884143
TCP:  Acknowledgement number = 606986090
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0xc0a5
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  ""
TELNET:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 18 arrived at 17:25:35.75
ETHER:  Packet size = 77 bytes
ETHER:  Destination = 8:0:20:1b:2e:ce, Sun
ETHER:  Source      = 0:7:85:60:3b:a0,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 23
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 59
IPv6:   Source address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:   Destination address = 2001:630:241:4:a00:20ff:fe1b:2ece
```

```
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 23
TCP:  Destination port = 34432
TCP:  Sequence number = 606986090
TCP:  Acknowledgement number = 1562884149
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0xbda2
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  ""
TELNET:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 19 arrived at 17:25:35.84
ETHER:  Packet size = 74 bytes
ETHER:  Destination = 0:7:85:60:3b:a0,
ETHER:  Source      = 8:0:20:1b:2e:ce, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 20
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:   Destination address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 34432
TCP:  Destination port = 23 (TELNET)
TCP:  Sequence number = 1562884149
TCP:  Acknowledgement number = 606986093
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0xbea9
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  ""
TELNET:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 20 arrived at 17:25:37.29
ETHER:  Packet size = 75 bytes
ETHER:  Destination = 0:7:85:60:3b:a0,
ETHER:  Source      = 8:0:20:1b:2e:ce, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
```

```
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 21
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:   Destination address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 34432
TCP:  Destination port = 23 (TELNET)
TCP:  Sequence number = 1562884149
TCP:  Acknowledgement number = 606986093
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0x5da0
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  "a"
TELNET:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 21 arrived at 17:25:37.29
ETHER:  Packet size = 75 bytes
ETHER:  Destination = 8:0:20:1b:2e:ce, Sun
ETHER:  Source      = 0:7:85:60:3b:a0,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 21
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 59
IPv6:   Source address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:   Destination address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 23
TCP:  Destination port = 34432
TCP:  Sequence number = 606986093
TCP:  Acknowledgement number = 1562884150
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0x5d9f
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  "a"
TELNET:
```

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 22 arrived at 17:25:37.39
ETHER:  Packet size = 74 bytes
ETHER:  Destination = 0:7:85:60:3b:a0,
ETHER:  Source      = 8:0:20:1b:2e:ce, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 20
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:   Destination address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 34432
TCP:  Destination port = 23 (TELNET)
TCP:  Sequence number = 1562884150
TCP:  Acknowledgement number = 606986094
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0xbea7
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  ""
TELNET:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 23 arrived at 17:25:37.52
ETHER:  Packet size = 75 bytes
ETHER:  Destination = 0:7:85:60:3b:a0,
ETHER:  Source      = 8:0:20:1b:2e:ce, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 21
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = 2001:630:241:4:a00:20ff:fe1b:2ece
IPv6:   Destination address = 2001:630:241:0:a00:20ff:fec1:19c9
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 34432
TCP:  Destination port = 23 (TELNET)
TCP:  Sequence number = 1562884150
TCP:  Acknowledgement number = 606986094
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
```

```
TCP:  Checksum = 0x529e
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TELNET:  ----- TELNET:   -----
TELNET:
TELNET:  "l"
TELNET:
```

This is shown as a hex decode below:

```
1051120106.993356 2001:630:241:4:a00:20ff:fe1b:2ece.35479 >
2001:630:241:0:a00:20ff:fec1:19c9.23: S 852465662:852465662(0) win 25920 <nop,nop,sackOK,mss
1440>
                        6000 0000 001c 063c 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 8a97 0017 32cf 97fe
                        0000 0000 7002 6540 dc1b 0000 0101 0402
                        0204 05a0
1051120106.996330 2001:630:241:0:a00:20ff:fec1:19c9.23 >
2001:630:241:4:a00:20ff:fe1b:2ece.35479: S 2705672132:2705672132(0) ack 852465663 win 25920
<nop,nop,sackOK,mss 1440>
                        6000 0000 001c 063b 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 0017 8a97 a145 47c4
                        32cf 97ff 7012 6540 f300 0000 0101 0402
                        0204 05a0
1051120106.996603 2001:630:241:4:a00:20ff:fe1b:2ece.35479 >
2001:630:241:0:a00:20ff:fec1:19c9.23: . ack 2705672133 win 25920
                        6000 0000 0014 063c 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 8a97 0017 32cf 97ff
                        a145 47c5 5010 6540 1fb1 0000
1051120107.001469 2001:630:241:4:a00:20ff:fe1b:2ece.35479 >
2001:630:241:0:a00:20ff:fec1:19c9.23: P 852465663:852465687(24) ack 2705672133 win 25920
                        6000 0000 002c 063c 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 8a97 0017 32cf 97ff
                        a145 47c5 5018 6540 c33f 0000 fffd 03ff
                        fb18 fffb 1fff fb20 fffb 21ff fb22 fffb
                        27ff
1051120107.003668 2001:630:241:0:a00:20ff:fec1:19c9.23 >
2001:630:241:4:a00:20ff:fe1b:2ece.35479: . ack 852465687 win 25920
                        6000 0000 0014 063b 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 0017 8a97 a145 47c5
                        32cf 9817 5010 6540 1f99 0000
1051120107.073984 2001:630:241:0:a00:20ff:fec1:19c9.23 >
2001:630:241:4:a00:20ff:fe1b:2ece.35479: P 2705672133:2705672148(15) ack 852465687 win 25920
                        6000 0000 0023 063b 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 0017 8a97 a145 47c5
                        32cf 9817 5018 6540 c441 0000 fffd 18ff
                        fd1f fffd 23ff fd27 fffd 24
1051120107.074306 2001:630:241:4:a00:20ff:fe1b:2ece.35479 >
2001:630:241:0:a00:20ff:fec1:19c9.23: . ack 2705672148 win 25920
                        6000 0000 0014 063c 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 8a97 0017 32cf 9817
                        a145 47d4 5010 6540 1f8a 0000
1051120107.075544 2001:630:241:4:a00:20ff:fe1b:2ece.35479 >
2001:630:241:0:a00:20ff:fec1:19c9.23: P 852465687:852465702(15) ack 2705672148 win 25920
                        6000 0000 0023 063c 2001 0630 0241 0004
                        0a00 20ff fe1b 2ece 2001 0630 0241 0000
                        0a00 20ff fec1 19c9 8a97 0017 32cf 9817
                        a145 47d4 5018 6540 5757 0000 fffa 1f00
                        7e00 19ff f0ff fc23 fffc 24
```

Appendix B – Neighbour Discovery Trace

This trace shows the IPv6 traffic from endeavour-ipv6 (2001:630:241:0:a00:20ff:fec1:19c9) to its subnet after reboot, and responses from door-ipv6:

```
18:26:39.976604 fe80::a00:20ff:fec1:19c9 > ff02::2: icmp: [|icmp] [ttl 255]
                         6000 0000 0018 3aff fe80 0000 0000 0000
                         0a00 20ff fec1 19c9 ff02 0000 0000 0000
                         0000 0000 0000 0002 8400 3998 0000 0000
                         ff02 0000 0000
18:26:41.336687 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
18:27:05.653184 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
18:27:29.893866 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
18:27:50.251763 fe80::a00:20ff:fec1:19c9 > ff02::1:ffc1:19c9: icmp: [|icmp] [ttl 255]
                         6000 0000 0018 3aff fe80 0000 0000 0000
                         0a00 20ff fec1 19c9 ff02 0000 0000 0000
                         0000 0001 ffc1 19c9 8300 0984 0000 0000
                         ff02 0000 0000
18:27:58.174350 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
18:27:58.456141 fe80::a00:20ff:fec1:19c9 > ff02::202: icmp: [|icmp] [ttl 255]
                         6000 0000 0018 3aff fe80 0000 0000 0000
                         0a00 20ff fec1 19c9 ff02 0000 0000 0000
                         0000 0000 0000 0202 8300 3898 0000 0000
                         ff02 0000 0000
18:28:06.871561 fe80::a00:20ff:fec1:19c9 > ff02::202: icmp: [|icmp] [ttl 255]
                         6000 0000 0018 3aff fe80 0000 0000 0000
                         0a00 20ff fec1 19c9 ff02 0000 0000 0000
                         0000 0000 0000 0202 8300 3898 0000 0000
                         ff02 0000 0000
18:28:26.879051 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
18:28:52.675620 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
18:29:21.660370 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
18:29:48.472910 fe80::207:85ff:fe60:3ba1 > ff02::1: icmp: [|icmp] [dsb 224] [ttl 255]
                         6e00 0000 0040 3aff fe80 0000 0000 0000
                         0207 85ff fe60 3ba1 ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 377a 4000 0708
                         0000 0000 0000
```

A verbose output from snoop is shown below:

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 16 arrived at 18:26:39.97
```

```
ETHER:  Packet size = 78 bytes
ETHER:  Destination = 33:33:0:0:0:2, (multicast)
ETHER:  Source      = 8:0:20:c1:19:c9, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 24
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::a00:20ff:fec1:19c9
IPv6:   Destination address = ff02::2
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 132 (Group membership termination)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 3998

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 18 arrived at 18:26:41.33
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 72 arrived at 18:27:29.89
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
```

```
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 97 arrived at 18:27:58.17
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 98 arrived at 18:27:58.45
ETHER:  Packet size = 78 bytes
ETHER:  Destination = 33:33:0:0:2:2, (multicast)
ETHER:  Source      = 8:0:20:c1:19:c9, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
```

```
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 24
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::a00:20ff:fec1:19c9
IPv6:   Destination address = ff02::202
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 131 (Group membership report)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 3898
ICMPv6:  Multicast address= ff02::202
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 105 arrived at 18:28:6.87
ETHER:  Packet size = 78 bytes
ETHER:  Destination = 33:33:0:0:2:2, (multicast)
ETHER:  Source      = 8:0:20:c1:19:c9, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 24
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::a00:20ff:fec1:19c9
IPv6:   Destination address = ff02::202
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 131 (Group membership report)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 3898
ICMPv6:  Multicast address= ff02::202
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 127 arrived at 18:28:26.87
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
```

```
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 153 arrived at 18:28:52.67
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 175 arrived at 18:29:21.66
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
```

```
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 197 arrived at 18:29:48.47
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 224 arrived at 18:30:39.08
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
```

```
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 246 arrived at 18:31:8.24
ETHER:  Packet size = 118 bytes
ETHER:  Destination = 33:33:0:0:0:1, (multicast)
ETHER:  Source      = 0:7:85:60:3b:a1,
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 224
IPv6:   Flow label = 0x0
IPv6:   Payload length = 64
IPv6:   Next Header = 58 (ICMPv6)
IPv6:   Hop Limit = 255
IPv6:   Source address = fe80::207:85ff:fe60:3ba1
IPv6:   Destination address = ff02::1
IPv6:
ICMPv6:  ----- ICMPv6 Header -----
ICMPv6:
ICMPv6:  Type = 134 (Router advertisement)
ICMPv6:  Code = 0
ICMPv6:  Checksum = 377a
ICMPv6:  Max hops= 64, Router lifetime= 1800
ICMPv6:  Managed addr conf flag: NOT SET, Other conf flag: NOT SET
ICMPv6:  Reachable time: 0, Reachable retrans time 0
ICMPv6:
ICMPv6:  +++ ICMPv6 Source LL Addr option +++
ICMPv6:  Link Layer address: 0:7:85:60:3b:a1
ICMPv6:
ICMPv6:  +++ ICMPv6 MTU option +++
ICMPv6:  MTU = 1500
ICMPv6:
ICMPv6:  +++ ICMPv6 Prefix option +++
ICMPv6:  Prefix length = 64
ICMPv6:  Onlink flag: SET, Autonomous addr conf flag: SET
ICMPv6:  Valid Lifetime 300, Preferred Lifetime 300
ICMPv6:  Prefix 2001:630:241::
ICMPv6:
```

Appendix C – Web Transfer Trace

This is a snoop trace of a telnet session over IPv6

Host name: daisy-ipv6
Host address: 2001:630:241:0:a00:20ff:fe1d:21e0
Host MAC: 8:0:20:1d:21:e0
Host OS: Solaris 8 (SPARC)

Server name: genesis-ipv6
Server address: 2001:630:241:0:a00:20ff:fe74:940c
Server MAC: 8:0:20:74:94:0c
Server OS: Solaris 8 (SPARC)

Router name: none


```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 1 arrived at 13:46:23.49
ETHER:  Packet size = 82 bytes
ETHER:  Destination = 8:0:20:74:94:c, Sun
ETHER:  Source      = 8:0:20:1d:21:e0, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 28
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = fe80::a00:20ff:fe1d:21e0
IPv6:   Destination address = fe80::a00:20ff:fe74:940c
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 33028
TCP:  Destination port = 80 (HTTP)
TCP:  Sequence number = 2912560765
TCP:  Acknowledgement number = 0
TCP:  Data offset = 28 bytes
TCP:  Flags = 0x02
TCP:        ..0. .... = No urgent pointer
TCP:        ...0 .... = No acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..1. = Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
```

```
TCP:  Checksum = 0xc307
TCP:  Urgent pointer = 0
TCP:  Options: (8 bytes)
TCP:    - No operation
TCP:    - No operation
TCP:    - SACK permitted option
TCP:    - Maximum segment size = 1440 bytes
TCP:
HTTP:  ----- HTTP:   -----
HTTP:
HTTP:  ""
HTTP:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 2 arrived at 13:46:23.49
ETHER:  Packet size = 82 bytes
ETHER:  Destination = 8:0:20:1d:21:e0, Sun
ETHER:  Source      = 8:0:20:74:94:c, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 28
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = fe80::a00:20ff:fe74:940c
IPv6:   Destination address = fe80::a00:20ff:fe1d:21e0
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 80
TCP:  Destination port = 33028
TCP:  Sequence number = 1574078520
TCP:  Acknowledgement number = 2912560766
TCP:  Data offset = 28 bytes
TCP:  Flags = 0x12
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..1. = Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0xdceb
TCP:  Urgent pointer = 0
TCP:  Options: (8 bytes)
TCP:    - No operation
TCP:    - No operation
TCP:    - SACK permitted option
TCP:    - Maximum segment size = 1440 bytes
TCP:
HTTP:  ----- HTTP:   -----
HTTP:
HTTP:  ""
HTTP:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 3 arrived at 13:46:23.49
ETHER:  Packet size = 74 bytes
ETHER:  Destination = 8:0:20:74:94:c, Sun
ETHER:  Source      = 8:0:20:1d:21:e0, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 20
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = fe80::a00:20ff:fe1d:21e0
```

```
IPv6:   Destination address = fe80::a00:20ff:fe74:940c
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 33028
TCP:  Destination port = 80 (HTTP)
TCP:  Sequence number = 2912560766
TCP:  Acknowledgement number = 1574078521
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0x099c
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
HTTP:  ----- HTTP:   -----
HTTP:
HTTP:  ""
HTTP:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 4 arrived at 13:46:23.49
ETHER:  Packet size = 720 bytes
ETHER:  Destination = 8:0:20:74:94:c, Sun
ETHER:  Source      = 8:0:20:1d:21:e0, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 666
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = fe80::a00:20ff:fe1d:21e0
IPv6:   Destination address = fe80::a00:20ff:fe74:940c
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 33028
TCP:  Destination port = 80 (HTTP)
TCP:  Sequence number = 2912560766
TCP:  Acknowledgement number = 1574078521
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0xe05e
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
HTTP: ----- HyperText Transfer Protocol -----
HTTP:
HTTP: GET /allans/about.htm HTTP/1.1
HTTP: Host: [fe80::a00:20ff:fe74:940c]
HTTP: User-Agent: Mozilla/5.0 (X11; U; SunOS sun4m; en-US; rv:1.0.1) Gecko/20020920
Netscape/7.0
HTTP: Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-
mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1
HTTP: Accept-Language: en-us, en;q=0.50
HTTP: Accept-Encoding: gzip, deflate, compress;q=0.9
HTTP: Accept-Charset: ISO-8859-1, utf-8;q=0.66, *;q=0.66
HTTP: Keep-Alive: 300
```

```
HTTP: Connection: keep-alive
HTTP: Referer: http://[fe80::a00:20ff:fe74:940c]/allans/leftframe.htm
HTTP: If-Modified-Since: Mon, 10 Mar 2003 22:41:46 GMT
HTTP: If-None-Match: "1d0dc-1295-f08efe80"
HTTP:
HTTP:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 5 arrived at 13:46:23.49
ETHER:  Packet size = 74 bytes
ETHER:  Destination = 8:0:20:1d:21:e0, Sun
ETHER:  Source      = 8:0:20:74:94:c, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 20
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = fe80::a00:20ff:fe74:940c
IPv6:   Destination address = fe80::a00:20ff:fe1d:21e0
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 80
TCP:  Destination port = 33028
TCP:  Sequence number = 1574078521
TCP:  Acknowledgement number = 2912561412
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0x0716
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
HTTP:  ----- HTTP:   -----
HTTP:
HTTP:  ""
HTTP:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 6 arrived at 13:46:23.55
ETHER:  Packet size = 256 bytes
ETHER:  Destination = 8:0:20:1d:21:e0, Sun
ETHER:  Source      = 8:0:20:74:94:c, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 202
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = fe80::a00:20ff:fe74:940c
IPv6:   Destination address = fe80::a00:20ff:fe1d:21e0
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 80
TCP:  Destination port = 33028
TCP:  Sequence number = 1574078521
TCP:  Acknowledgement number = 2912561412
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x18
```

```
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0x20d5
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
HTTP: ----- HyperText Transfer Protocol -----
HTTP:
HTTP: HTTP/1.1 304 Not Modified
HTTP: Date: Wed, 26 Mar 2003 13:46:23 GMT
HTTP: Server: Apache/2.0.44 (Unix)
HTTP: Connection: Keep-Alive
HTTP: Keep-Alive: timeout=15, max=100
HTTP: ETag: "1d0dc-1295-f08efe80"
HTTP:
HTTP:

ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 7 arrived at 13:46:23.55
ETHER:  Packet size = 74 bytes
ETHER:  Destination = 8:0:20:74:94:c, Sun
ETHER:  Source      = 8:0:20:1d:21:e0, Sun
ETHER:  Ethertype = 86DD (IPv6)
ETHER:
IPv6:   ----- IPv6 Header -----
IPv6:
IPv6:   Version = 6
IPv6:   Traffic Class = 0
IPv6:   Flow label = 0x0
IPv6:   Payload length = 20
IPv6:   Next Header = 6 (TCP)
IPv6:   Hop Limit = 60
IPv6:   Source address = fe80::a00:20ff:fe1d:21e0
IPv6:   Destination address = fe80::a00:20ff:fe74:940c
IPv6:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 33028
TCP:  Destination port = 80 (HTTP)
TCP:  Sequence number = 2912561412
TCP:  Acknowledgement number = 1574078703
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 25920
TCP:  Checksum = 0x0660
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
HTTP:  ----- HTTP:   -----
HTTP:
HTTP:  ""
HTTP:
```

This is also shown as a Hex decode below:

```
14:53:04.276529 fe80::a00:20ff:fe1d:21e0.33032 > fe80::a00:20ff:fe74:940c.80: S
3010780077:3010780077(0) win 25920 <nop,nop,sackOK,mss 1440>
                        00 20ff fe1d 21e0 fe80 0000 0000 0000
                        0a00 20ff fe74 940c 8108 0050 b374 dbad
                        0000 0000 7002 6540 07f9 0000 0101 0402
                        0204 05a0
14:53:04.276722 fe80::a00:20ff:fe74:940c.80 > fe80::a00:20ff:fe1d:21e0.33032: S
1682397960:1682397960(0) ack 3010780078 win 25920 <nop,nop,sackOK,mss 1440>
```

```
                        6000 0000 001c 063c fe80 0000 0000 0000
                        0a00 20ff fe74 940c fe80 0000 0000 0000
                        0a00 20ff fe1d 21e0 0050 8108 6447 5b08
                        b374 dbae 7012 6540 4898 0000 0101 0402
                        0204 05a0
14:53:04.278195 fe80::a00:20ff:fe1d:21e0.33032 > fe80::a00:20ff:fe74:940c.80: . ack 1682397961
win 25920
                        6000 0000 0014 063c fe80 0000 0000 0000
                        0a00 20ff fe1d 21e0 fe80 0000 0000 0000
                        0a00 20ff fe74 940c 8108 0050 b374 dbae
                        6447 5b09 5010 6540 7548 0000
14:53:04.281052 fe80::a00:20ff:fe1d:21e0.33032 > fe80::a00:20ff:fe74:940c.80: P 0:644(644) ack
1 win 25920
                        6000 0000 0298 063c fe80 0000 0000 0000
                        0a00 20ff fe1d 21e0 fe80 0000 0000 0000
                        0a00 20ff fe74 940c 8108 0050 b374 dbae
                        6447 5b09 5018 6540 0fcc 0000 4745 5420
                        2f61 6c6c 616e 732f 686f 6d65 2e68 746d
                        2048
14:53:04.282213 fe80::a00:20ff:fe74:940c.80 > fe80::a00:20ff:fe1d:21e0.33032: . ack 645 win
25920
                        6000 0000 0014 063c fe80 0000 0000 0000
                        0a00 20ff fe74 940c fe80 0000 0000 0000
                        0a00 20ff fe1d 21e0 0050 8108 6447 5b09
                        b374 de32 5010 6540 72c4 0000
14:53:04.287290 fe80::a00:20ff:fe74:940c.80 > fe80::a00:20ff:fe1d:21e0.33032: P 1:182(181) ack
645 win 25920
                        6000 0000 00c9 063c fe80 0000 0000 0000
                        0a00 20ff fe74 940c fe80 0000 0000 0000
                        0a00 20ff fe1d 21e0 0050 8108 6447 5b09
                        b374 de32 5018 6540 33e9 0000 4854 5450
                        2f31 2e31 2033 3034 204e 6f74 204d 6f64
                        6966
14:53:04.289460 fe80::a00:20ff:fe1d:21e0.33032 > fe80::a00:20ff:fe74:940c.80: . ack 182 win
25920
                        6000 0000 0014 063c fe80 0000 0000 0000
                        0a00 20ff fe1d 21e0 fe80 0000 0000 0000
                        0a00 20ff fe74 940c 8108 0050 b374 de32
                        6447 5bbe 5010 6540 720f 0000
```

Appendix D – Quick Reference Commands

DNS:

host –t aaaa hostname

IP address config

(Solaris) ifconfig –a
(Windows) ipconfig /all

Ping:

(Solaris) ping -A inet6 -a hostname
(Windows) ping6 hostname

Routing Table

netstat -rn

Snoop:

snoop -vr ip6 host hostname

Tcpdump:

tcpdump -S -s96 -tt -n -x ipv6 host hostname

-S	shows absolute TCP sequence numbers instead of relative
-s96	capture first 96 Bytes of data - allows better analysis if optional headers are used
-tt	shows time unformatted
-n	doesn't resolve host names
-x	print packet in hex

Appendix E – ERG IPv6 Addresses

Machine Name	IPv4 Address	Site-local IPv6 Address	Global IPv6 Address
blake	139.133.210.30	fe80:: a00:20ff:fe1b:2ece	2001:630:241:4:a00:20ff:fe1b:2ece
mavis	139.133.204.77	fe80:: a00:20ff:fe86:ecdf	2001:630:241:0:a00:20ff:fe86:ecdf
door-ipv6*	139.133.204.10 139.133.210.10	FE80::207:85FF:FE60:3BA1 FE80::207:85FF:FE60:3BA0	2001:630:241:0:207:85FF:FE60:3BA1 2001:630:241:4:207:85FF:FE60:3BA0
endeavour-ipv6*	139.133.204.100	fe80:: a00:20ff:fec1:19c9	2001:630:241:0:a00:20ff:fec1:19c9
genesis-ipv6*	139.133.204.128	fe80::a00:20ff:fe74:940c	2001:630:241:0:a00:20ff:fe74:940c
blade-ipv6*	139.133.204.154	fe80:: 203:baff:fe09:87c	2001:630:241:0:203:baff:fe09:87c
churchward-ipv6*	139.133.204.110	fe80:: a00:20ff:fe86:354b	2001:630:241:0:a00:20ff:fe86:354b

* To obtain the IPv4 address, remove the suffix –ipv6

Appendix F – ICMPv6 codes

The table below lists a number of error messages and corresponding codes used in ICMPv6.

Message Number	Message Type	Code Field
1	Destination Unreachable	0 = no route to destination 1= communication with destination administratively prohibited 2 = beyond scope of source address 3 = address unreachable 4 = port unreachable
2	Packet Too Big	Code field set to 0 by the sender ignored by the receiver
3	Time Exceeded	0 = hop limit exceeded in transit 1 = fragment reassembly time exceeded
4	Parameter Problem	0 = erroneous header field encountered 1 = unrecognised next header type 2 = unrecognised IPv6 option encountered

The informational messages and codes can be seen in the table below:

Message Number	Message Type	Code Field
128	Echo Request	RFC 2463. Both used for the ping command
129	Echo Reply	
130	Multicast Listener Query	RFC 2710. Used for multicast group management (IPv4 uses IGMP for this functionality)
131	Multicast Listener Response	
132	Multicast Listener Done	
133	Router Solicitation	RFC 2461. Used for neighbour discovery and autoconfiguration
134	Router Advertisement	
135	Neighbour Solicitation	
136	Neighbour Advertisement	
137	Redirect Message	
138	Router Renumbering	RFC 2894
139	ICMP Node Info Query	
140	ICMP Node Info Response	
141	Inverse ND Solicitation	RFC 3122
142	Inverse ND Adv Message	
150	ICMP Home Agent Address Discovery Request Message	Experimental messages for mobile IPv6
151	ICMP Home Agent Address Discovery Reply Message	
152	ICMP Mobile Prefix Solicitation Message Format	
153	ICMP Mobile Prefix Advertisement Message Format	