Passive fingerprinting on an IPv6-enabled network

Koen Zandberg
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
koen@zandberg.student.utwente.nl

ABSTRACT

A lot of information can be gathered from a local broadcast domain by passively capturing packets. How much information can be gathered from IPv6 enabled hosts and whether this is more than over IPv4 is not clear. Often only IPv4 is regarded, or only ARP or NDP packets are used when passively monitoring a network for active hosts. It is also possible to gather information about hosts by looking at data received from other broadcast or multicast protocols such as mDNS. In this paper, data from passive host measurements on a single subnet is analysed and a comparison between information found on IPv4 and IPv6 is made, and the uptime of hosts is estimated based on this data. With that, it becomes clear that IPv6 does not leak more information about a host than IPv4.

Keywords

IPv6, fingerprinting, local network, security

1. INTRODUCTION

We bring more and more portable devices with us. Almost everyone today carries a phone, a tablet and/or a notebook. These devices often require a network connection. With this in mind, most places have some kind of wireless network available for the devices employees or customers bring with them. Although this is sometimes implemented as a separate network, isolated from other local network resources, these unknown devices can pose a security risk for the site and for the other devices. The challenges created by these devices are numerous and are not only security related. The increase in devices also brings challenges for the network infrastructure.

The need for a larger address space can be resolved by deploying IPv6. While IPv6 was, among other things, designed for the need of a larger network with more devices, other issues it tries to solve is the possibility to communicate securely with each other. These measures are often not used[12]. Often for administrators the deployment of IPv6 often brings new and unknown challenges in securing the local network[22]. Two of the issues are the security of the clients on the IPv6-enabled network and insight in the clients connected to the network.

The first issue is the possibility of clients monitoring each other. By utilizing the possibilities enabled by connection to the same broadcast domain, information about other connected devices can be gathered. A comparison of IPv4 and IPv6 and whether a client is more vulnerable for this on an IPv6 enabled network can be made. This gives an estimate about risks of IPv6 and whether deployment of IPv6 creates additional security risks for clients of the network. These risks might include the information gathering of other connected clients or spoofing information to other clients.

The second issue is an overview of connected hosts. This is useful for local administrators among other things because of the insight it gives into the utilization of the network. An overview of the local network could be generated by actively probing each network address, but with the recommended subnet size of 64 bit, this is with IPv6 no longer feasible. Instead of active probing of addresses, a more practical way is listening for packets already sent out by those hosts. By using the existing traffic that connected hosts emit, it is possible to collect information about these hosts. Each host connected to a network will send a significant number of packets as multicast or broadcast. For example, NDP [18] for discovery of hosts. Addresses from these hosts can be learned by listening for the Router Solicitation packets and from the Duplicate Address Detection packets.

This paper proposes a fully passive way to detect hosts and estimate the uptime of these hosts using packets emitted by those hosts. An analysis is made of the possible information gathered from hosts. Furthermore the amount of this information gathered passively from hosts is compared between IPv4 and IPv6. In Section 2 the theory behind the used packets and protocols is explained. In Section 3 a number of other related works are discussed. Section 4 describes the measurement network and analysis methods. The results gathered from the proposed methods are in Section 5 and discussed in Section 6 and conclusions are finally drawn in Section 7. A final recommendation based on the results is given in Section 8

2. BACKGROUND ON IPV6

2.1 IPv6

This paper looks at local traffic caused by IPv6. IPv6 tries to solve a number of problems of IPv4. The main problem solved is the number of unique available addresses. The address size of IPv6 is raised to 128 bit from the 32 bit of IPv4. With IPv6 there is also ICMPv6 which provides a lot of the same tasks as ICMP in IPv4 does. With the larger address space, the IPv6 header is also larger, displayed in Figure 1. In this paper a number of other properties of IPv6 are used. IPv6 does not use broadcast traffic. Where in IPv4 a lot of traffic needed for network operation was broadcasted over the subnet, with IPv6 the traffic is sent to multicast addresses. This has the advantage that only the parties interested in the traffic receive it. For example, with DHCP in IPv4, the discovery is broadcasted over the subnet. In IPv6 a *local all DHCP servers* address is defined. Only DHCPv6 servers should subscribe to this address, so only the DHCPv6 servers receive the DHCP requests.

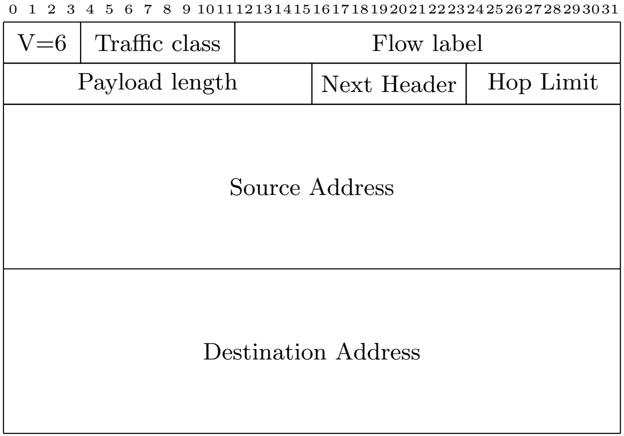


Figure 1: IPv6 header specification

IPv6 has different approaches to address configuration. *Stateless address autoconfiguration* is a commonly used way to generate a unique per host address without the need of a central administrative entity. Router advertisements are sent out periodically with the information about the subnet. Information contained in these advertisements are the network prefix, the local gateway and lifetime information. Optionally DNS information can also be included. Hosts generate their addresses based on their MAC address and the network prefix and ensure the uniqueness of their address with neighbor solicitations. Thus, the first 64 bit of the address is the local prefix of the subnet, announced via the router advertisements, and the last 64 bit are based on the IEEE identifier. With Ethernet that is the MAC address of the network interface. For example, with a MAC address of "00:16:3e:14:91:83" and a network prefix of "2001:67c:2564:a120::/64", the SLAAC address is "2001:67c:2564:a120:0216:3eff:fe14:9183".

A privacy related problem arises when using addresses based on the MAC address of a device. If the last 64 bit of the address of a device is always the same and since a MAC address should be globally unique, a device can be traced between networks based on the last 64 bit of the IPv6 address.

Privacy extensions were defined for IPv6 SLAAC to solve this issue. If the interface identifier, which the address is based of, is randomly generated each time the device connects to a network, the identifying factor of the hosts is lost. With 64 bits of possibilities, the chance for collisions is small, but uniqueness is again verified with neighbor solicitations. RFC 4941[19] states that when generating a random interface this interface identifier should be used for each prefix advertised, minimizing the required multicast subscriptions. It does however not require this, it is allowed to have a different generated interface identifier per prefix.

RFC 7217[9] proposes a method to generate an identifier unique for each prefix, but one that is not changed over time. This way stable addresses are generated per prefix, but the privacy of the user is retained by generating different addresses per network prefix. Thus if the device moves between networks, the associated address changes.

2.2 Multicast

Including IPv6 ICMP messages, a lot of the protocols on local networks make use of multicast[15], making it an essential part of IPv6. Multicast differs from broadcast traffic in that only the hosts that are subscribed to the address should receive the traffic. This prevents hosts from receiving unwanted traffic. This also requires that if traffic is to be received, a multicast subscription is needed for the destination addresses of that traffic. For the handling of multicast subscriptions, MLD[24] is used with IPv6 and IGMP[5] is used with IPv4.

Table 1: Multicast addresses and their designation

IPv6 Address	IPv4 equivalent	Designation
ff02::1	224.0.0.1	All nodes
ff02::2	224.0.0.2	All routers
ff02::1:2	-	All DHCP servers
ff0x::fb	224.0.0.251	MDNS
ff02::1:ffxx:xxxx	-	Solicited node multicast address

Each multicast aware network switch keeps track of the subscriptions of the attached devices. Multicast traffic is only forwarded to network interfaces with a subscription on the specific traffic. This is called IGMP snooping or MLD snooping, depending on which type of traffic is handled. If no multicast aware network devices are present in the network, IPv6 multicast is essentially behaving as if it was broadcast.

In IPv6 a number of multicast addresses are defined. Table 1 lists IPv6 multicast addresses and their IPv4 counterpart relevant for this research. The *Solicited node multicast address* is special in that it is formed from the last 24 bits of the host unicast address. This way, a node specific multicast address is formed, enabling packets to be sent to at least the target host without knowing the MAC address of that host.

Multicast tries to ensure that only parties interested in the traffic receive the traffic. With the proper equipment in place, this reduces network traffic to the end user. Although multicast can prevent hosts from receiving certain traffic, it is by no means a security measurement. Hosts can subscribe to any multicast group they want. Multicast snooping thus makes it a bit harder to receive all traffic, but it does not increase the security of the network.

2.3 NDP

NDP[18] is similar to ARP[20] in that the protocol is used to discover the MAC-addresses of devices. Where ARP is contained on an own protocol layer, NDP packets are a type of ICMPv6 packets. There are two types of NDP packets: NDP solicitations, used to query for the MAC address of a host, and the NDP advertisement, used as a response to the solicitations. NDP solicitations are sent to the *solicited-node multicast address* [14] or the *all nodes multicast address*.

Duplicate Address Detection is also done with NDP, this to verify the uniqueness of the configured address. When NDP solicitation is used for duplicate address detection, the source IP should be set to the unspecified address instead of the actual source address.

2.4 MDNS

MDNS, or multicast DNS, is an enhanced version of DNS suitable for service discovery on local area networks. Multicast DNS works by announcing services as DNS records on the local network. Information contained in these records often span the network addresses of the host and the ports that are used by services provided on that host. Sometimes records also contain device specific information such as firmware versions. With MDNS all packets, the queries and the responses, are sent to the multicast MDNS address.

3. RELATED WORK

A number of tools use a passive approach to network enumeration. Both for IPv4 and IPv6 there are tools available to discover hosts on a local subnet. THC-ipv6's detect-new-ip6 [13], NDPMon [4] and SLAACer [7] are tools for detection of IPv6 enabled hosts. All of these tools rely on receiving neighbor solicitation packets. NDPMon's focus is on the detection of invalid or malicious packets. The focus of SLAACer is reporting address bindings of hosts. Neither of these detect when an address is no longer reachable on the network.

From a security perpective a lot of research has been done on local IPv6 enable networks. A comparison of privacy features between IPv4 and IPv6 is available in RFC 4864[23]. They state that the privacy advantages that originate from the boundary formed by the NAT of a local network are also possible with IPv6. Although provided by different methods, the same privacy goals are achieved.

Local vulnerabilities that are possible because of IPv6 enabled hosts are also researched[22]. For example router advertisements can be spoofed by malicious clients. Most of these vulnerabilities are mitigated on the network infrastructure by implementing features such as RA-guard[17]. This methods works, although if not implemented properly it can still be evaded [10]. A lot of these possibilities to evade protective measurements come from the IPv6 header extensions. The extension headers are not constant in length and the start of a payload is thus not at a fixed number of octets. For embedded devices it is often not possible to do a deeper packet analysis or reassemble fragments for decision making. Another way to evade Intrusion detection systems was by using overlapping fragments in a fragmented payload[3]. Although instruction on how to handle these packets is available [16], most popular operating systems did not follow this.

Although a direct practical comparison between functionality of IPv4 and IPv6 on a single network was not available, functional differences arise when looking at a network with only IPv6 connectivity announced[2]. A network with native IPv6 connectivity and IPv4 connectivity via transition methods was built. The experience from such network states that most problems are with applications that are unable to connect over IPv6 only, sometimes because of the use of IPv4 literals.

4. APPROACH

The base of the research consists of the analysis of traffic of a large campus network. The traffic analysed consists of packets that are sniffed in a passive way from the network, and use multicast or broadcast addresses. The packets are used for two purposes. First the analysis of the packets for host identification. The headers as well as the content of the packets are used for analysis of the hosts sending them. Second, the uptime of hosts is tracked by the time between subsequent packets from a single host.

4.1 Measurement network

The network used for the research is an actively used campus network. It is a single subnet, a /20 subnet on IPv4 and /64 on IPv6, with a large number of different hosts (more than 2400 unique MAC-addresses) actively using the network for both local and remote resources. On the network, IGMP-snooping is enabled but MLD-snooping is disabled due to practical problems. Among the operating systems running on the hosts are the popular operating systems such as Linux, Windows and OS-X. Within this network, every device needs to be registered for Internet access. There are two possible registrations: one with a fixed IP address and a DNS entry for the IP address and a dynamic registration where the IPv4 address is dynamic with a generic IP based DNS entry for the host. On the network SLAAC is used for the assignment of IPv6 addresses. Two IPv6 network prefixes are announced. One legacy range and a new range. Both ranges are announced with a preferred lifetime of 604800 seconds (7 days) and a valid lifetime of 2592000 seconds (30 days).

A topological overview is given in Figure 2. This overview is a schematic overview: the cloud with the switch represents the whole network of switches connecting all hosts. Although depicted as a separate interface, the monitoring server is connected to an interface that could also be used for a regular host. No access to the router or switches is available nor necessary. This due to the fact that all measurements happen passively on the network without any privileged port or port mirroring setup. All information gathered is representative for information that can be gathered by anybody connected to the network. Packets captured are the same packets received by other attached devices. Only multicast and broadcast traffic is received, unicast traffic is not received.

The traffic is captured from the network in a passive way and without intercepting any traffic going through the router. The topological location of the monitoring interface does not matter and the interface does not need any different configuration than any other host on the subnet. This interface is configured such that it should not sent out any data except for the necessary things to maintain a connection to the network. For example, IGMP and MLD packets to ensure that we can also capture multicast traffic if we are dealing with multicast aware network hardware.

Because multicast snooping on IPv6 is disabled in the network, no effort has to be made to receive IPv6 multicast traffic. With multicast snooping enabled, registration would have to be made for all relevant multicast addresses including the ff02::1:ff00:0/104 range. Whether this is possible with the switch infrastructure in place is not tested.

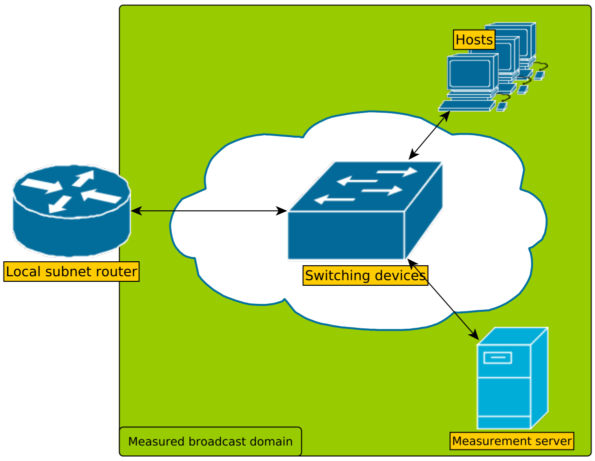


Figure 2: Simplified network diagram of the measurement setup

4.2 Host identification

Identification of hosts is done by analysis of a number of packet types broadcasted by hosts.

4.2.1 *Address enumeration*

The measurement system tracks each host by the MAC-address. It is assumed that this MAC-address is constant. The source of each MAC-address is recorded and used in the measurement system as a host identifier. For each IPv6 address, duplicate address detection needs to be done by the host [21]. Because of this, each configured IPv6 address can be detected by the measurement system. By recording all measured ARP and NDP packets an enumeration of the configured addresses on a host is built.

It is possible to accurately estimate the procedure used for generating privacy addresses. By generating a list of all addresses on a hosts, a number of situations are possible. RFC 4941[19] advises a host to use the same randomly generated identifier for each suffix. This is a key difference with RFC 7217[9] where the identifiers are unique for each prefix. However, RFC 4941 advises, but not enforces the same identifier. It is possible that a host generates a unique for each network prefix. By looking for longer time, the privacy addresses generated according to RFC 4941 should be different over time for a single network prefix, while addresses generated according to RFC 7217 are stable over time.

With this, the detected device can be categorized in a number of categories depending on their behaviour with IPv6 and privacy extensions:

- No IPv6: No IPv6 addresses are detected, the device does not use IPv6
- Link local only: Only IPv6 link local addresses are detected.
- Only MAC-based SLAAC: Only addresses based on the 48 bit MAC address is detected
- RFC 4941: Multiple privacy addresses are detected, Each Identifier is used for both prefixes.
- RFC 4941 with unique identifiers per prefix: Multiple privacy addresses are detected, unique privacy addresses for each prefix
- RFC 7217: Only two privacy addresses are detected, one for each prefix.
- Undefined: No determining factor could be identified

The No IPv6 category contains all host for which no IPv6 address was detected. These hosts only have IPv4 connectivity. The Link local only category have an address within the fe80::/64 range. No other IPv6 addresses are detected. The category of Only MAC-based SLAAC are the hosts of which in addition to the link local address an IPv6 address based on the MAC address is detected. If at least one address is detected that contains `ff:fe` in the address and no privacy extension addresses are found, the host is placed in this category. RFC 4941 is used when multiple privacy addresses are detected, but at least one address has for both prefixes an identical local part. If more than two unique privacy addresses are detected, but there is never a pair of privacy addresses with identical local part and different prefix, the host is categorized as RFC 4941 with unique identifiers per prefix. If only two privacy addresses are detected and not with an identical local part, the host is categorized as RFC 7217. If none of these categories match, the host is placed in the Undefined category, this because there is no determining factor of which method is used, for example, when only a single privacy address is detected, it is not possible to be able to determine the method of address generation. Incorrect estimation of hosts as RFC 7217 is possible when the host uses unique identifiers per prefix, but generates them according to RFC 4941. This only happens if the host is not online long enough to generate multiple identifiers.

4.2.2 Service Discovery

Multicast DNS (MDNS) [6] traffic is used for service discovery because initial measurements show that a lot of discovered host use this as a way to announce provided services. Multicast DNS is used to announce the hostname and services provided by the announcing host. The announced hostname has ".local" as a domain. Since this hostname is the hostname configured by the end user on the device, it is not related to the hostname contained within the regular DNS information. The hostname being chosen by the end user and influenced by the operating system gives information about the operating system. For example, OS X defaults to a hostname beginning with the device type.

4.3 Uptime analysis

An application was built in order to analyse the received traffic in real time. A real time application is needed because the verification of the hosts needs to be done at the moment a host is suspected to go offline. The application keeps track of the hosts that are turned on and connected to the network, the packets of the individual hosts are analyzed and from this the time that the host is shut off is measured for verification.

The moment a host becomes active on the network is possible to pinpoint due to the fact the is is required for a host to send a neighbor solicitation packet for duplicate address detection when configuring a network interface.

Estimated is that each host generates packets within a certain time in between. From the measured time between the packet arrival, the packet interarrival time, a cumulative density function is build. From this density function, an interarrival time is chosen such that a fixed percentage of packet interarrival time is below this time. It is assumed that after a multiple of this estimated time out has passed without receiving any packets from a host, this host is not active on the network anymore. The lasts 1000 packet interarrivals measured from an address are stored in a database for the real time uptime estimation.

```
if len(delays) > 20:
    delays.sort()
    measure_point = math.floor(
        len( delays ) * cutoff_point
    )
    delta_t = delays[measure_point]
    if delta_t < 3:
        delta_t = 3
    return delta_t
```

Calculations are only performed on a host when 20 or more interarrival times are gathered. This prevents hosts that connect only shortly from influencing the measurements. With IPv4 and IPv6, around two to five neighbor solicitations are expected from a single connecting host depending on the privacy extensions used. As an interarrival time, 20 times was chosen as a bottom limit so that calculations are done on representative traffic of the host and not only on a short burst of traffic for the duplicate address detection. The calculated timeout is limited such that a minimum time out of 3 seconds is used if it is calculated to be below 3 seconds. This to prevent false positives from busy hosts such as gateway routers. It also prevents accidental flooding of hosts with ARP requests.

Due to the size of the broadcast domain and limited computational resources only a subset of the maximum of 4094 hosts are measured in real time. MAC addresses are filtered on the last hexadecimal number of their address to be able to restrict the number of packets that need to be processed while still receiving the complete overview of packets of a single host. The restriction is based on the last hexadecimal character because this is not based on the vendor of the MAC address, ruling out any chance that a certain vendor is overly represented on the network. Furthermore, the last hexadecimal character should differ even with hardware from the same batch, ruling out any chance that a large group of the same hardware is selected by accident. From a unfiltered total of 2445 MAC addresses, groups were made based on the last hexadecimal character of the address. The average group size was 152.8, with a standard deviation of 14.7. The largest group was group ‘F’, with 187 Addresses and the smallest was group ‘A’ with 134 addresses. Altogether, most groups were between 140 and 160 addresses

4.3.1 Verification

For verification of the estimated uptime of the host an active probing method is used. This verfication is only done for the case that a host is suspected to be disconnected. The moment a host is connected can be accuratly pinpointed because the host starts sending traffic. Probing was done at every `cutoff_point` and the result is recorded. For example, if a host has a calculated `cutoff_point` of 10 seconds, every 10 seconds the host is probed and the result is stored. If after a number of probes the host sends new traffic unrelated to the probes, the probing stops and restarts after the new `cutoff_point` is reached.

Probing was done with ARP requests at the IPv4 address of the host. This is used for a few reasons. ICMP ping echo requests can be dropped in the firewall. For example, recent versions of Microsoft Windows drop ICMP echo requests when the network is configured other than “home network”. ARP requests were chosen over neighbor solicitations because almost every host on the network has an IPv4 address whereas a significant number of hosts does not use IPv6. This simplified verification in that only a single tool is nessecary and reduces variation due to different verification tools used between hosts. Furthermore, the IPv4 address of a host stays constant over a single connection period where some of the IPv6 addresses vary due to privacy extensions. There were also practical problems with the neighbor solicitations, when many parallel solicitations are sent errors occur on the measurement platform. With ARP requests no problems have been experienced. For sending ARP requests, the tool “arping”[11] is used and the output is parsed. The neighbor solicitations are tried with “ndisc6”[8].

5. RESULTS

5.1 Host identification

Passive host identification of the devices on the network spans device vendor, configured addresses, hostname and an estimation of services and operating system.

5.1.1 Vendor

With the source MAC address captured in the packets the vendor of the device can be looked up. This information can be used to guess the type of device. For example, a device with a MAC address of a network vendor gives a good estimation that the device is some sort of network device, while a MAC address of a printer manufacturer indicates that the device likely is a network connected printer.

Table 2: Top ten vendors

Vendor	Amount
ASUSTeK	282
Hewlett-Packard	243
Dell	202
Apple	154
Wistron	137
TP-LINK	131
Micro-Star International	130
ASRock	78
Gigabyte Technology	76
Sitecom	71

Table 3: Address classifications

Address classification	Number of hosts	% of hosts
No IPv6	531	22%
IPv6 Link local only	320	13%
SLAAC addresses	189	8%
RFC 4941	1047	43%
RFC 4941 with unique identifiers per prefix	94	4%
RFC 7217	31	1%
Undefined	199	8%
Failed Configuration	34	1%
Total hosts	2445	100%

Although the top ten device manufacturers consist mostly of consumer and office computer manufacturers, among these vendors are also manufacturers of network equipment such as TP-LINK and Sitecom.

5.1.2 Address enumeration

Analysis of the network addresses was done over a span of seven days. No filtering of the MAC addresses was done with this measurement.

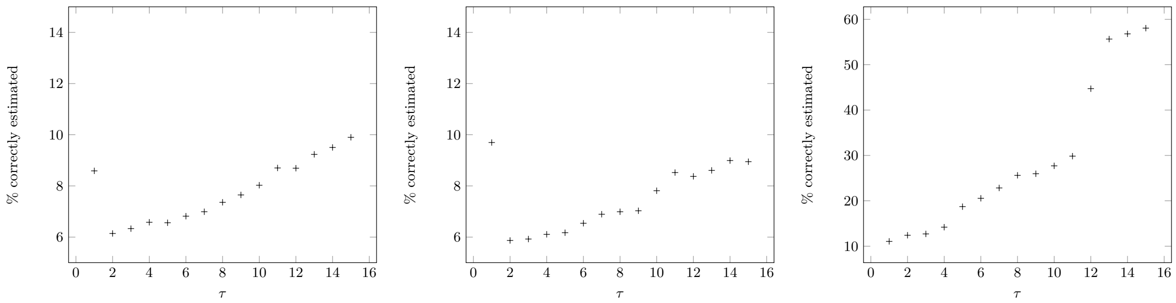
A total of 2445 MAC addresses were measured and a total of 21515 IPv4 or IPv6 addresses were measured. Classification of the configurations is in Table 3. In this table, two additional categories are shown besides the categories from Section 4: failed configuration, which consist of hosts that showed an RFC 1918 address and undefined, which consist of devices of which it was not clear in which category they should belong.

A number of devices show no usage of privacy extensions. Closer inspection by MAC vendor showed that a lot of these devices are likely to be routers, approximately 64 devices of the 189 SLAAC only devices were from network equipment manufacturers such as TP-LINK and Netgear. These devices are fixed, they do not move between different networks. The goal of privacy extensions, namely hiding the momevent of the device between networks, is not needed with these devices.

Another remark from this data is that the devices that generate unique privacy addresses for each prefix are mainly Apple allocated MAC addresses. Of the 94 devices that showed this behaviour, 74 were Apple devices. As metioned before, this is not a violation of the RFC specification.

Of the 31 devices that are detected as RFC 7217, 12 are Apple devices. It is possible that these are wrongly detected as RFC 7217 because of their setting to generate a unique privacy address for each prefix.

Of the 531 device of which no IPv6 traffic was observed,



(a) cutoff_point at 90%

(b) cutoff_point at 75%

(c) cutoff_point at a fixed 300 seconds

Figure 3: Percentage correct estimated at various cutoff_points

around 260 were from manufacturers of embedded devices and routers. This group thus largely consists of devices that do not have an IPv6 network stack in their firmware.

One of the things noticeable in the measurements is that some devices did duplicate address detection for a large number of addresses over the measurement time. The largest number detected was a single Apple device with over 600 addresses. These measurements might have been caused by mobile devices that connects via a wifi bridge to the network, connecting often per day when waking up.

Of the devices using IPv6 and having a public reachable IPv6-address, a lot of devices employ some kind of privacy extensions. Of the 1361 devices that have a public reachable address, 1172 have privacy addresses enabled.

5.1.3 Hostname

5.1.4 Service Discovery over IPv6

The MDNS packets also tell us something about the services offered by hosts. As stated in Subsection 4.2.2, the MDNS packets contain information about the services offered by the sending device. There were 229 hosts detected that use MDNS. These were taken from a total of 577 sampled hosts. This group consists of 100 hosts that also used IPv6 for MDNS. Of the hosts that utilized MDNS and had IPv6 connectivity, every hosts also used MDNS over IPv6. Of the records recorded, 3961 were announced over IPv4, where only 1182 were announced over IPv6. When MDNS answers were announced over IPv6 A-records were never included in the answers. When IPv4 was used, AAAA-records and corresponding PTR-records were included. There was also a host present that seemed to re-announce records from other hosts, causing a total of 10331 additional measured records. This host was not included in the measurements.

When comparing the broadcasted hostname from MDNS with the hostname from the central DNS it was found that in only 24.6% of the cases where the device has a fixed DNS entry, the configured hostname matches with the central DNS. In the case of the dynamic entry, the configured hostname never matched. This of course because the DNS name of the address is derived from the IP address whereas the hostname is statically configured by the user.

5.1.5 Operating system

Because each type of operating system has their own preferred mechanism for autoconfiguration, an estimation of the device operating system family can be given. e.g. Windows uses LLMNR[1], while OS X relies on MDNS for hostname and service discovery. Although this gives a hint about the operating system, this is no guarantee. This relies heavily on the default configuration of operating systems to use a set of resolving techniques.

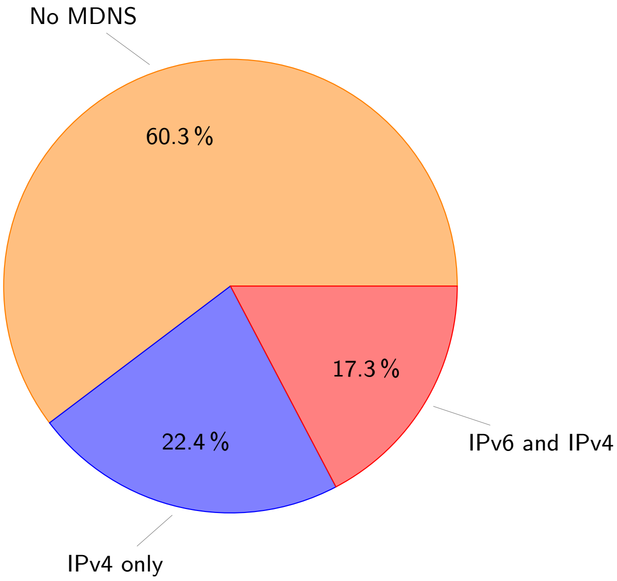


Figure 4: Distribution of MDNS usage

5.2 Host uptime estimation

The measurement system ran for 120 hours. Only MAC-Addresses ending with 0, A, 3 or 8 were measured to limit the number of packets per second that needed processing. For each host measured, a CDF of the packet interarrival time was calculated. The point at which 90% of the packet interarrival times are below that point is taken. This value τ is used as a measurement interval for that host.

With this method, the percentage of hosts that were actually down for a value of τ is plotted in Figure 3a. This shows that the chosen method is not accurate. Although a rising slope, even with 15 times the τ value, the chance that the hosts is correctly estimated as shut down is around 10%

With a lower cutoff percentage the correct estimation percentage is even lower as shown in Figure 3b.

The same test is also done with a fixed τ of 300 seconds. This to test if a longer timeout has a significant effect on the correct estimated percentage. The result of this is shown in Figure 3c

Although a better result is achieved, it is still not enough to reliably estimate the uptime of a host. The jump up at $\tau = 12$ cannot be directly explained. This is at 3600 seconds of idle time. It might be due to a default setting of workstations to go into stand-by after an hour of idle time and stop the network connection.

6. DISCUSSION

6.1 Uptime analysis

We have found a sever limitation in the reliability of the uptime analysis. While enumerating hosts is reliable, it was found that the uptime analysis suffers from the long periods of silence maintained by devices. The cause of this is probably due to the caches of the host. As long as a neighbor is reachable and no network errors are experienced, no new neigbor solicitations are sent. Without this activity, the hosts appear inactive and thus as disconnected.

For the passive uptime detection, the assumption that a host emits a constant stream of packets per second is thus not true. Instead of a continuous stream of packets, a better assumption would be that a host sends periodic bursts of traffic. Although this assumption might be better, it seemed that when idle, a device does not have to emit any broadcast or multicast traffic. It might be possible to enhance these measurements by looking at the first few packets announced by a host after a time of silence, if among these first packets is a DHCP discovery or a Duplicate Address Detection packet, the network interface was down in the time between. If not, the host might have just been inactive for the silent time.

6.2 Multicast snooping

On the measured network, IPv6 multicast snooping was disabled on the network switching infrastructure. Because of this, all IPv6 multicast traffic was received by the measurement setup. If multicast snooping would be enabled, some effort would have to be made to continue to receive this traffic. For example, subscriptions would be nessecary for the whole ff02::1:ff00:0000/104. Furthermore a number of other subscriptions would be required such as the ff02::1 and ff02::2 destination addresses. It was not investigated whether this is possible with the switch hardware used.

6.3 Combining data

A lot of the data gathered can be combined for more accurate estimations of the hosts. For example the information gathered from MDNS and information of the MAC address vendor can be used to estimate the type of device in a more reliable maner. Other sources that can be combined is the type of privacy extensions used and if other auto discovery methods are used.

7. CONCLUSION

Comparing information gathering between IPv4 and IPv6, it can be concluded that IPv6 does not expose more information to other devices. While it is possible to gather a lot of information about the hardware of a connected device, this information is not dependant on the use of either IPv6 or IPv4.

An overview of connected hosts can be gathered passively. It is well aided by duplicate address detection mechanism. Due to the need to do duplicate address detection on every configured address, all addresses of a host can be gathered without difficulties. This detection can be hampered by multicast snooping, making it harder, but not impossible to detect hosts.

Using the traffic of hosts to estimate the uptime is not reliable in the tried way. This due to the long perceived silences from hosts in the periods they do not need neighbor information. This makes the detection unreliable enough that it can not be used for detailed measurements. The moment a host is connecting to the network is accurately measurable due to duplicate address detections.

When looking at IPv6 usage on the network, around 80% of the hosts employ some form of IPv6. Instead of measuring the deployment of IPv6 on different networks, this represents the deployment of IPv6-compatible devices on a network. Of the devices not using IPv6, one third are embedded or networking devices. Of the hosts using IPv6, 73% employ some kind of privacy extensions, making this fairly well used. Almost none of these hosts use the new RFC 7217 based address generation. This scheme is thus not yet used on common operating systems.

8. RECOMMENDATION

When looking at these IPv6 measurements, two recommendation can be made. First, although a firewall would still have to be configured for IPv6, IPv6 does not compromise your hosts. Hosts do not emit more information over IPv6 than at IPv4. Second is not to ignore IPv6. With more than three quarters of the devices employing at leas link local addresses, IPv6 is used on a network whether the network itself is configured for it or not. With devices actively trying to deploy IPv6 connectivity, harm can be done if left without any measurements. Deploying IPv6 thus should not be halted for these reasons but actively deployed instead.

References

[1] B. Aboba, L. Esibov, and D. Thaler. Link-local multicast name resolution (LLMNR). RFC (4795). RFC Editor, 2007. URL: `http://www.rfc-editor.org/rfc/rfc4795.txt`.

[2] J. Arkko and A. Keranen. Experiences from an IPv6-Only network, 2012.

[3] A. Atlasis. Attacking IPv6 implementation using fragmentation. *Blackhat europe*, 2012.

[4] F. Beck, J. Mohacsi, and J. Baskwill. NDPMon. [Online; accessed 8-June-2015]. URL: `http://ndpmon.sourceforge.net/`.

[5] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet group management protocol, version 3. RFC (3376). RFC Editor, 2002. URL: `http://www.rfc-editor.org/rfc/rfc3376.txt`.

[6] S. Cheshire and S. Cheshire. Multicast DNS. RFC (6762). RFC Editor, 2013. URL: `http://www.rfc-editor.org/rfc/rfc6762.txt`.

[7] A. Clouter. SLAACer. [Online; accessed 8-June-2015]. URL: `http://www.digriz.org.uk/slaacer`.

[8] R. Denis-Courmont. NDisc6. [Online; accessed 29-Februari-2016]. URL: `http://www.remlab.net/ndisc6/`.

[9] F. Gont. A method for generating semantically opaque interface identifiers with IPv6 stateless address autoconfiguration (SLAAC). (7217), 2014. URL: `http://www.rfc-editor.org/rfc/rfc7217.txt`.

[10] F. Gont. Implementation advice for IPv6 router advertisement guard (ra-guard). RFC (7113). RFC Editor, 2014. URL: `http://www.rfc-editor.org/rfc/rfc7113.txt`.

[11] T. Habets. Arping. [Online; accessed 29-Februari-2016]. URL: `http://www.habets.pp.se/synscan/programs.php?prog=arping`.

[12] L. Hendriks, A. Sperotto, and A. Pras. Characterizing the IPv6 security landscape by large-scale measurements.

[13] M. Heuse. THC IPv6 attack tool kit. [Online; accessed 8-June-2015]. URL: https://www.thc.org/thc-ipv6/.

[14] R. M. Hinden and S. E. Deering. IP version 6 addressing architecture. (4291):15–17, 2006. URL: http://www.rfc-editor.org/rfc/rfc4291.txt.

[15] H. Holbrook, B. Cain, and B. Haberman. Using internet group management protocol version 3 (IGMPv3) and multicast listener discovery protocol version 2 (MLDv2) for source-specific multicast. RFC (4604). RFC Editor, 2006, 1–11. URL: http://www.rfc-editor.org/rfc/rfc4604.txt.

[16] S. Krishnan. Handling of overlapping IPv6 fragments, 2009.

[17] E. Levy-Abegnoli, G. V. d. Velde, C. Popoviciu, and J. Mohacsi. IPv6 Router Advertisement Guard. RFC (6105). RFC Editor, 2011, 1–10. URL: http://www.rfc-editor.org/rfc/rfc1654.txt.

[18] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor discovery for IP version 6 (IPv6). RFC (4861). RFC Editor, 2007, 1–97. URL: http://www.rfc-editor.org/rfc/rfc4861.txt.

[19] T. Narten, R. Draves, and S. Krishnan. Privacy extensions for stateless address autoconfiguration in IPv6. (4941), 2007. URL: http://www.rfc-editor.org/rfc/rfc4941.txt.

[20] D. C. Plummer. Ethernet address resolution protocol: or converting network protocol addresses to 48. bit ethernet address for transmission on ethernet hardware. RFC (826). RFC Editor, 1982. URL: http://www.rfc-editor.org/rfc/rfc826.txt.

[21] S. Thomson, T. Narten, and T. Jinmei. IPv6 stateless address autoconfiguration. RFC (4862). RFC Editor, 1998, 1–30. URL: http://www.rfc-editor.org/rfc/rfc6105.txt.

[22] J. Ullrich, K. Krombholz, H. Hobel, A. Dabrowski, and E. Weippl. IPv6 security: attacks and countermeasures in a nutshell. *Magdeburger journal zur sicherheitsforschung*, 1:514–529, 2015.

[23] G. Van de Velde, T. Hain, R. Droms, B. Carpenter, and E. Klein. Local network protection for IPv6. RFC (4864). RFC Editor, 2007. URL: http://www.rfc-editor.org/rfc/rfc4864.txt.

[24] R. Vida and L. Costa. Internet group management protocol, version 3. RFC (3810). RFC Editor, 2004. URL: http://www.rfc-editor.org/rfc/rfc3810.txt.