Message Agents and IPv6 interoperability problems

Research Project
Universiteit van Amsterdam
System and Network Engineering (MSc)

Class of 2009-2010

Michiel Timmers (michiel.timmers@os3.nl)
Sebastian Carlier (sebastian.carlier@os3.nl)

August 28, 2010

Abstract

This paper presents problems that can occur in the e-mail architecture with a mixed IPv4/IPv6 environment. By using exemplary e-mail architectures the e-mail routing problems between Dual-Stack and IPv4-only or IPv6-only Message Agents are revealed. This proves how cautious one needs to be while configuring e-mail components. Furthermore e-mail client's problems are exposed by testing various Message User Agents, this exposes compatibility problems between those clients and correct IP setups.

Acknowledgments

We would like to thank all the people at SARA[1] for their helpful feedback that we got during this project. Especially we thank Ronald van der Pol and Freek Dijkstra, our project supervisors, for their insight and knowledge of IPv6.

Contents

1 Introduction

The depletion of IPv4 is apparent and at the time of this research (June 2010) the transition to IPv6 still lags behind. When the Internet Assigned Numbers Authority (IANA) and the various Regional Internet Registries (RIRs) run out of IPv4 addresses and only allocation of IPv6 address space is possible we could face connectivity problems if we do not look for any issues beforehand. E-mail based communication is still one of the most important techniques used in electronic communication. The different protocols that are involved in e-mail transport (SMTP, POP3 and IMAP) could give IPv6 interoperability problems if various implementations do not follow standards. This project will look if there are any pitfalls if one would introduce IPv6 on an e-mail architecture.

1.1 General Description of the Project

Different studies have already been done to measure IPv6 problems on web-based protocols[2]. These studies are typically implemented by adding two objects to an IPv4-only web site, one loaded from a Dual-Stack host name, the other from a Single-Stack IPv4 host name. These objects help to see which protocol is in use. You can easily pinpoint any problems by analysing the web-server log and indicate where problems originate (i.e. user client[3], network or operating system[4]). The outcome of these studies has shown that if one would set up an IPv6 AAAA Resource Record (RR) for a web-server there would be a 0.01% drop in user reachability at the moment. Even this small decline in user reachability has held content providers back from publicly announcing AAAA RR [5]. This project is based on different protocols and agents that are involved in e-mail transport. Studies for website traffic have already been done. E-mail is another key component that much of the Internet communication relies on.
Measurements are needed to see where problems originate in order to fix or avoid them. Only then a judgement can be made to see if it is safe to deploy IPv6 on an e-mail infrastructure.

1.2 Prerequisites

IPv6 is not commonly used at this moment and professional network administrators still have limited knowledge on basic IPv6 operation. Moreover sales representatives trying to sell IPv6 as a separate product instead of selling Internet connectivity as a whole add to the confusion. Many research papers, studies and presentations about IPv6 start with an introduction of IPv6 and its basic functioning. If you are reading this research paper and have no basic knowledge about IPv6 you are already behind. Prior knowledge of the following subject is therefore required from the reader:

- IPv4 and IPv6 knowledge
- Knowledge about A and AAAA Resource Records

1.3 Goals and Research Question

This research project is inspired by the work done by Tore Anderson who pointed out problems with IPv6 and web traffic[2]. His study gave insight into

the behaviour of clients' operating systems and web clients when introduced to IPv6. The primary goal of this research is to check if e-mail agents behave according to the corresponding standards and if that behaviour has the desired effect.

The research question for this project is:

Which Message Agents and configurations introduce connectivity problems in an IPv4/IPv6 mixed environment?

1.4 Outline of This Report

Section 2.1 is an overview of different message agents and their behaviour according to the corresponding standards. A small introduction about different IPv6 Transition Mechanisms can be found in Section 2.3. The theoretical effect of Transition Mechanisms is included in the tests in Section 2.4. Section 3 holds various methods that check if Message Agents' behaviour conforms with the corresponding RFCs and if the behaviour is desirable. The paper finishes with best practice and recommendations in Section 4.

The paper includes a lot of abbreviations that may not be familiar to the reader, especially the various Message Agents described in Section 2.1 . To avoid confusion a list of all the abbreviations, that are used throughout this document, is attached at the end of the report.

2 E-mail and IPv6: The Theory

2.1 Message Agents

An e-mail architecture [6] is composed of five main parts, four of which are described in this section. They are as follows:

- Message User Agent (MUA)
- Message Submission Agent (MSA)
- Message Transfer Agent (MTA)
- Message Delivery Agent (MDA)
- Message Store (MS)

Figure 1 shows the relationship between these agents. The figure is divided between the local and Message Handling Service (MHS) parts to describe where each component is located and how they relate to each other. The MHS's role is to transfer the messages from the author to the recipient. In other words it provides an end-to-end transfer service through the MSA, MTA (this component can occur multiple times) and MDA components.

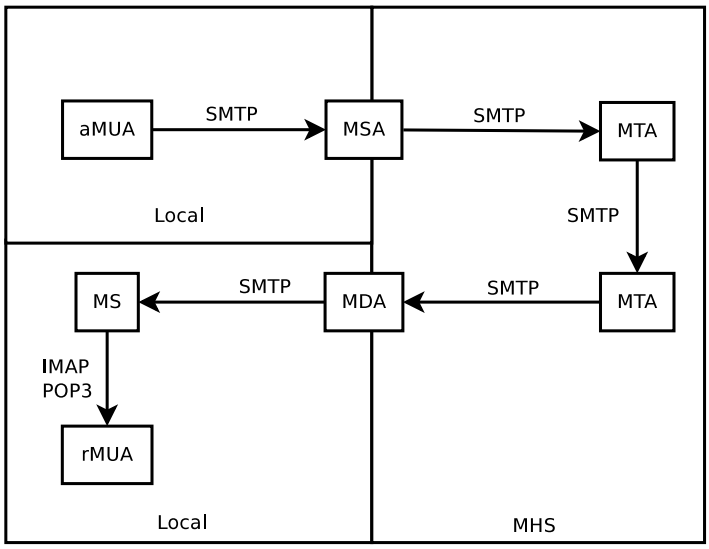

Figure 1: Message Agent relationships as described in RFC 5598.

Message User Agent (MUA)

There are two roles a MUA fulfils: author and recipient. The Author MUA (aMUA) sends the message by using the Simple Mail Transfer Protocol (SMTP) to either an MSA or MS. While performing the Recipient MUA (rMUA) role the rMUA pulls messages from its MS by using the Internet Message Access Protocol (IMAP) or the Post Office Protocol version 3 (POP3). Common MUAs include:

- Microsoft Outlook Express
- Microsoft Outlook
- Mozilla Thunderbird
- Apple Mail

Message Submission Agent (MSA)

MSAs, described in RFC 4409 "Message Submission for Mail" [7], receive messages from MUAs to later transfer them to the appropriate MTAs so that the message can be delivered. MSAs are separate from MTAs (although that was not the case in the past) for the following reasons:

- MSAs require user authentication and authorization, therefore a spammer can be easily identified
- MSAs are able to correct some fields in a message like date, invalid recipient, because they interact with the MUA
- because of the separate MSA the MTA can deny relaying for a non local domain.

The above arguments make it clear why an MSA has been introduced as a separate e-mail architecture component.

Message Transfer Agent (MTA)

An MTA receives messages from an MSA or MTA an forwards them to either another MTA or MDA. It relays the message to another MTA if the recipient is not a local host, adding a *Received:* header to each message. Modifying the *Received:* headers is a way to save the route a message has taken. MTAs use SMTP to send the messages.

Message Delivery Agent (MDA)

An MDA receives the messages from the MTA and places them in the Message Store of the receiving MUA. Commonly used protocols for this transport are POP3 and IMAP.

2.2 Internet Protocol v6 (IPv6)

E-mail transport is a higher layer protocol that should not be affected by routing. However, because of the transition between IPv4 to IPv6 there are two separate IP networks. In addition to that there are also IPv6 Transition Mechanisms (like Teredo, ISATAP and 6to4, to name a few) that can affect on the transport mechanism that in turn could affect higher layer protocols when problems arise. This section briefly describes those protocols and suggests where problems might occur.

IPv6 is not backwards compatible, which greatly slows down migration. At this moment it is impossible to reach all the places on the Internet while only using an IPv6 address. The problem of migration is not a purely technical issue,

it is to some extent caused by a human factor. To be able to reach every place on he Internet with IPv6 only, every server administrator in the world would have to:

- obtain an IPv6 address
- configure the server for IPv6
- announce the IPv6 address in DNS (Domain Name Server)

Obviously, doing that for each server in a given facility is a big expense, without any immediate or short term gain for that facility. The scale and growth of the implementation of IPv4, switching costs and irresponsible administrators were clearly not considered an issue. The solution that could have been implemented is including all the IPv4 addresses in the IPv6 address space. Since that was not done several different solutions, called Transition Mechanisms, have been developed to make IPv6 and IPv4 interoperable.

2.3 Transition Mechanisms

Tunnelling techniques are used when a host with IPv6 wants to connect with another IPv6 enabled host, but part of the route in between is only routable with IPv4. IPv6 packets are encapsulated in IPv4 packets and, if needed, the new IPv4 packets are fragmented at the start of the tunnel. The tunnel has two end-points, either of them can be a host or a router. The tunnel is not transparent for the host, it is treated as one hop by the IPv6 packet; the TTL from the IPv6 header is decremented by one when the packet enters the tunnel. The tunnels can be configured automatically or manually. Manual set up requires the user to specify a point-to-point tunnel. IPv4-encapsulated IPv6 packets are identified by protocol number 41 in the IPv4 header, by this field the end point of the tunnel knows that the packet should be reassembled if need be, stripped of the IPv4 header and sent to the IPv6 destination address.

Teredo

Teredo is a transition mechanism that enables clients with private IPv4 addresses to use IPv6 communication. A Teredo server assigns clients, which can be located behind a NAT (Network Address Translation) device, an IPv6 address, this part of the communication is known as the handshake. The address depends on the type of NAT. The packets are forwarded by a Teredo relay which is located between the native IPv6 and IPv4 networks. Teredo as a client has been introduced in Windows XP Service Pack 1 and is by default enabled in Windows Vista, Windows 7 and Windows Server 2008.

Miredo[8] is an open source equivalent of Teredo designed for Linux and BSD operating systems, but it is not enabled by default in any of the well known distributions.

6to4

6to4 has been introduced for the transition period from IPv4 to IPv6. It allows IPv6 endpoints to communicate through an IPv4 network. As described in the

tunnelling techniques section, IPv4 is used to encapsulate the IPv6 packets at a 6to4 gateway. The gateway is placed on the border of the IPv6 and IPv4 networks. The 6to4 router is required to have a public IPv4 address and can not be placed behind a NAT device.

ISATAP

ISATAP stands for Intra-Site Automatic Tunnel Addressing Protocol. It works on top of IPv4. An IPv4 host configures a virtual IPv6 interface with a link-local address starting with fe80:0000:0000:0000:0000:5efe: followed by his IPv4 address. For example from an IPv4 address 192.168.1.2 the host would get a link-local address fe80::5efe:c0a8:0102. This address is requested by the host from an ISATAP Router. After receiving the address the native IPv4 host with the link-local IPv6 address can communicate with an IPv6-only host through the ISATAP router.
ISATAP does not support multicast, therefore for router discovery, the host needs to be configured with a Potential Router List, or PRL. These lists are obtained by querying DNS (isatap.local.com). This is controversial since the lower layer protocol (ISATAP) relies on a higher level protocol; DNS. Therefore the DNS server needs to be on IPv4.

Selecting the appropriate transport

The choice of the appropriate transport while sending e-mail occurs between each node in the e-mail architecture whenever transport is needed. as depicted in Figure 1 there are numerous relations between various components. Nevertheless IPv6 operability does not suffer from different relations, but from specific software fulfilling one of the described roles. To be able to pinpoint the issues one needs to examine at which point the transport is chosen. The flowchart depicted in Figure 2 represents the SMTP algorithm used by a Dual-Stack SMTP sender. A Dual-Stack algorithm is described, because that is the situation where a choice between IPv4 and IPv6 may occur. Furthermore, IPv6-only environments do not present the problems that occur in a Dual-Stack environment.

The choice of the transport protocol is determined two layers below SMTP, the rules are described by RFC 3484 "Default Address Selection for Internet Protocol version 6 (IPv6)"[9]. Most of the rules formed in the document relate to choosing the address with the appropriately matching prefix between the sender and receiver. The metric table that is used for different addresses is shown below.

Prefix	Precedence	Label
::1/128	50	0
::/0	40	1
2002::/16	30	2
::/96	20	3
::ffff:0:0/96	10	4

Rule 9 in Section 6, for choosing the destination address, states that the addresses should be chosen based on the longest matching prefix between the source and destination. This does not agree with RFC 1794 "DNS Support for

Load Balancing"[10] which points out that reordering the records received from the DNS interferes with load balancing. Even though DNS ordering of the RR is not guaranteed it is easy to predict.

Rule 7 in Section 6 states that native transport should be chosen over a transition mechanism. However when a host does have a private IPv4 address as mentioned in RFC 1918 "Address Allocation for Private Internets"[11], the host is not recognized as a native IPv4 host and the transition mechanism takes preference. Moreover section 10.3 informs that by default IPv6 has preference over IPv4. Rule 9 from section 6 recommends to use the longest matching prefix when comparing the source and destination addresses. This is criticized in draft-arifumi-6man-rfc3484-revise-02 "Things To Be Considered for RFC 3484 Revision"[12] as it interferes with load balancing of services. The behaviour proposed by the draft is to use the Round-Robin technique for load balancing for IPv4 addresses and use rule 9 of section 6 for IPv6 addresses with a prefix longer than 32 bits by default.

2.4 Combining Message Agents and IPv6 according to the standards

RFC 3974 "SMTP Operational Experience in Mixed IPv4/v6 Environments"[13] suggests for every MX record, pointing to a Message Agent to be able to operate with IPv4 and IPv6. Section 3 of RFC 3974 describes the algorithm for a Dual-Stack SMTP sender. The sender looks up MX RRs for the domain the message is addressed to. After it receives the host addresses of the MX RRs it compares the host addresses with the sending client's addresses. If an address matches the algorithm it drops all MX RR with an equal or greater value. Then the algorithm sorts the MX RRs in ascending order. Next the sender queries A and AAAA RR if it is compatible with IPv4 and/or IPv6 respectively. After receiving IPv4 and/or IPv6 addresses the sender tries to establish a TCP connection with one of the received addresses. However RFC 3974 does not state which address to choose first, if more than one is found. It depends on the implementation, with the restriction of not sorting A and AAAA records together and not sorting addresses from different priority MTAs together. It is the most crucial part of the algorithm for this study, yet it is not clearly specified. Various applications can treat the received addresses differently and route the messages in different ways. The flowchart in Figure 2 is a full interpretation of the described algorithm, it also includes errors that might occur.

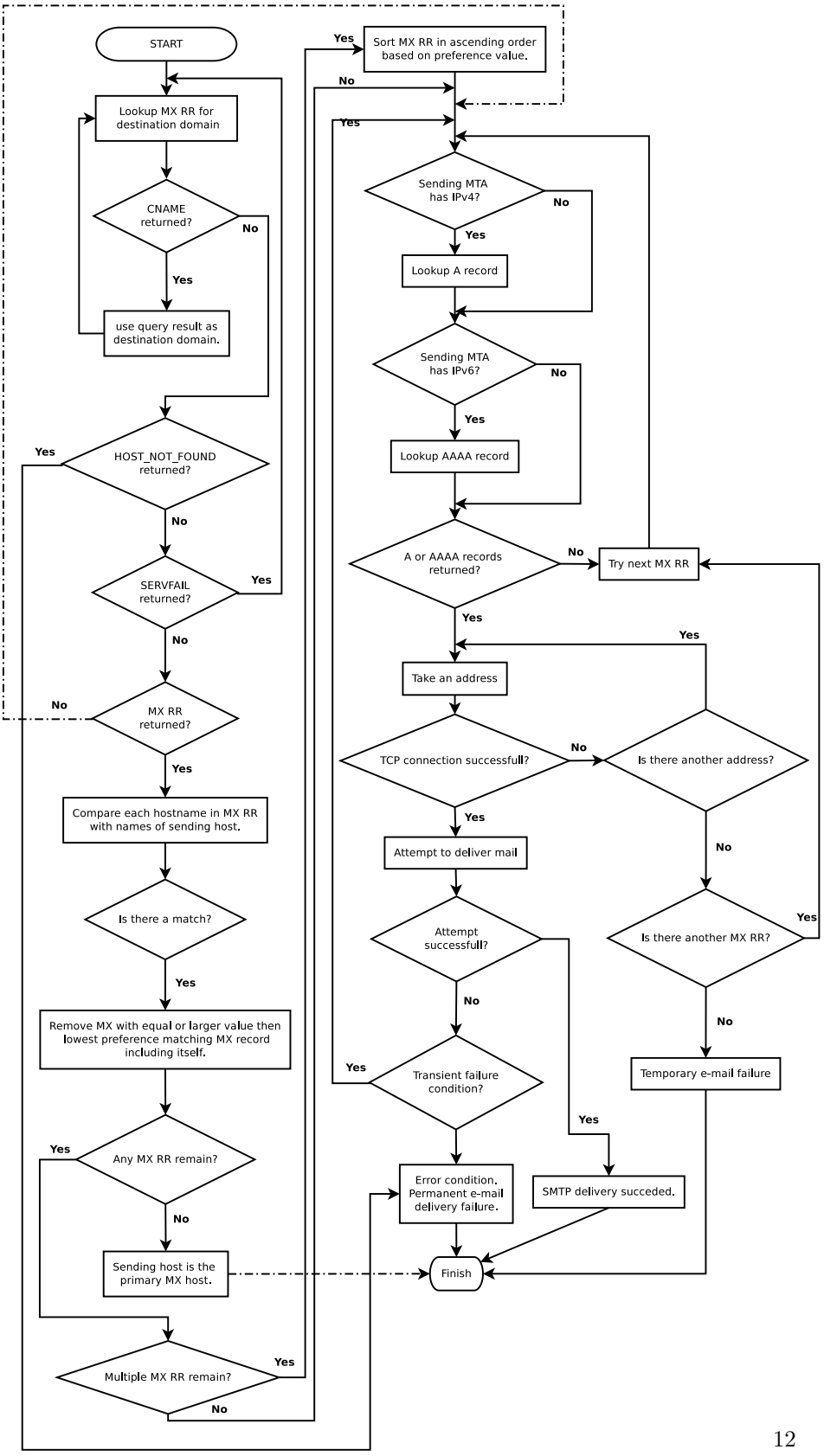

Figure 2: Algorithm for a Dual-Stack SMTP sender as described in RFC 3974 section 3.

3 Method and Findings

The following section shows several methods to analyse the behaviour of the various Message Agents that were described in Section 2.1. With each method there is a section that describes the expected behaviour according to the standards as well as the findings from testing various configurations.
The test setup that was used for this research consisted of two separate IPv4/IPv6 networks located at SARA[1] and SNE Lab[14]. Both networks have a /64 prefix for IPv6 and a /28 and a /27 respectively for IPv4 addresses. On these two networks various Operating Systems, including clients and servers, were installed using virtual machines created with the Xen[15] technology on a single physical machine at each location. Apart from these two networks, several other networks were used at various locations, i.e. SARA Office, SNE Lab and home networks (using 6to4, Teredo, ISATAP and Tunnel Broker protocols).
The tests are globally divided into the routing of MX RRs between MTAs and possible MTA reachability issues, the various implementations at the MUA and implementation problems that inflict both server and client behaviour.

3.1 MX RRs routing between MTAs

MX Resource Records are used for routing messages between various MTAs. Described in RFC 5321 "Simple Mail Transfer Protocol"[16] MX Resource Records have a preference and an address. With the introduction of AAAA RRs in RFC 1886 "DNS Extensions to support IP version 6", which are used to specify IPv6 addresses, an MTA can now use different kinds of transport to send messages, either with IPv4 or IPv6.
The following three tests will show various possible MX RR configurations and their impacted behaviour. The first test will look at how the selection of a destination address is done. The second and third test will show when design problems can occur in an IPv4/IPv6 mixed environment. These MTA tests have been done on Ubuntu 10.04 Server using the GNU libc version 2.9. The following MTAs were tested:

- Exim 4.71-3
- Sendmail 8.14.3-9.1
- Postfix 2.7.0-1
- Exchange 2007 Service Pack 1

These were installed using the default package provided by the Ubuntu repository. Exchange 2007 Service Pack 1 was installed on Windows 2008 Server.

Test 1: MX RRs routing between MTAs

Description

Section 2.3 shows that DNS round-robin can be used to load balance between Internet services. For MTAs this round-robin technique can be used in two ways. The first method is to have multiple MX RRs having the same preference. The second method is to use multiple A records within an MX RR. With the introduction RFC 3484 (Section 6, Rule 9) this load balancing technique using round-robin does not work properly any more as explained in Section 2.4.

Expected behaviour

- As described in RFC 3484 nodes must select a native IPv6 connection over IPv4.
- As described in RFC 3974 Dual-Stack nodes must try all AAAA and A records (in that order) before trying the next MX records.
- As described in RFC 3484 (Section 6, Rule 9) nodes must use the longest matching prefix.
- As described in draft-arifumi-6man-rfc3484-revise-02 nodes should not use the longest matching prefix. As quoted from this draft, it gives the following three possible changes in regards to RFC 3484:

  1. *To delete Rule 9 completely.*
  2. *To apply Rule 9 only for IPv6 and not for IPv4. In IPv6, hierarchical address assignment is the general principle, hence the longest matching rule is beneficial in many cases. In IPv4, as stated above, the DNS based load balancing technique is widely used.*
  3. *To apply Rule 9 for IPv6 conditionally and not for IPv4. When the length of matching bits of the destination address and the source address is longer than N, rule 9 is applied. Otherwise, the order of the destination addresses does not change. N should be configurable and it should be 32 by default. This is simply because the two sites whose matching bit length is longer than 32 are probably adjacent.*

The expected behaviour is tested by disabling different network connections to test the fall back mechanisms. Log analysing and wire sniffing was done to see the transport selection.

Findings

To begin with: all the tested MTAs support IPv6. It must be noted that IPv6 for Microsoft Exchange is supported as of version 2007 Service Pack 1 when installed on Windows 2008 Server. The most interesting part of this test is to see if the various MTAs comply with RFC 3484 and if any of the recommendations, that are described in draft-arifumi-6man-rfc3484-revise-02, are implemented.

Many administrators use DNS to implement some sort of load balancing for their services, this also applies for e-mail services. DNS round robin is used to specify multiple A records to link multiple addresses to a single MX RR. The client uses round robin to select one of the given addresses, this way the load is evenly distributed. With RFC 3484 this approach to load balancing is no longer an option as rule 9 in section 6 states that the longest matching prefix must be used, for both IPv4 and IPv6. The following MX RR configuration shows three A RR.

```
      IN     MX     10     mx10.skimbee.net.
mx10         IN     A       192.168.0.100
mx10         IN     A       192.168.1.100
mx10         IN     A       192.168.2.100
```

If, for example, a client has an IPv4 address of 192.168.0.1 it will use round robin to select one of the above A records. With the introduction of RFC 3484 this selection is not round robin any more as 192.168.0.100 is the longest matching prefix and the client will always try this address first. This kind of behaviour is theoretically good to have, but in practice not desirable as IP address assignment is not hierarchical, even with IPv6 as IPv6 PI addresses are now admitted by some RIRs. It would also break many DNS architectures that use DNS round robin for load balancing. To test this behaviour one can use `getent ahosts <domainname>` to see how GNU libc will present the list of available addresses to the applications to use. As observed, GNU libc version 2.9 has implemented RFC 3484 with the addition of option 3 of draft-arifumi-6man-rfc3484-revise-02:

> *3. To apply Rule 9 for IPv6 conditionally and not for IPv4. When the length of matching bits of the destination address and the source address is longer than N, the rule 9 is applied. Otherwise, the order of the destination addresses do not change. The N should be configurable and it should be 32 by default. This is simply because the two sites whose matching bit length is longer than 32 are probably adjacent.*

This is something to keep in mind when you want to load balance your services, and not only Message Agents, that clients from a particular network could always select the same address. When this behaviour was tested using the various MTAs it was observed that the address list that GNU libc will present to the MTA will not be used in that order but it will use round robin to first try all IPv6 addresses and all IPv4 addresses (in that order), as shown in Figure 3, and neglects Rule 9 in section 6 of RFC 3484. This behaviour is proposed as option 1 in draft-arifumi-6man-rfc3484-revise-02:

> *1. To delete Rule 9 completely.*

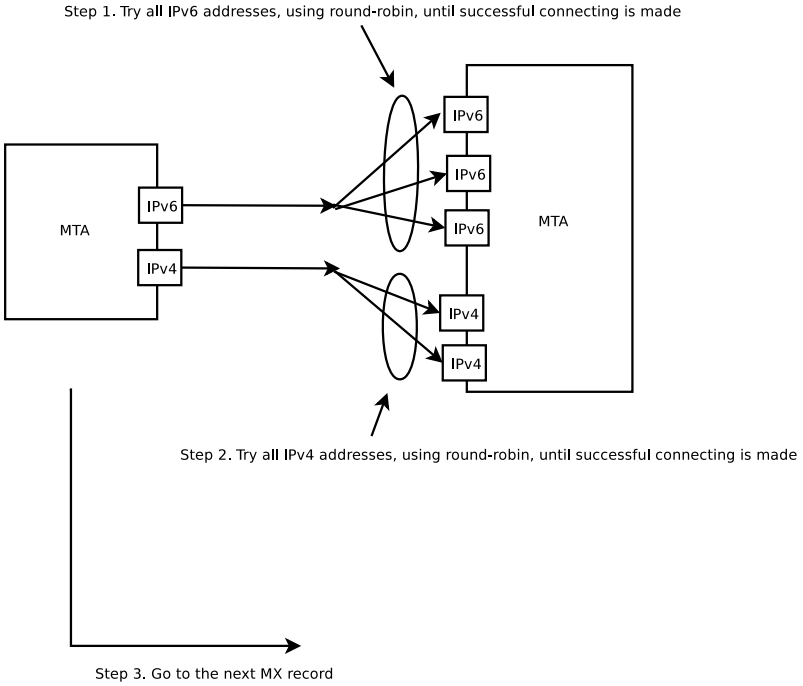

Figure 3: Observed behaviour of Message Submission Agents when connecting to a remote node.

The following is a section of an Exim log file where all three steps in Figure 3 are taken.

```
18:39:00 mx10.skimbee.net [2001:610:158:1056:145:100:106:195] No route to host
18:39:03 mx10.skimbee.net [2001:610:158:1056:145:100:106:b95] No route to host
18:39:06 mx10.skimbee.net [2001:610:158:1056:145:100:106:a95] No route to host
18:39:09 mx10.skimbee.net [145.100.106.199] No route to host
18:39:10 mx10.skimbee.net [145.100.106.195] No route to host
18:39:10 R=dnslookup T=remote_smtp H=mx20.skimbee.net [2001:610:108:2025:145:100:15:243]
```

It should be noted that there are cases where an MTA will cache successful connections for a given domain. This behaviour is equal for IPv4 and IPv6. If for example all IPv6 connections break in figure 3 it will use a successful connection for IPv4 even if the IPv6 connection is restored, it will try to reconnect to an IPv6 address once the cache is cleared or is renewed after a particular time in this given situation. Caching depends on the configuration of a particular MTA.

Test 2: MX RRs routing between MTAs

Description

In RFC 3974 a problem is described with having IPv4-only and IPv6-only serving the same domain. In this situation it is not possible to route e-mail traffic between MTAs. A solution is provided by having the preferred MX MTA Dual-Stacked, so that messages that arrive at a lower preferred MTA can always reach the highest preferred MTA.

Expected behaviour

With this highest preferred MTA all other MTAs, having IPv4-only or IPv6-only, will first try to deliver messages at this Dual-Stack MTA. So one does not end up with unroutable messages between MTAs.

Findings

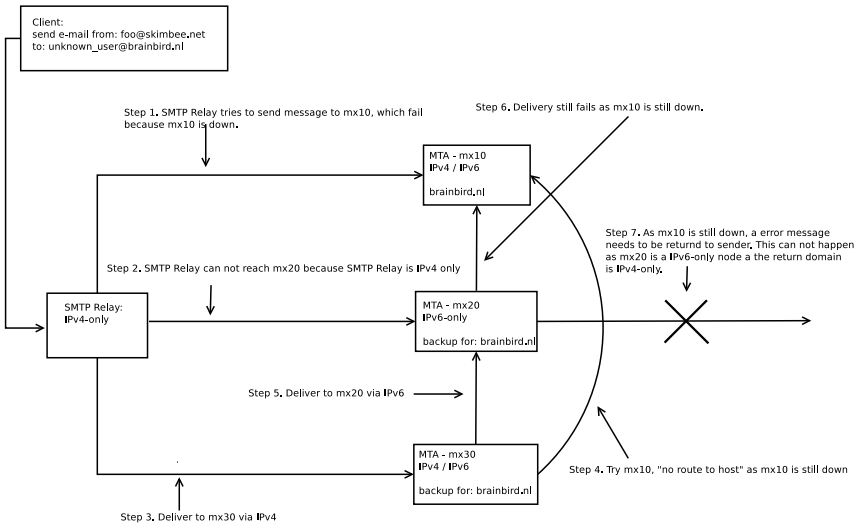

Figure 4: Routing between MX records in mixed IPv4/IPv6.

For this test an environment is created, as depicted in figure 4, where the highest preferred MX record is Dual-Stack, with having a preference of 10. The second preferred MTA has a preference of 20 and is IPv6-only. And the MTA with the lowest preference of 30 is Dual-Stack as well. All network interfaces on mx10 were shut down, which can be interpreted as this e-mail node being offline. An e-mail was sent via an IPv4-only SMTP Relay. As one could guess the SMTP can not reach mx10, as it is down, neither mx20 because it is IPv6-only. The SMTP Relay will route the message to mx30 which is Dual-Stack. Once it has arrived at mx30 it will try to contact mx10. When delivery to mx10 is unsuccessful, because it is still down, it will send the message to mx20, this is now possible as the mx30 is Dual-Stack.
Once it has arrived at mx20 multiple situations can be possible that would cause

the message to not reach its final destination, mx10. For example, it would be possible that mx10 does not become active after a certain period of time and an error message would now need to be send back to the sender because the recipient could not be reached. This delay time before it is returned is historically seven days. One can argue that over a seven day period a mail system should be back up and running to be considered reliable. However it is in practice possible that this will not be the cause, perhaps due to network issues between mx20 and mx10, where the IPv6 network is not monitored. In this situation the message would be frozen at mx20 because it did not reach the receiver, nor was an error message returned to the sender. Another possibility is that mx20 and mx30 do not check the final e-mail recipient for existence. In this situation an error message should be returned to the sender. This return message is then only generated after mx20 has sent the message to mx10 and at this final MTA the check is being performed on the existence of the end user. Most correctly configured MTAs do check if the final recipient can be reached in order to prevent backscatter. However while configuring our backup MTAs for this test, the backup MTAs are not aware of the users that reside on mx10.

Test 3: MX RRs routing from the SMTP relay

Description

Users can use an SMTP Relay to send messages to and from IPv4-only and IPv6 domain. This could cause problems, because the final destination is not checked for reachability before the message is accepted.

Expected behaviour

There are some obvious examples, like using IPv4-only and IPv6-only domains that can not be reached by the SMTP Relay. In this situation the behaviour is that messages will fill up the queue at the SMTP Relay. There are also some situations, that will be shown in the findings below, where normally messages will get sent but under certain circumstances will fail.

Findings

A user is using an IPv4-only SMTP relay for sending its messages to an IPv6-only domain, see step 1 in Figure 5. This is not a problem as long as messages arrive at their final destination. Reply to this e-mail can be problematic as it is an IPv6-only domain, but the replier will at least receive an error message about this. However when a message reaches the SMTP relay and that message cannot be delivered, for whatever reason, to its final destination, step 2 in Figure 5, an error message should be returned to the e-mail address of the sender. However, this e-mail address has an IPv6-only RR and the SMTP relay is IPv4-only so the delivery will fail, step 3 in Figure 5.

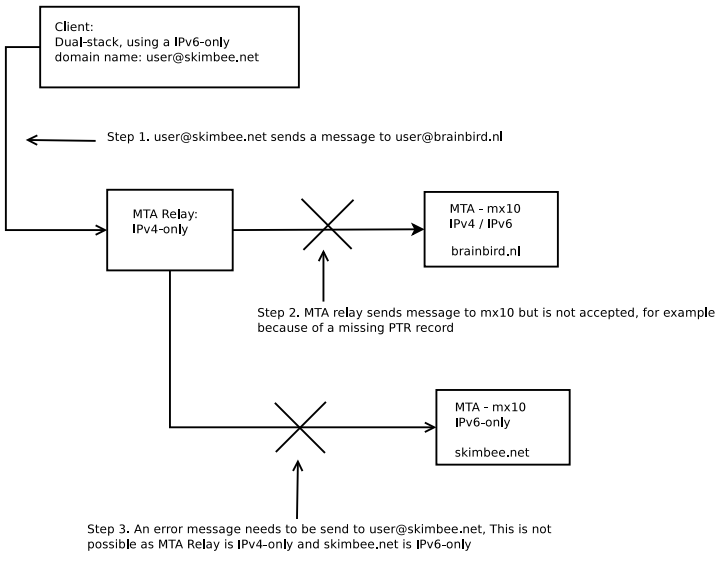

Figure 5: Routing at a SMTP Relay.

3.2 Message Delivery Agent (MDA)

Description

For accessing messages from a MUA a MDA can be used. Access to the MDA is commonly provided by POP3 and IMAP protocols. For this test we have selected the following MDA daemons:

- qpopper 4.0.9
- Dovecot 1:1.2.9
- Courier 0.63.0-2
- Cyrus 2.2.13-9

These where installed using the default package provided by the Ubuntu repository.

Expected behaviour

As the MDA only gives services from an access perspective and does not need to contact a client by itself (an therefore not having to make the appropriate address selection) IPv6 interoperability problems should be minimum. However we do expect the modern daemons should have IPv6 accessibility by default.

Findings

All the MDAs that were tested did not have any problems with providing the same service over IPv6 as they do with IPv4. The only MDA that had some problems with IPv6 was Dovecot as it does not listen on IPv6 by default. The following telnet session to the POP3 services to a Docevot deamon shows that the telnet session falls back to IPv4.

```
$ telnet zoot.skimbee.net 110
Trying 2001:610:108:2025:145:100:15:245...
Trying 145.100.15.245...
Connected to zoot.skimbee.net.
Escape character is '^]'.
+OK Dovecot ready.
```

Firstly it tries to contact the services over IPv6 and then switches back to IPv4. This is not a problem if the MUA provides this fall back mechanism, however as we will see in Section 3.3 this is not always the case. It is a small change to let Dovecot listen to IPv6 by making the following change to "/etc/dovecot/dovecot.conf":

```
< listen=*
---
> listen=[::]
```

If we now contact the POP3 services it will connect via IPv6 without any problem.

```
$ telnet zoot.skimbee.net 110
Trying 2001:610:108:2025:145:100:15:245...
Connected to zoot.skimbee.net.
Escape character is '^]'.
+OK Dovecot ready.
```

In Section 3.3 it will be apparent that not all MUAs fall back to IPv4 when IPv6 is not working, therefore one must make sure that the deamon is working correctly over IPv6 before announcing AAAA RRs for the MDA services.

3.3 Message User Agent (MUA)

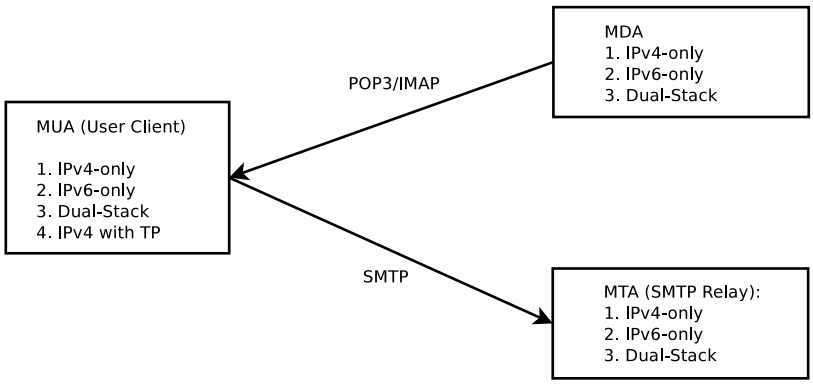

Figure 6: Message User Agent (aMUA).

Description

Test the MUAs on connectivity to the mail box. Tests must be done with IPv4-only, IPv6-only, Dual-Stack for both nodes and with the additional test with IPv4 with IPv6 transition technique only for the aMUA.

Expected behavior

1. As described in RFC 3484 nodes must select a native IPv6 connection over IPv4.

2. Applications should fall back to IPv4 when IPv6 is not working.

Expected behaviour is tested by disabling different network connections.

3.3.1 Findings

The submission of messages, via SMTP, and the retrieval, via POP3 and IMAP, at the MUA is very similar when looking at IPv4 and IPv6. That is why they are combined in this one section.

Clients can have IPv6 transition technologies enabled by default. Selecting the proper transportation mechanism, as described in RFC 3484, and fall back mechanisms is the main cause of problems for Message User Agents.

All MUAs on Apple Mac OS X 10.6 (Snow Leopard)
Mac OS X has support for IPv6. However for address selection it does not implement RFC 3484 for selecting the appropriate transport. As of Mac OS X 10.6 Apple has made a change to mDNSResponder where transport selection is based on the DNS response which is returned first after simultaneously sending a query for an A and an AAAA RR[4]. This way of selecting the transport is implemented to try to give the user the faster working connection and make it available without waiting for all the responses. With this implementation no

fall back is possible because the second answer is completely discarded. It can cause more serious problems if an end host is IPv6-only and only an AAAA RR is available but the answer for the A RR is returned first with "NOERROR".

Mozilla Thunderbird on Linux and Windows based systems
Thunderbird first tries all AAAA and then all A records when it wants to send a message to an MSA. Thunderbird will connect to every RR three times, initially it retries after 3 seconds and doubles the previous time after each try. The following is a cut of message log of a Thunderbird client that makes three attempts to connect to an IPv6 address and then falls back to IPv4. The increase in time between each attempt is shown below.

```
18:25:42. IP6 2001:610:108:... > 2001:610:108:....smtp: Flags ....
18:25:45. IP6 2001:610:108:... > 2001:610:108:....smtp: Flags ....
18:25:51. IP6 2001:610:108:... > 2001:610:108:....smtp: Flags ....
18:26:03. IP michiel.local.33585 > 145.100.15.245.smtp: Flags ....
```

If there is no successful connection it will give an error message. Having multiple RRs will increase the time before an error message is given when all connections fail.

Microsoft Live Mail
Live Mail, on Windows clients that are IPv6 capable, have the same behaviour as Mozilla Thunderbird.

Microsoft Outlook 2007 and 2010
In many business environments Microsoft Outlook is used as a MUA. As of Outlook 2007 IPv6 can be used to access various IPv6 services. As with all applications fall back to IPv4 must be implemented if the IPv6 transport is not working. However Outlook 2007 and 2010, tested on Windows Vista, did not provide this fall back to IPv4 as the following error message shows:

```
Task 'michiel@skimbee.net - Sending' reported error (0x80042109) :
'Outlook cannot connect to your outgoing (SMTP) e-mail server.
If you continue to receive this message, contact your server administrator
or Internet service provider (ISP).'
```

The above shows the error message that the user is presented when a connection to an IPv6 SMTP service fails. In this situation the SMTP service, which is announced with A and AAAA RRs, is only accessible over IPv4. Not having correctly implemented a fall back to IPv4 can cause a lot of user problems when IPv6 transport fails.

3.4 Implementations that affect both Servers and Clients

Description

Selecting the appropriate transport mechanism is described in RFC 3484. This document states that native transportation should be preferred over a transition technology, only when the end node is IPv6-only an IPv6 transition technology must be used. There is however a shortcoming in RFC 3484 that does not take into account that a node using RFC 1918 address space can be connected to the Internet via, for example, NAT. This shortcoming is addressed in draft-arifumi-6man-rfc3484-revise-02 but has not yet been implemented by some operating systems.

Findings

GNU libc implements RFC 3484 but does not apply to draft "draft-arifumi-6man-rfc3484-revise-02" because the maintainer of GNU libc wants to wait for the draft to become final. However Fedora, Canonical, Gentoo, Novell, Mandriva and Debian have all applied patches.

4 Recommendations and Conclusion

The tests and analyses from Section 3 imply that the various Message Agents, described in Section 2.1, can be divided into two environments when it comes to practical operational architectures. One for submitting and receiving e-mail at the user client and another for routing e-mail between various MTAs.
Because all tests were done in a lab environment and not on a full running e-mail infrastructure only problems that originated in this controlled environment were revealed. Despite this limitation, there are some very interesting results that can help administrators with the introduction of IPv6 on their infrastructure.

4.1 Recommendations

It is advisable to start introducing Dual-Stack on MTAs first so that remote hosts can always reach the destination in the event of a remote IPv6-only node where messages can become unroutable, as shown in Section 3. No implementation problems occurred at the popular MTA daemons that were tested. When introducing IPv6 at the SMTP relay servers and access servers, the servers that the MUAs are using, problems can be expected as shown in Section 3. It is recommended to firstly monitor the impact in a controlled environment, like announcing AAAA RRs on a separate test network, or a network where the administrator has control over the clients.

When introducing IPv6 on your e-mail infrastructure it is recommended to always use, if possible, a Dual-Stack setup to prevent unroutable e-mail messages, as shown in Section 3.1. Furthermore it is important to keep the use of transition mechanisms and tunnels to a bare minimum because of their unreliability and unpredictable behaviour. This unreliability is the result of minimal attention that these connections get in comparison with native connections, that are usually monitored in larger networks. Moreover transition mechanisms rely on IPv4 to work, this introduces a double reliability. Lastly some of them introduce layer violation relying on a higher level protocol to perform routing [17]. Transition mechanisms are automatically configured on Windows Server 2008 and it is perhaps a good idea, to disable this and enforce native connectivity to servers. It goes without saying that an administrator should never make any services available over transition mechanisms, like configuring an AAAA RR that points to a transition mechanism. Some top-level domain (TLD) authorities, like DENIC which is responsible for the .de domain, prohibit the use of transition mechanisms as an authoritative name server to fore come these kinds of configurations.

IPv6 introduces a privacy issue when Route Advertisements (RA) are used at clients that use different networks. With RAs the IPv6 network address is constructed based on the client's MAC address. This MAC address is unique for every client and can keep track of a client when it changes networks. RFC 4941 "Privacy addresses" [18] allows a node to create an IPv6 random address at a particular interval. The partly random IPv6 address is then used for communication. Therefore an administrator must keep in mind that a client that has implemented RFC 4941 can have different IPv6 addresses each time the client makes a connection. Log analysing and debugging make this construction more problematic. All Windows clients have implemented this privacy behaviour for IPv6 addresses.

It can be turned off on an internal network by changing the registry[19]. An IPv6 statefull address can be forced on a client using DHCPv6, however not all operating systems (Windows XP and Mac OSX) have a DHCPv6 client available.

Spam is the largest problem that e-mail systems currently have to deal with. With IPv4 a sending spam node can be easily blacklisted based on the nodes IP address[20]. In an IPv6 network a different approach is needed as nodes can easily switch between different, clean, IPv6 addresses when in a /64 network, which is not uncommon. The swapping of addresses makes blacklisting useless as it only creates an enormous list of IPv6 addresses. There are incentives[21] that are now operational that would block a complete /64 network if multiple abusing IPv6 addresses are detected. It is therefore advisable to separate the e-mail infrastructure on a separate /64 network to make sure the servers will not get blacklisted because of clients sending abuse.

If Dual-Stack is introduced on any of the services the administrator must also make a double reachability configuration in the monitoring solutions. For all of the Dual-Stack services this means that the administrator has to double the amount of service checks. Keeping track of the amount of IPv4 and IPv6 network utilization is also an approach to see if the Dual-Stack network is still functioning.

4.2 Conclusion

The study presents problem situations in designs and architectures that are not uncommon. It must be noted that all of the problems were caused by a mixed IPv4/IPv6 environment in the transition to IPv6. These problems would not be present in an IPv6-only setup.
Despite these problems, our conclusion is that it is safe to introduce IPv6 on an e-mail architecture if a correct design (meaning Dual-Stack on all services) is implemented and if the possible problems described with MUAs are kept in mind.

Abbreviation

aMUA	-	Author Message User Agent
CPE	-	Customer-Premises Equipment
DNS	-	Domain Name Server
IANA	-	Internet Assigned Numbers Authority
IMAP	-	Internet Message Access Protocol
IPv4	-	Internet Protocol version 4
IPv6	-	Internet Protocol version 6
MAC	-	Media Access Control
MDA	-	Mesage Delivery Agent
MHS	-	Message Handling System
MSA	-	Message Submission Agent
MS	-	Message Storage
MTA	-	Message Transfer Agent
MUA	-	Message User Agent
MX	-	Message eXchange
NAT	-	Network Address Translation
POP3	-	Post Office Protocol - Version 3
PTR	-	Pointer Resource Record
RA	-	Route Advertisement
RFC	-	Request for Comments
RIR	-	Regional Internet Registry
rMUA	-	Recipient Message User Agent
RR	-	Resource Record
SMTP	-	Simple Mail Transfer Protocol

References

[1] **Website: SARA Reken- en Netwerkdiensten**
As seen on: June 16 2010
http://www.sara.nl/

[2] **Website: IPv6 dual-stack client loss in Norway**
Author: Tore Anderson
As seen on: June 16 2010
http://fud.no/ipv6/

[3] **Website: IPv6 operators forum, Archives**
Subject: Opera Browser problem
As seen on: June 16 2010
http://lists.cluenet.de/pipermail/ipv6-ops/2009-October/002635.html

[4] **Website: Mac OSX 10.6 Rejected DNS responses causing IPv6 failures**
Product: Mac OS X, Product Version: 10.6, Classification: Serious Bug
As seen on: June 16 2010
http://openradar.appspot.com/7333104

[5] **Slides: Google IPv6 Implementors Conference 2010 - IPv6 at Google**
Author: Lorenzo Colitti
Date: June 10th 2010
https://sites.google.com/site/ipv6implementors/2010/agenda/09_Colitti_IPv6atGoogle.pdf

[6] **RFC 5598 - Internet Mail Architecture**
Date: July 2009
http://www.rfc-editor.org/rfc/rfc5598.txt

[7] **RFC 4409 - Message Submission for Mail**
Date: April 2006
http://www.rfc-editor.org/rfc/rfc4409.txt

[8] **Miredo: Teredo for Linux and BSD**
As seen on: 28th June 2010
http://www.remlab.net/miredo/

[9] **RFC 3484 - Default Address Selection for Internet Protocol version 6 (IPv6)**
Date: February 2003
http://www.rfc-editor.org/rfc/rfc3484.txt

[10] **RFC 1794 - DNS Support for Load Balancing**
Date: April 1995
http://www.rfc-editor.org/rfc/rfc1794.txt

[11] **RFC 1918 - Address Allocation for Private Internets**
Date: February 1996
http://www.rfc-editor.org/rfc/rfc1918.txt

[12] **Things To Be Considered for RFC 3484 Revision)**
Date: October 19th 2009
http://ietfreport.isoc.org/all-ids/draft-arifumi-6man-rfc3484-revise-02.txt

[13] **RFC 3974 - SMTP Operational Experience in Mixed IPv4/v6 Environments**
Date: January 2005
http://www.rfc-editor.org/rfc/rfc3974.txt

[14] **SNE/OS3 Homepage**
As seen on: 13th July 2010
https://www.os3.nl/

[15] **Xen**
As seen on: 28th June 2010
http://www.xen.org/

[16] **RFC 5321 - Simple Mail Transfer Protocol**
Date: October 2008
http://www.rfc-editor.org/rfc/rfc5321.txt

[17] **Internet Protocol Version 6, Teredo, and Related Technologies in Windows Vista**
As seen on: 28th June 2010
http://technet.microsoft.com/en-us/library/cc722030(WS.10).aspx

[18] **RFC 4941 - Privacy Extensions for Stateless Address Autoconfiguration in IPv6**
Date: September 2007
http://www.rfc-editor.org/rfc/rfc4941.txt

[19] **Technet Magazine: IPv6 Autoconfiguration in Windows Vista**
As seen on: 8th July 2010
http://technet.microsoft.com/en-us/magazine/2007.08.cableguy.aspx

[20] **CBL: Composite Blocking List**
As seen on: 8th July 2010
http://cbl.abuseat.org/

[21] **The Virbl-project**
As seen on: 8th July 2010
http://virbl.bit.nl