

2009

PACIFIC NW SOFTWARE  
QUALITY  
CONFERENCE



MOVING  
QUALITY  
FORWARD

OCTOBER 27-28, 2009

Conference Paper Excerpt  
From the

CONFERENCE  
PROCEEDINGS

Permission to copy, without fee, all or part of this material, except copyrighted material as noted, is granted provided that the copies are not made or distributed for commercial use.

# Testing IPv6 Enabled Applications

Travis Luke

Software Development Engineer in Test

Microsoft Corporation

tluke@microsoft.com

## Abstract:

Each year IPv6 gains wider adoption around the world, both as a replacement for traditional TCP/IP and in side by side usage. IPv6 transition technologies are now enabled by default on many modern operating systems and applications have already begun to take advantage of them. However, very few testing guidelines exist for this emerging technology. It is important that we move quality forward in this ecosystem. First, I will present an overview of the current state of IPv6 deployment. Next, I will focus on the most popular transition technologies (6to4, Teredo and ISATAP). I will describe how each technology acts as a bridge to a full IPv6 deployment and what software developers and testers need to know about it. Then, I will describe how to build a test lab that simulates various IPv6 environments such as the home, the Internet café, and the enterprise. Last, and most importantly, I will outline practical test cases for testers of IPv6 enabled applications.

## Author Bio:

Travis Luke has been working at Microsoft a Software Development Engineer in Test for ten years. For the past five years, he has worked in the Windows Networking division on Peer-To-Peer network technologies. Prior to Microsoft, Travis worked as a Network Administrator and consultant. He enjoys talking about himself in the 3<sup>rd</sup> person.

## Why should we all care about IPv6?

There seems to be a never ending debate about IPv6. Is it really needed? Is IPv6 the solution to IPv4 address depletion? Why can't we all use a NAT and be happy? (For a definition of what a NAT is see the appendix). If you are interested in adding to this debate then there is a big world on the Internet for you. The fact of the matter is IPv6 is already here. Comcast recently announced that some residential customers will have IPv6 connectivity by 2010. Some of the most popular web sites on the Internet now have IPv6 offerings. (ipv6.netflix.com, ipv6.Google.com). The 2008 Olympics featured an official IPv6 enabled website (ipv6.beijing2008.cn/en). In fact all network operations of the Olympic Games that year were conducted using IPv6. Even the Pirate Bay has IPv6 enabled web sites and trackers (ipv6.thepiratebay.org). Microsoft Windows Vista and Apple Mac OS X v10.3 support IPv6 and is enabled by default.

There is a wide range of applications using IPv6. Of course, the obvious applications are the web browsers. Since the World Wide Web is the biggest killer app for IPv4, it makes sense that it will be for IPv6 as well. However, there is another class of applications that use IPv6 for its ability to traverse NATs and tunnel over IPv4. Examples of this include Videoconferencing tools such as Ekiga/Gnomemeetin, and ISABEL. Other examples are Media streaming applications like Windows Media Player and Videolan. This may also include games such as Quake3, or peer to peer file sharing applications such as Gnuetella. The Remote Assistance tool in Windows 7 uses IPv6 tunneled through IPv4 to establish end-to-end connections with your peers to request or offer support.

So it is clear that IPv6 is something we should all care about. Chances are we have used it without realizing it. Therefore, as we build applications it becomes important to test that they work well with IPv6 and provide a seamless transition to the end user.

## The Five-Minute IPv6 Refresher Course

This is a brief overview of IPv6. This will help build the context for the discussion later in this paper. Many important details and concepts will be skipped over. For a complete overview of IPv6 there are

### A Link Local Address

**fe80::205:ddff:fe27:3840**

many books available. See the bibliography at the end of this paper for references to my favorites. An IPv6 Address is a 128-bit address that is printed in hexadecimal format. Each 16-bit block is separated by a colon. For example the address on my PC is 2001:4898:1b:5:ac39:2d8a:2229:e051. There is a shortcut to compress the zeros by using two consecutive colons to indicate 16-bit blocks comprised of zeros. For example fe80:0:0:0:205:ddff:fe27:3840 is the same as fe80::205:ddff:fe27:3840. Addresses come in three types: Unicast, Multicast, and Anycast. For the purposes of this discussion we will focus on Unicast addresses. The first 48-bits (3 blocks) represent the Global Routing Prefix. This distinguishes the scope of the address. Global addresses are addresses that can be reached by any other global address. This is

**2001:4898:1b:5:ac39:2d8a:2229:e051**  
  
Network ID                      Host ID

similar to the public addresses range on IPv4 networks. The next 16 bits represent the subnet ID. Link-local addresses are not routable and are only reachable by nodes on the same subnet. Link-local addresses always have the Global Routing Prefix and subnet ID of fe80:0:0:0. Together the Global Routing Prefix and the subnet ID make up the network ID of the address.

The remaining 64 bits is the interface ID. Your interface ID is assigned by the network stack. The MAC address of your adapter is often used to create the interface ID. Each adapter you have is given a link-local address by the network stack. If your router is configured for IPv6 it will assign your adapter a Global subnet ID and your network stack will create one or more Global IPv6 addresses. In this way, each adapter may have many addresses. Unlike IPv4 the subnet ID and interface ID is fixed so there is no need for a subnet mask.

Sometimes you will see a %nn following an IPv6 address. This is called the scope ID. The number following the '%' sign is the adapter ID that this address is associated with. Usually scope IDs are only used in link-local addresses.

## IPv6 Transition Technologies

To help ease migration to IPv6, a number of transition technologies have developed. Each of these helps in its own way and has its own quirks.

### Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

ISATAP creates IPv6 addresses based on IPv4 addresses. An ISATAP address looks like this: 2001:4898:0:fff:200:5efe:157.59.29.11. An ISATAP address can have a link-local or a global scope. Link-

local addresses are limited to connectivity to nodes on the same subnet. Global Scope ISATAP addresses can go anywhere that the router lets them go. To get a Global ISATAP address there must be a ISATAP server in the network that is configured to assign addresses, and you must have routes defined to route the addresses across subnets. Due to these limitations ISATAP is typically deployed in Enterprise Scenarios. ISATAP addresses can be identified by looking at the first 32-bits of the host ID. They always have a starting host ID of 0:5efe or 200:5efe.

A Link Local ISATAP address

**fe80::200:5efe:157.59.29.11**

### 6to4

This protocol allows IPv6 packets to be transmitted over a IPv4 network without the need to create an explicit tunnel. The Global 6to4 IPv6 address is generated based on the public IPv4 address. The upside of this is that if

you have a public IPv4 address you can instantly have access to a 6to4 IPv6 address and fully access IPv6 resources. The downside is you must have a public IPv4 address. If you are behind a NAT then most likely your NAT hides the public IPv4 address and assigns you a private ipv4

A 6to4 Address

**2002:836b:1::836b:1**

address. However some modern NATs such as the Apple AirPort Extreme, actually assign 6to4 addresses to all of its connected nodes. 6to4 addresses can be identified with the first 16 bits of the network ID which are always 2002.

## Teredo

Teredo is a tunneling protocol that grants Global IPv6 connectivity through NATs. It works by connecting to a Teredo Server on the internet using UDP

over IPv4. That server assigns a Global IPv6 address to the Teredo Client. Then the server routes traffic between that client and other Teredo and Non Teredo hosts. Teredo is not compatible with all NAT devices. Teredo has become very popular not just for IPv6 connectivity, but also as a convenient way to seamlessly connect to nodes behind NATs. In fact, the name Teredo is the Latin name for shipworm which is a reference to the protocol's ability to "punch holes" through NATs. Teredo addresses can be identified by the first 32 bits of the network ID which are always 2001:0.

### A Teredo Address

`2001::cd49:7601:a866:efff:62c3:fffe`

## Building your Test Infrastructure

In order to test your applications on IPv6 networks you need to create an IPv6 network. However as you can see from the many transition technologies above, there are many different types of deployments. The first step to building your test lab is to identify who your customer is and where they will be running your software. Then you can understand your customer's environment and scenarios. Let's analyze the home, public, and enterprise network topologies.

Today's home network usually consists of one high speed Internet connection connected to a wireless NAT, which connects to one or more PCs. However, not all home networks are created equal. Some do not have a NAT and get their connection straight from the ISP. And not all NATs are created equal. There are various NAT algorithms in use such as Cone Nats, Port restricted NATs, and Symmetric NATs. The latter, symmetric NATs, may inhibit Teredo functionality. Some NATs natively support 6to4. Others have firewalls that block all IPv6 traffic. To make matters worse, some NATs have been known to crash or hang in the presence of some specific IPv6 traffic. So it is important, when you test your home scenario, that you use a wide variety of NAT hardware and settings. Most likely you will not be able to solve all problems related to home networking hardware and configurations. But as a software developer, you will be able to discover how your software behaves in the presence of a hostile networking ecosystem. What you learn can then be applied to product support, recommended hardware guides, and documentation.

Network Environment	IPv6 type
No Internet Access	Link-local IPv6
Dialup	6to4
Direct Connection to Public Internet	6to4
NAT (non-6to4)	Teredo / Link-local IPv6
NAT (6to4)	6to4 / Link-local IPv6

Public networks are in many ways similar to home networks; however there are a few things to be aware of. Administrators of public networks want to ensure safe Internet access for its users. Therefore, the firewall settings on public networks tend to block more traffic than usual. Because of this, Teredo functionality can be limited. Some public networks take the additional precaution of blocking traffic between peers connected on the same network. Many public networks require you to go to a login web-site and accept their terms of usage before your traffic is routed to the public Internet. This timing may affect the Teredo Service's ability to connect to the Teredo server. Also, some public networks are configured incorrectly, which breaks IPv6 connectivity. For example, you may occasionally encounter a public network that assigns public IPv4 addresses to all of its clients, but is still behind a NAT. This will make the clients assign themselves 6to4 addresses but be unable to connect to IPv6 resources outside of the NAT. If you are building an application that will be used in public networks you may want to build a test lab that simulates these types of problems. Additionally you may want to visit public networks in your local area and test it out yourself in the real world.

Network Environment	IPv6 type
Basic Internet Access	Teredo / Link-local IPv6

Enterprise Networks are the easiest to simulate in a lab. This is because enterprises typically provide greater connectivity between all nodes. Most enterprises use proxy servers which completely limit external Global IPv6 access. Increasingly Enterprises are deploying either Native IPv6 internally or configuring an ISATAP server. There may be challenges involving remote access however. Not all remote access hardware and software is IPv6 aware. Interestingly, Windows 7 "Direct Access" technology is based on Teredo.

Network Environment	IPv6 type
IPv4 Only	Link-local IPv6 / Link-local ISATAP
ISATAP Servers	Link-local IPv6 / Link-local ISATAP/Global ISATAP
Native IPv6	Link-local IPv6 / Native Global IPv6
Remote Access	? Depends on method of remote access

### Test Cases for your IPv6 Enabled Applications

There are a number of test cases that should be executed against all IPv6-enabled applications.

1. **Verify proper address selection.**

With IPv6 there are many addresses associated with a single adapter. Your application must choose to bind or advertise the correct address.

- a. *Is the application choosing an address from the correct scope?*

If you want resources outside of your subnet to connect to your application you need to make sure that you bind to and advertise a Global IPv6 address. If you are only going to communicate with machines on the same subnet then you need to verify that it chooses addresses on the link-local scope. You also need to verify that your application performs the correct action in the event the address scope you want is not available.

- b. *Does the application properly react to address changes?*

Sometimes the router or transition technology is slow about assigning a global address, so your application may need to listen for and react to address changes. You may need to simulate address changes by adding routes to your router or enabling/disabling an upstream Internet connection.

- c. *Is the application choosing the best address within the selected scope?*

It is possible that you have multiple Global IPv6 addresses. You could have a Teredo address and a 6to4 address. You may have a Native IPv6 address and a ISATAP address. You need to verify that your application listens to or advertises the best address for its needs.

- d. *Is my application using the Public or Anonymous Address?*

When dealing with Native IPv6, the networking stack will often assign a second Global IPv6 address. This is called an anonymous or temporary address. This address has the same network ID as the first address but with a different host ID. While the Host ID for the first address is based on your MAC address, the Host ID for the anonymous address is random, and changes every few hours. The rationale behind this is to allow for anonymous access to Internet resources. If you connected with your public address while browsing the Internet web sites could track your usage since they could derive your MAC address from the address. However if you connect with your anonymous address they can only track you until your address changes. So in some cases an application should use the anonymous address. In other cases, primarily involving server scenarios requiring fixed, unchanging addresses, you must use the public address.

## **2. Verify the application works in IPv6 only environments**

This means configuring an environment in which there is no IPv4, and disabling IPv4 in your operating system. If your application is designed to run in Native IPv6 only scenarios then disable IPv4 and make sure that you don't have any hidden dependencies. Often times networking applications make use of protocols such as HTTP, ICMP, SSDP, or WSD. The application may need to invoke the correct flags or APIs to ensure that these protocols are not using IPv4.

## **3. Verify the application behaves when IPv6 is not available.**

Disable IPv6 on your machine and try to use your application. Does it failover to IPv4? Does it crash? Or does it give a useful error message?

## **4. Verify security settings.**

Because Teredo "punches holes through NATs", it opens up potential security risks of unsolicited Internet traffic reaching your application. To combat this, the Teredo implementation on Microsoft Windows has a strict security model that must be followed for the Teredo client

service to operate. This includes having an IPv6 capable firewall. Additionally, there are some socket options which can be set to allow or disallow unsolicited traffic from the Internet. For more information on this see the MSDN article at: <http://msdn.microsoft.com/en-us/library/aa832668.aspx>.

**5. Test with many personal firewall vendors.**

Many PCs come bundled with personal firewalls. Not all of these firewalls have been fully tested with IPv6. Some firewalls block all IPv6 traffic. Some allow all IPv6 traffic unfiltered. By testing on a variety of the most popular firewalls you can add to your knowledge base how your software reacts. This knowledge can be added to your documentation, support library, or recommended software/hardware lists.

**6. Do not share addresses with the scope ID.**

Link-local addresses often have the scope ID at the end to indicate the adapter associated with that address. It is written in the form of %nn. For example fe80::ac39:2d8a:2229:e051%13 has a scope ID of 13. The scope ID is only useful for the machine that the address is on. If you convert that address to a string, copy it to another computer, and try to ping that address the scope ID would be meaningless or misleading. So if your application ever shares the address with another machine you must verify that the scope ID is removed.

**7. Correctly format any address displayed.**

Whenever an IPv6 address is displayed it should be correctly formatted. These are not major blocking issues. They are cosmetic bugs. IPv6 addresses are large and hard to read. By following these guidelines it will be easier to read the addresses. When used in application tracing and test logs it will be easier to spot other bugs. In general you may want to avoid displaying addresses to the user. When you must display them then it is important to follow these guidelines. Most IPv6 implementations come with an API that helps in properly formatting an address for displaying. The following table shows common bugs with displayed addresses:

Wrong Format	Correct Format	Comments
fe80:0:0:0:ac39:2d8a:2229:e051%13	fe80::ac39:2d8a:2229:e051%13	Use the :: shortcut to cut out octets with a zero value
2001:4898:001b:0005:ac39:2d8a:2229:e051	2001:4898:1b:5:ac39:2d8a:2229:e051	Do not display leading zeros in octets
fe80::200:fe:9d3b:1d0b	fe80::200:5efe:157.59.29.11	ISATAP addresses should display the embedded IPv4 address rather than the hexadecimal equivalent.

## Happy Testing

IPv6 is a technology which we have to face. As more applications begin to taking advantage of the end-to-end connectivity and ISP begin IPv6 offerings the need for support will increase. The wide variety of IPv6 transition technologies makes testing difficult because of the wide amount of test environments. By focusing on your primary customer scenarios, test coverage becomes achievable. The test cases outlined in this document will help ensure quality in your product as the transition to IPv6 moves forward. More than anything a good background knowledge of IPv6 and your customer scenarios is the key to success.

## Appendix

### What is a NAT?

NAT stands for Network Address Translation. It is a device that acts as a router between the Internet and a private network. Nearly all home wireless routers serve as a NAT. On the private side of the NAT, or the home network, devices are assigned addresses private IPv4 addresses. These addresses are not routable by devices with public IPv4 address which reside on the Internet side of the NAT. The NAT software takes outgoing packets from the private network and modifies the header to make it appear it came from the NAT's public address. The NAT attempts to route incoming packets to the correct host on the private network. If it cannot determine what host the packet is intended for the NAT will drop the packet. This technology works great for web browsing and email. However it works poorly for establishing end-to-end connectivity between two hosts. Wikipedia has an excellent article on NATs at [http://en.wikipedia.org/wiki/Network\\_address\\_translation](http://en.wikipedia.org/wiki/Network_address_translation).

## Bibliography

Davies, Joseph. Understanding IPv6, Second Edition. Redmond WA: Microsoft Press, 2008

Hagen, Silvia. IPv6 Essentials, Second Edition. Sebastopol, CA: O'Reilly Media, 2006

Blanchet, Marc. Migrating to IPv6. West Sussex, England: John Wiley & Sons Ltd, 2006

Kerner, Sean Michael. "Comcast Embraces IPv6" 18 June 2009 <  
<http://www.internetnews.com/infra/article.phpr/3825696/Comcast+Embraces+IPv6.htm>>