



Intel[®] Technology Journal

Interoperable Home Infrastructure

**Internetworking Using
IPv6 Technology Inside
and Outside the Home**

Internetworking Using IPv6 Technology Inside and Outside the Home

Venkat R Gokulrangan, Desktop Platform Group, Intel Corporation

Index words: IPv6, IPv4, UPnP, Home Networking, NAT, Broadband

ABSTRACT

The lack of sufficient globally routable Internet Protocol (IP) addresses to be assigned to subscribers has led service providers to deploy techniques such as Network Address Translation (NAT). However, NAT breaks end-to-end connectivity in several applications resulting in a poor end-user experience. Internet Protocol version 6 (IPv6) is the next-generation network layer technology designed to provide globally unique IP addresses to every endpoint in the Internet for years to come, thereby resolving the address depletion problem. IPv6 technology has several features such as auto-configuration, security, and mobility, built into its design, thereby creating an easy, rich, plug and play networking experience for the end user.

The key to IPv6 deployment is a smooth transition from current IP version 4 (IPv4)-based networks to new IPv6-based networks. It is generally accepted that the transition will span several years, marked by the emergence of isolated IPv6 islands in the customer premises, co-existing with IPv4. The inter-island communication will be carried over tunnels created over existing IPv4 networks. Eventually, as the core Internet progressively deploys native IPv6 networks, the tunnels will be removed, leading to the completion of the transition.

This paper discusses the core architecture of IPv6, its key features related to end-user experience, and the common IPv4 to IPv6 transition models. We also present a simple solution implemented on an Intel[®] XScale[™] core-based platform, which illustrates the easy adoption of IPv6 in a home network. Finally, we discuss the evolving support for IPv6 in the Universal Plug and Play (UPnP^{*}) forum

that facilitates the easy deployment of IPv6 in a home network.

INTRODUCTION

With the rapid growth of endpoints requiring Internet access across the globe, assigning global Internet Protocol (IP) addresses to the connecting endpoints is becoming an increasing problem for Service Providers (SP) due to the paucity of available global IPv4 addresses. The effect is compounded with new devices such as cellular and wireless hosts being enabled to access Internet content. With the available pool of global IPv4 addresses predicted to be exhausted in the near future [1] and in order to meet the growing demands of subscribers, SPs are starting to deploy techniques such as Network Address Translation (NAT) (RFC2663). While the deployment of NAT has not prevented endpoints from seamlessly using common Internet applications such as the World Wide Web, e-mail etc., it has resulted in the breaking of several existing peer-to-peer applications, often resulting in a poor end-user experience.

With broadband Internet deployment on the rise, home networks that connect multiple PCs and consumer Internet appliances that need global Internet connectivity are emerging. Normally, Residential Gateways (RG) are used to multiplex the global Internet connection by assigning private IP addresses to the devices needing Internet access. While the private addresses assigned to these devices are sufficient for in-home networking, they cannot be used by applications outside the home that need access to those devices and their services, due to the private nature of the addresses.

The above issues result in several customer support calls that place an undue burden on the SP's support network.

Another challenge in a home network is to make it easy for the average user to add, install, and configure new Internet appliances. The networking layer is key to facilitating this "plug and play" usage model. The absence of mandated support for auto-address

[™]Intel XScale is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

* Other brands and names are the property of their respective owners.

configuration in IPv4 technology remains a key barrier for plug and play networking, despite the emergence of temporary solutions [2].

IPv6 is the next-generation network layer technology that attempts to solve the problems outlined above. With its 128-bit address width, IPv6 provides a very large address space, enough to provide global Internet addresses to endpoints for years to come. Moreover, it makes routing more efficient by virtue of its hierarchical address architecture. This solves the critical problems of address paucity and ubiquitous access to devices and services from anywhere, anytime. With its built-in design and support for auto-configuration, security, and mobility, IPv6 facilitates easy-to-use, end-to-end secure networking.

While the benefits of migrating to IPv6 networks are clear, the transition itself cannot happen instantaneously due to the involvement of several network elements. The transition from IPv4 to IPv6 networks will span the next several years, during which time existing IPv4 networks will be used to transport IPv6 packets. It is expected that IPv6 technology will initially emerge in customer premises as islands. As such, it will be relatively easy to deploy IPv6-based networks that co-exist with IPv4-based networks both inside and outside the home without affecting existing operations.

In this paper, we first present the problems and pitfalls of deploying NAT in an IPv4 environment and propose IPv6 as the remedy at the network layer. We follow this with a brief discussion of the IPv6 core addressing architecture and its key features. We also discuss the migration from IPv4 to IPv6 networks—the transition architecture and the models that facilitate a smooth transition. Then, we discuss a simple solution that was implemented on an Intel® XScale™ core-based platform, to deploy IPv6 in home networks that co-existed with IPv4 networks. Finally, we discuss the current deployment status of IPv6 and related issues.

IN-HOME NETWORKING

Endpoint Addressing and IP Multiplexing

As the number of homes with multiple PCs increases, the necessity for those PCs to share the Internet connection becomes obvious. With Broadband Internet access becoming more common across the globe, the emergence of special-purpose Internet appliances and digital media devices also necessitates the sharing of the Internet access as the Internet becomes the means to deliver rich, dynamic multi-media content. Figure 1 shows a home network

with two PCs, an Internet appliance, and a printer, all sharing a single Internet connection.

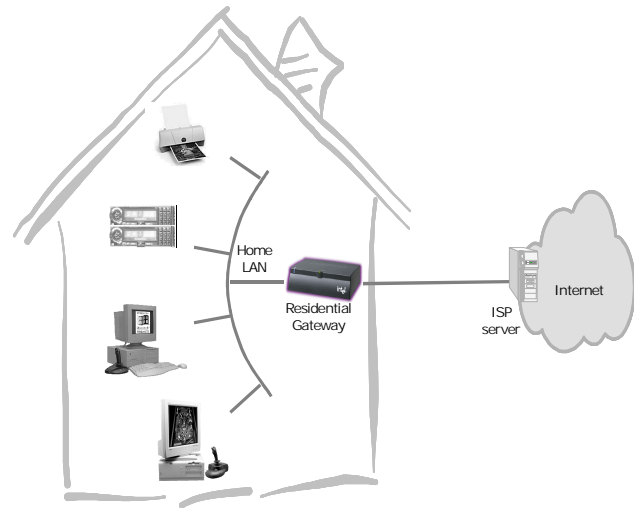


Figure 1: Two PCs, an Internet appliance, and a printer sharing the Internet connection

Since service providers charge more to provision global IPv4 addresses to devices inside the home, consumers often buy Residential Gateways (RGs) to multiplex the single Internet connection they have in their home. RGs assign private addresses to the various endpoints within the home and translate the private addresses to the global address as the packets fly through them. However, this fragments the Internet-enabled endpoints into private address space and public address space.

While such a Network Address Translation-based (NAT) scheme is relatively transparent to the end user, NATs with private addresses do break certain operational semantics that would have been preserved if the endpoints had obtained global IP addresses. Of particular interest are applications that communicate point-to-point by exchanging their IP addresses and port numbers in the IP datagram itself. As the exchanged addresses are private ones, the packet communication between the endpoints breaks down, due to NAT, since private addresses cannot be routed in the Internet. In general, any communication that takes place by the exchange of host addresses may be broken when a NAT is deployed in the path of the communication.

RGs use corrective measures such as Application Layer Gateways (ALGs) to make the private addresses and ports embedded in the packet payload appear to have emanated from the Internet connection of the RG. Since ALGs need to know the application protocol to fix the payload, it is difficult to dynamically create ALGs for new network applications that use unknown protocols.

TM Intel XScale is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

NAT also prohibits standard Internet services from being hosted on more than one endpoint. Since services are often hosted on a combination of IP addresses and ports, simultaneous usage of a port by two applications or hosts is not possible as they share the Internet address. For example, it is not possible to host more than one game server when the game uses a fixed port. While specialized applications can be used to configure RGs to allow service forwarding, it is beyond the average end-user to effectively use such customized applications.

Therefore, even though it is evident that addresses for multiple endpoints inside the home are required, NAT-based private IP addressing is not the correct solution for the following reasons:

- Several peer-to-peer applications that exchange their IP address and port will not work.
- ALGs alleviate the problem by translating the payload of well-known protocols; however, they cannot be extended to new unknown application protocols.
- Hosting multiple identical services behind a NAT is not possible.
- Service forwarding requires knowledge and skills beyond the average user.

While the obvious solution is to assign global IPv4 addresses to every endpoint, the cost of IP addresses in conjunction with the very limited pool of available IPv4 addresses, where cost is not a concern, prevents this from being the solution.

IPv4 Protocol Issues

The lack of global IPv4 addresses now is often cited as the result of the less-efficient design of the IPv4 addressing architecture. In particular, the “class A” IPv4 addresses, which constitute half of the address space, were given to a small number of organizations. This resulted in an uneven distribution of addresses and underutilization of the overall IPv4 address space.

Further, the lack of inherent support for secure end-to-end communication and the sub-optimal support for mobility make IPv4 less suitable for modern communication patterns. These features were added later as quick-fix solutions. The support for implementing Quality of Service (QoS) was very coarse-grained and for all intents and purposes it resulted in the underutilization of the feature. Finally, communication anonymity could not be added into the protocol; instead it was added at a higher layer such as application tunnels.

IPV6 TECHNOLOGY

The next-generation network layer protocol, IP version 6 (IPv6) provides a remedy for the problems found in IPv4 as follows:

Global Internet Address: By provisioning a very large address space with 128 bits for every IP address, IPv6 can assign global addresses to every Internet endpoint for several decades.

Plug and Play Networking: By connecting to a network, an IPv6-enabled endpoint automatically acquires IPv6 addresses. Multiple addresses could be assigned to an interface in different realms—local addresses and global addresses. The scheme has also been made flexible enough to re-address endpoints quickly if necessary.

Better Quality of Service (QoS) Support: IPv6 provides better support for fine-grained QoS. This facilitates better delivery of multi-media data.

Mobility, Anonymity, and Security by packet encryption (IPSec) have also been built into the protocol itself. It has been mandated that security must be supported by every IPv6 implementation.

IPv6 Core Architecture

The IPv6 provisions (RFC2460) a small protocol header where the essential information is stored and allows dynamic extensions to it when necessary. This is shown in Figure 2 below.

| | | | |
|---------------------------|----------------------|--------------------|------------------|
| Version (4) | Traffic Class (8) | Flow Label (20) | |
| Payload Length (16) | | Next Header (8) | Hop Limit (6) |
| Source Address (128) | | | |
| Destination Address (128) | | | |

Figure 2: IPv6 core address architecture

The core header looks similar to that of IPv4 with the exception that the number of fields is reduced. Except for the flow label field, all fields functionally correspond to similar fields in IPv4. Additional header information can be dynamically added as extension headers by using an indirection field (Next Header) in the core header that points to a chain of extension headers.

IPv6 technology classifies every address into one of three types: Unicast, Multicast, and Anycast.

Unicast and Multicast addresses are comparable to their IPv4 counterparts. A Unicast IPv6 address is the address

assigned to an interface in a host. It differs from IPv4 in that an interface can have one or more IPv6 Unicast addresses assigned to it.

Multicast addresses are usually assigned to a set of interfaces, usually a set of hosts. Every packet destined for the Multicast address is received by every node/interface in the set. The delivery semantics are similar to that of IPv4.

The new class of addresses in IPv6, the Anycast address, is an address that is assigned to a set of interfaces/hosts. However, a packet destined for the Anycast address is delivered to at least one of the interface hosts as defined by a set of criteria (usually governed by the underlying routing protocol). We discuss the global Unicast address in detail as it is the most relevant address for an interoperable home.

IPv6 Global Unicast Address

The structure of a global IPv6 Unicast address is shown in Figure 3. The 128-bit native IPv6 address is hierarchically partitioned as follows:

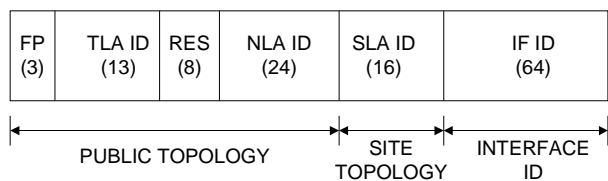


Figure 3: IPv6 Unicast address structure

FP – 3-bit *Format Prefix field*. This indicates the type of IPv6 address. At the time of this writing, all native global IPv6 Unicast addresses will have a value of 001.

Top-Level Aggregator (TLA). This is a 13-bit identifier used to denote the infrastructure provider.

RES. This is an 8-bit field reserved for IANA usage.

Next-Level Aggregator (NLA). This is used to denote the different entities the infrastructure provider services.

Site-Level Aggregator (SLA). This is used to identify one of the different partitions in a given site.

INTERFACE-ID This is the interface identifier used to identify a unique host. This is referred to as the Extended Unique Identifier (EUI), and normally an extended form of the MAC address is used.

In general, the 64 bits to the left are used to identify the logical subnet to which the interface belongs, while the 64 bits to the right are used to identify the interface in the subnet. Often the subnets are represented in a CIDR-like notation (RFC1519).

The key features of the IPv6 architecture are the large address space and the routing efficiency resulting from the hierarchical organization.

IPv6 Address Auto-Configuration

One of the mandated requirements of the IPv6 protocol is that it be able to automatically acquire an IPv6 address to instantly communicate in the network neighborhood. This is facilitated by making every host implement an interface with the special local IPv6 address as described in RFC2373. These are referred to as link-local addresses and can be used to communicate only in the local subnet. Additionally, a host can configure itself with addresses using any route advertisement appearing in the subnet to which it is connected. Usually, IPv6-enabled routers send such advertisements so that hosts can self-configure.

Transition Address Architecture

For existing IPv4 endpoints running an IPv6 protocol stack, several compatibility and transition architecture addresses have been provisioned. The important transition architecture address involves assigning the special TLA-ID of 2. Implied in almost every global Unicast IPv6 address that has this special TLA-ID is an IPv4 address in the next 32 bits (bits 16-47) and an arbitrary number in its SLA to form a unique subnet prefix **2002:<IPv4>::/48**. This prefix and the EUI combine to form a unique global IPv6 address that enables an IPv4 host to communicate via the Internet.

These are referred to as *6to4 addresses* (Figure 4), and the hosts that acquire 6to4 addresses based on their global IPv4 address are referred to as 6to4 hosts. Normally, such hosts require the implementation of both IPv4 and IPv6 networking protocol stack support in their operating system and are called *dual-stacked hosts*.

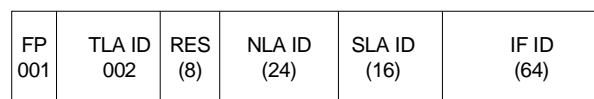


Figure 4: IPv6 6to4 address

THE APPROACH TO TRANSITIONING TO IPV6

The necessity to adopt and natively support IPv6 across the entire Internet involves the following:

- Every host that accesses the Internet needs to support IPv6 at every layer.
- The edge network, a collective term for both the home network and the provider network, needs to support IPv6, i.e., both intra-home and home-to-provider communication must support IPv6.

- The core backbone of the Internet connected to the provider network must support IPv6.

As you can see, the transition to an IPv6 network involves a great deal of effort in terms of cost, time, and planning. The migration to IPv6 will happen incrementally only over a period of time; it won't happen with a simple flip of a switch. During the transition, it is expected that all IPv6 communication will be transported as payload in existing IPv4 networks. This technique is called *tunneling*, or *tunneling IPv6 over IPv4*, as shown in Figure 5.

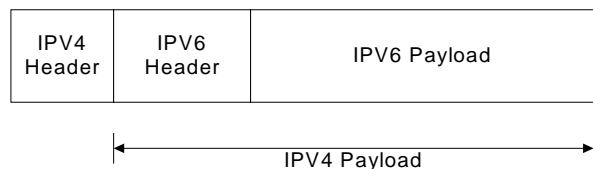


Figure 5: IPv6 packet tunneled in an IPv4 packet

We expect the transition to evolve and emerge initially in the home network where IPv6 deployment is relatively simple. This will result in the creation of islands of IPv6-enabled homes that are connected to the Internet over IPv4 networks. During this period, applications will be IPv6-enabled but will preserve their IPv4 connectivity for maximum Inter-networking. We also anticipate that host applications will automatically initiate IPv6 connectivity and use it if available. Otherwise, IPv4 will be used for connectivity. Such IPv6 islands will expect minimal assistance from the service providers to resolve host names to IPv6 addresses. We also expect communication between the IPv6 islands to happen over the IPv4 Internet by the use of tunneling.

As the transition progresses and more Internet content is delivered using IPv6 as the preferred protocol for connectivity, networked homes are expected to use IPv6 on a regular basis. Increased usage and transport of content using IPv6, although over IPv4 networks, coupled with the reduced support burden, will serve as incentives for the service providers to deploy IPv6 natively in their networks. Similarly, as more traffic gets transported over IPv6, the core network providers will start deploying IPv6 in their networks alongside IPv4 networks resulting in a dual-stacked core network.

The above two actions would help establish IPv6 as the mainstream network protocol, while IPv4 usage diminishes. It is important to note that, for IPv6 to become the mainstream networking protocol, host operating systems need to incorporate dual networking stacks, and most network applications need to be modified to make use of IPv6 connectivity if present.

IPv6 TRANSITION ARCHITECTURE FOR HOME NETWORKS

In general, the transition architecture is classified into two broad categories: tunneling and translation.

The tunneling models transport IPv6 datagrams over IPv4 as the protocol payload. The translation models involve translating the IPv4 headers into IPv6 headers and vice versa as the packets fly through a translation device.

During the initial stages of the transition, we expect the tunneling models to prevail over the translation models due to a lack of standards and the problems associated with translation. However, it is to be noted that wherever IPv6 support is absent or when heterogeneous communication has to happen, such as an IPv4-only host to an IPv6-only host, the network has to deploy translation techniques such as NAT-PT(RFC2766).

6to4 Tunneling

Several tunneling techniques have been proposed to transport IPv6 over IPv4 [1]. All the tunneling schemes work as long as the applications are IPv6-enabled and communicate end-to-end using IPv6. To support the applications, an endpoint must implement IPv4/IPv6 dual stack to make IPv4 the transport layer for the IPv6 communication.

Of the several tunneling techniques, the important ones are the (manually) configured tunnels and automatic tunnels. Automatic tunnels are the preferred choice as they leverage existing global IPv4 addresses to set up the tunnel. Of particular interest are the 6to4 automatic tunnels (RFC3056). Since the 6to4 tunneling scheme uses an existing IPv4 address to create IPv6 subnets, it is an important element of the transition architecture. In fact, the transition model that is important for IPv6 deployment during the transition in a home uses 6to4 transition techniques. We briefly discuss the 6to4 automatic tunneling scheme below and follow it up with our implementation.

The 6to4 model uses the ISP-assigned global IPv4 address (32 bit) to obtain the special prefix **{2002:IPv4 address}** and uses this global 48-bit prefix to advertise to all the devices in the home network. Any IPv6 device inside the home automatically acquires a global 6to4 IPv6 address, thus becoming a 6to4 host.

The communication between two 6to4 hosts in the Internet is relatively straightforward as each host can tunnel the IPv6 packets using their respective IPv4 connection. However, the communication between a 6to4 host and a native IPv6 host travels through a tunnel to a well-known router, called the *relay router*. The relay router in turn routes the IPv6 packet in the native IPv6 cloud. Usually

the relay routers are assigned a fixed Anycast address so hosts can route the native IPv6 payload to them over the IPv4 network. This is illustrated in Figure 6 below: In this figure, the RG itself is the 6to4 host communicating with other 6to4 hosts.

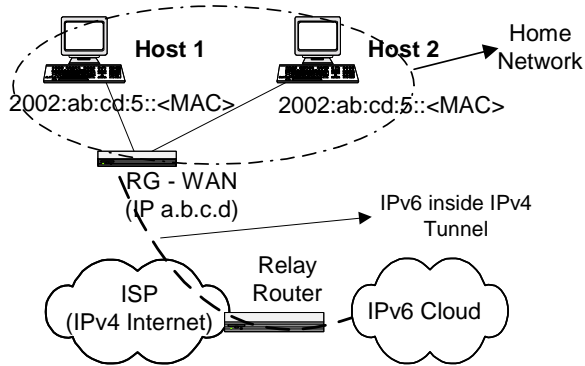


Figure 6: 6to4 tunneling

The communication back from the native IPv6 cloud to the 6to4 host is routed by the nearest dual-stacked router and transported over an IPv4 network back to the host and not necessarily via the relay router.

Translation Model

In the translation model, the IPv4 packets are translated into IPv6 packets (and vice versa) using specialized translation schemes [1]. Usually such translations are more specific to the protocol headers and are performed by network elements dedicated to such purposes. Thus, these translation schemes are transparent to network applications and usually do not require the application to change. Since translation schemes rewrite headers, they are susceptible to the same problems as ALGs. The disadvantages of the translation schemes outweigh the advantages, and we do not use them unless there is a compelling necessity to do so.

Sometimes, the translation schemes have to be used in conjunction with the tunneling schemes such as in the case of some legacy hosts that do not support dual IP stacks.

Requirements for Home Network Transition

The key to successful implementation of the above transition architecture is as follows:

- Hosts should be dual-stack-enabled to transport IPv6 packets over their IPv4 layer.
- ISPs should have the ability to support at least one global IPv4 address to the home.

- An always-on device, usually Residential Gateways (RGs), should be present to administer the model in the home network without additional intervention.
- The endpoints should initiate IPv6 tunnels such as Teredo [4] when an IPv6 supporting RG is not present or an ISP cannot provision a global IPv4 address.

OUR IMPLEMENTATION OF THE IPV6 TRANSITION

Internet appliances such as Residential Gateways (RGs) are important for the IPv6 transition inside the home as they always remain powered on and can make the route advertisement service available to the home network. Additionally, the RG is responsible for the following:

- It must automatically convert the global IPv4 address assigned by the ISP to an IPv6 global prefix and announce it in the home network.
- It must take the native IPv6 packets emanating in the home and tunnel them to the relay router or route them natively in the Internet.
- When heterogeneous communication happens (IPv4 to IPv6 networks or vice versa), the RG must translate the packets, if necessary.

Due to this important role of RGs, we decided to use an Intel® XScale™ core-based reference platform (IOP80310) and make it an RG with IPv6 routing capability. This platform was chosen because of its suitability in an embedded environment. The platform has an on-board Ethernet port and two expansion slots. We used the on-board Ethernet port to connect to the home network and one of the expansion slots to connect to the Broadband Internet.

Due to its immediate availability for the IOP80310 platform with source code, we chose Linux* as the operating system to implement our solution in this platform.

We used the IPv6 code base from the USAGI project [3] and ported it to the Intel XScale core-based platform. The resultant IPv6 Linux kernel was used along with some software tools, which were required to configure static IPv6 routes and default rules for routing in the kernel

[™] Intel XScale is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

* Other brands and names are the property of their respective owners.

routing table. We also ported the route advertisement daemon to the IOP80310 platform to announce the 6to4 global prefix in the home network.

Home Network Setup

We used the IOP80310 reference platform as the residential gateway (RG). The on-board Ethernet port was connected to a four-port hub, so the home network computers and devices could connect to the RG. We used an Ethernet card on one of the expansion slots to connect and simulate a broadband Wide Area Network (WAN) connection.

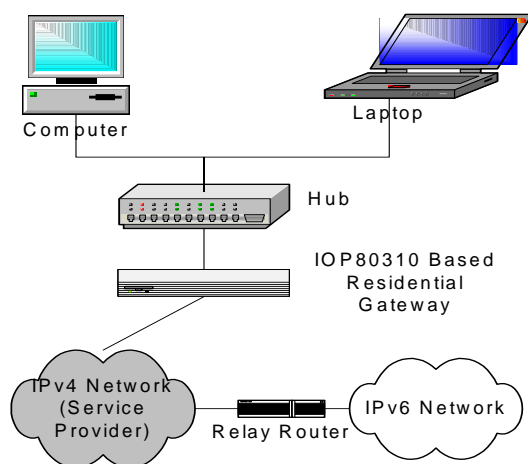


Figure 7: Home network setup of our implementation

We used a couple of computers running Windows XP* and Windows 2000* operating systems to connect to the hub and create a home network environment. Both computers were configured to support dual IPv4/IPv6 stack. This setup is illustrated in Figure 7.

Software Setup

We used the Linux kernel, ported to the IOP80310 platform and configured it to provision the functions usually found in RGs such as NAT etc. This enabled all clients to run all existing Internet applications by sharing the broadband connection. The RG was assigned a static IPv4 address on its WAN interface and connected to the Internet.

The IPv6 software stack was added as a loadable driver in the RG. The RG was configured to create the 6to4 global prefix from its static IPv4 address and announce it as the global subnet prefix in the home network with the RG as the router for both IPv4 and IPv6 packets.

Additionally, the RG's initialization software created a IPv4-based tunnel to the standard relay router so that all non-local IPv6 traffic can be routed to the relay router. Entries in the routing table were created to provision this. Additionally, specific routing table entries were also created so that all IPv6 traffic destined for the home network clients coming from the Internet are properly routed by the RG's IPv6 stack back to the home network. Our service provider had provisioned a DNS server that supported name resolution to both IPv4 and IPv6 records so applications could take advantage of it.

End-to-End IPv6 Connectivity

To test end-to-end IPv6 connectivity and internetworking over the (IPv4) Internet, we used an IPv6-enabled version of the popular PC game Quake* and used it to connect to an IPv6-enabled Quake server over the Internet. Even though the client and the server were communicating over IPv6, the packets were actually transported over an existing IPv4 network.

We were able to successfully connect to the Quake server and execute the game validating our end-to-end internetworking vision.

THE CHALLENGES OF TRANSITIONING TO IPV6 IN THE HOME

From an implementation standpoint, the key challenge faced, in our experience, was the inability to address endpoints using the Fully Qualified Domain Names (FQDN) of the endpoints. There were two reasons attributed to this: a) a lack of standards to dynamically discover the IPv6 Domain Name Server (DNS); and b) the inability to dynamically register the name with the DNS server so peers could communicate end-to-end without having to know the IPv6 address. Standards are still evolving in the Internet Engineering Task Force (IETF) to solve this barrier to IPv6 deployment.

The other key challenge we faced is very specific to the transition mode. By using 6to4, we compromised one of the distinguishing features of IPv6 over IPv4 i.e., communication anonymity. By virtue of using the global IPv4 address as part of the prefix, we simply ended up with the same problem existing in IPv4, i.e., the traffic from an endpoint could be tracked based on the IP address.

From a deployment and ease-of-use perspective, RGs play a crucial role as they function at the junction of both the home and the service provider networks. As RGs hold the key to enable IPv6 services inside the home, the clear challenge is to be able to configure them to adopt the transition model in the provider network. In the following

section, we discuss the role of Universal Plug and Play (UPnP*) technology in enabling IPv6 in-home networks.

IPv6 AND UNIVERSAL PLUG AND PLAY

Universal Plug and Play (UPnP*) technology provides a control protocol to easily install, configure and control devices and appliances used in small and home networks. In a home network with different kinds of digital devices present, UPnP technology holds the key to facilitating ease-of-use leading to rich end-user experience. In this section, we discuss the design aspects that influence the easy deployment of IPv6 using UPnP technology. To use IPv6 in conjunction with UPnP technology, there are two aspects that have to be considered.

1. UPnP technology currently uses IPv4 as the underlying network layer for all communication. To use IPv6 as the network layer, extensions and modifications to the UPnP protocol and specifications are needed. Currently, this is being reviewed by the UPnP Forum.
2. Residential Gateways (RGs) fall under the category of Internet Gateway Devices (IGDs). In order for IGDs to support IPv6, and administer IPv6-related functions used in the home network, standardized means of configuring the IGDs using the UPnP protocol are needed.

To successfully enable IPv6 addresses for clients inside the home, the IGD should have the ability to be configured using the UPnP protocol to support the following:

- A default configuration of the UPnP IGD should automatically detect all the IPv6 services present in the attached service provider network and relay those services locally to make them available to all the devices in the home network. In the absence of IPv6 services in the provider network, an RG should have the ability to be configured to support one of the transition models, with the preferred automatic tunneling model being 6to4. The RG should supplementally act as an IPv6 router so that IPv6 packets from/to the home are properly handled as they flow to and from the provider network.
- Endpoints such as PCs should detect the presence of the IPv6 capabilities of an UPnP IGD and use them. In the absence of IPv6 support in the network, the PCs should initiate techniques such as Teredo to enable their IPv6 stack.

* Other brands and names are the property of their respective owners.

THE STATUS OF IPV6 DEPLOYMENT IN THE HOME

At present, most modern operating systems including Windows* XP*, and Windows 2000* (clients and servers), all flavors of Unix* including Linux*, and several embedded operating systems support dual-stacked networking protocol stacks. While the Windows operating systems support them, they are not enabled for usage automatically. Integrated support for IPv6 in the operating system has been included in the test releases of future versions of the Windows* operating system such as Windows XP-SP1*.

Residential Gateway (RG) vendors are starting to show an interest in incorporating dual-stack operating systems in their access devices. From the perspective of UPnP technology, several vendors including Intel are participating in the UPnP forum to standardize IPv6 implementation in RGs. Initial Intel participation has resulted in a set of guidelines for IPv6 implementation in RGs. Standardization in the UPnP forum will further solidify support for IPv6 in the RG community.

Core network equipment vendors such as Hitachi have started to add IPv6 support in the operating platforms and environments of their infrastructure equipment, such as routers [1].

CONCLUSION

In a broadband-enabled digital home, several Internet appliances and digital devices are expected to be present. IPv4 Network Address Translations (NATs) would only result in limited end-user experiences when using those devices. Current solutions based on IPv4 are mere patch-work solutions. For easy networking inside and outside the home, IPv6 is emerging as the preferred next-generation protocol for communication in the Internet. The design of the entire gamut of Intel's products that participate in the digital home including PCs, Internet appliances, and digital home products should be enabled to use IPv6 as the protocol of choice.

ACKNOWLEDGMENTS

I thank my colleagues in the Corporate Technology Group, Intel, for exchanging and sharing ideas, views, and opinions while writing this paper. In particular, I thank Vijay Rao for supporting the IPv6 work on Intel® XScale™

* Other brands and names are the property of their respective owners.

™ Intel XScale is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

core-based platforms. Sincere thanks go to Christopher Lord and Ulhas Warriar for several fruitful discussions that laid the foundations for the arguments presented here. Many thanks go to Yasser Rasheed and Charlie Tai for their encouragement.

REFERENCES

- [1] D.G. Waddington and F Chang, "Realizing the transition to IPv6," *IEEE Communications Magazine*, June 2002, pp. 139-144.
- [2] Stuart Cheshire and Bernard Aboba, "Dynamic Configuration of IPv4 Link-local Addresses," *IETF Draft*, March 2001.
- [3] USAGI Project. <http://www.linux-ipv6.org/>
- [4] C Huitema, "Teredo: Tunneling IPv6 over UDP through NATs," *IETF Draft*, Sep 2002.

AUTHOR'S BIOGRAPHY

Venkat R Gokulrangan is a Senior Software Engineer in the Desktop Platform Group. While at Intel Labs, he initiated the work on IPv6 for Residential Gateways using Intel platforms. Venkat actively participates in standards activities for easy IPv6 adoption in residential networks. His work resulted in proposing standards to the IGD subcommittee in the UPnP forum. Venkat has designed and implemented data subsystems such as 802.1D bridges and voice subsystems such as VoDSL software. He has an M.S. degree in Computer Science and Engineering from the University of Michigan, Ann Arbor. His e-mail is venkat.r.gokulrangan@intel.com

Copyright © Intel Corporation 2002. This publication was downloaded from <http://developer.intel.com/>.

Legal notices at <http://developer.intel.com/sites/developer/tradmarx.htm>.

For further information visit:

developer.intel.com/technology/itj/index.htm