IPv6 and Multicast Filtering for High-Performance Multimedia Applications

Thomas Pike*, Craig Russell†, Alex Krumm-Heller†, Vijay Sivaraman*
**School of Electrical Engineering and Telecommunications, University of New South Wales, Australia*
†ICT Centre, CSIRO, Australia

***Abstract*—It is widely acknowledged that the IPv4 address space will be close to exhaustion within the next few years, and the future growth of the Internet increasingly depends on the timely deployment and availability of IPv6. Both the Internet Corporation for Assigned Names and Numbers (ICANN) and the Australian Government have recently issued statements pushing for the adoption of IPv6. This paper reports on our experiences in migrating CSIRO's multicast-based high-performance multimedia platform, currently used in medical conferencing and remote education applications, to IPv6. Application code changes, along with network equipment configuration, are discussed. Additionally, to optimise network bandwidth and end system resource usage, we implement and compare three multicast filtering techniques: at the application, in the OS kernel, and in the network via source specific multicast (SSM). We show that the integrated multicast support in IPv6 allows high-performance multimedia applications to be enhanced with minimum effort when migrating to IPv6.**

I. Introduction

Since the early 1990's when IPv4 address exhaustion was first considered, there has been a concerted effort by the Internet industry to develop an alternative protocol with the goal of one day replacing IPv4 altogether so that address shortage would never again be a problem. As early as 1992 Huitema [1] describes writing a draft recommendation after leaving the Internet Society (ISOC) and Internet Architecture Board (IAB) conference in Kobe, Japan. To Huitema and others it was clear that a new version of the Internetworking Protocol was required, *"[the] Internet was in great danger of running out of network numbers, routing tables were getting too large, and there was even a risk of running out of addresses altogether"*. The new protocol, named IPv6, was published in 1995 as RFC 1883 [2] and refined in 1998 as RFC 2460 [3].

The major router vendors have supported IPv6 for several years and while initially much of this support was confined to software and not in the high speed hardware-based forwarding plane, our recent experience with Cisco Systems and Nortel Networks devices has been that IPv6 is fully supported in hardware with switching performance the equal of IPv4 ie. at multiple gigabits per second. Similarly, Miller [4] notes that most major computer hardware and software vendors; Apple, Hewlett-Packard, Hitachi, IBM, Linux, Microsoft, Novell and Sun support IPv6 to differing extents. The launch of Microsoft Windows Vista provides the first large scale implementation of an end user configurable IPv6 stack together with two IPv4 to IPv6 transition technologies (Teredo and 6to4 tunnelling).

Technological developments have led to the increasingly widespread use of consumer devices that require Internet connectivity (PDAs, laptop computers, home entertainment systems, even refrigerators), and coupled with the rapidly expanding broadband networks being deployed by ISPs around the world the number of devices requiring IP addresses is increasing rapidly. Several analysts such as Hain [5] and Huston [6], [7] have attempted to estimate the date when the last IPv4 address blocks will be allocated by the Regional Internet Registries (RIRs) and current projections are that it will be sometime around 2010 or 2011 [8].

IPv6 is much more than an attempt to provide a protocol to overcome address shortage. IPv6 seeks to incorporate the wisdom of the last twenty five years of networking experience, drawing on the rich experimentation and innovation which has occurred. Worthy extensions to IPv4 have been merged into the core of IPv6 whilst obsolete components have been discarded.

King et al. [9] identify five key areas which were considered during specification of IPv6:

- Addressing and routing
- Eliminating Special Cases
- Minimizing Administrative Workload
- Security
- Mobility

There is a discussion in [5], however, that suggests that the IPv4 address space will never in fact entirely deplete, rather that the end game for IPv4 should be considered to have occurred *"at the point at which one needs to start designing networks and subnets, not in a way that is optimal from a network architecture or network management and growth standpoint, but in order to conserve address space"*. The discussion continues to state that this point has already occurred, networks are no longer being designed in a manner that is optimal from an architectural sense, rather address conservation is key, witnessed through the introduction of technologies such as NAT. In the light of these facts, it may seem strange to the network scientist or engineer that the transition to IPv6 hasn't occurred sooner or more rapidly. Reasons for this are many and are equally due to economic and policy reasons as much as technical or engineering considerations. Huston [7] provides a substantial discussion of some of these issues.

In July 2007 the Internet Corporation for Assigned Names and Numbers (ICANN) adopted a board resolution [10] to work with the RIRs and other stakeholders to promote the timely deployment of IPv6, while in the same month the

Australian Government Information Management Office (AGIMO) presented its plan towards 2010 [11] that investigates the application of IPv6 within government.

Our experience within the CSIRO is typical of many large R&D organisations with regard to IPv6; namely, the chicken-and-egg problem. Network engineers cite the lack of applications doesn't warrant the time and expense of deploying IPv6 enabled networks. At the same time application developers claim the lack of IPv6 network deployment hampers their development of IPv6-based code. The purpose of this paper is to provide an overview of our experiences in transitioning an advanced multimedia application to IPv6 and to highlight some of the issues encountered.

II. Background

This section briefs the reader on the high-performance multimedia platform based on the Virtual Tearoom^{TM} Technology we have developed at CSIRO, along with some background on multicast operations in IPv6.

A. The Virtual Tearoom^{TM} Platform

The Virtual Tearoom^{TM} is a framework developed in-house at CSIRO for multi-site videoconferencing applications. Written in C++, its core consists of around $20,000$ lines of code that co-ordinate communication and usage of resources, with a further $65,000$ lines of code that handle streaming of audio and video, sharing of files, synchronised viewing of presentations, white-boarding, and chatting. The platform is currently being used at hospitals for medical conferencing applications, at universities for remote education, and at CSIRO for multimedia meetings between the various sites around Australia. Current deployments operate over several interconnected networks, including the ICT Centre's research network (built and operated by us), CSIRO's production network, and Australia's academic and research network (AARNet3).

Each end-point can generate multiple digital video (DV) streams, each at 30 Mbps, packetised and transported over any IP network. Messaging between sites in a tearoom session is carried in UDP packets. Although initial communication between sites is via unicast packets, the framework switches to multicast communication as soon as it is able to negotiate an acceptable multicast group address. Thereafter all data and control messages are sent via multicast. Additional sites are added to a session if they attempt to contact any of the participants of an active session, or if the participants attempt to contact them. Sites maintain their presence in a session by sending periodic keep-alive messages. During the connection process each site makes multicast announcements on its available resources to allow other conference participants to select the feeds they want to subscribe to.

B. IPv6 Multicast

Multicasting is an inherent part of the design of IPv6, so much so that there is no broadcast capability available and multicasting is used rather than broadcasting to minimise the impact of solicitations, advertisements, updates and so forth on multicast-capable links. IPv6 multicast routing operates conceptually in much the same way as IPv4 multicast routing. Receivers that wish to receive data belonging to a particular group must join the group by signalling the local router. In IPv4 this signalling is done using IGMP whereas in IPv6 it is done through the MLD (Multicast Listener Discovery) protocol. MLD uses ICMP to carry its messages and all MLD messages are link local in scope with a hop limit of 1. MLD is used by IPv6 routers to discover multicast listeners on directly attached links. There are two versions of MLD: MLD version 1 [12] is based on version 2 of IGMP for IPv4, and MLD version 2 [13] is based on IGMP version 3 for IPv4. MLD version 2 is fully backward compatible with MLD version 1. MLD version 2 is required for source specific multicast (SSM) [14] as IGMPv3 is required for SSM in IPv4. PIM-SM (Protocol Independent Multicast - Sparse Mode) is used between routers as a multicast routing protocol in the same way as IPv4 and PIM in SSM mode is used to forward packets exclusively on source-based trees. An IPv6 multicast address is an IPv6 address with a prefix of FF00::/8 while the prefix FF05::/16 is a multicast address with site scope, similar to 239.255.0.0/16 in IPv4.

III. Implementation

A. Deploying and Configuring a Test Network

In order to properly test the multicast application in IPv4 and IPv6, a network testbed was built in the Marsfield laboratory of the CSIRO ICT Centre. The network architecture was designed to represent a typical enterprise deployment with sufficient links and complexity to demonstrate the behaviour of the multicast filtering techniques. A diagram of the physical architecture is shown in figure 1.

The physical architecture of the network comprised a core of three routers R1, R2 and R3 connected via gigabit ethernet links to a switch S4. Each of the three routers also had a fast ethernet link to an access switch, R1 to Sl, R2 to S2 and R3 to S3. Each of the switches S1, S2 and S3 additionally had a fast ethernet link to a PC and to an interface on an IXIA 400T. The IXIA is a specialised hardware traffic generator and analyser. It is capable of generating arbitrary packet streams, including unicast, multicast, IGMP, etc., and analysing packet statistics including data rates, latencies, etc. at high data rates. Our tests below rely on its capabilities to emulate a large number of multicast sources.

The routers R1, R2 and R3 were Cisco 3825 models running the Advanced IP Services image of IOS version 12.4(12a). Switches S1, S2 and S3 were Cisco Catalyst 3560-24TS models also running the Advanced IP Services image of IOS version 12.2(25)SEE1. The Advanced IP Services image is required for the IPv6 features used in the testing. Switch S4 was a Nortel Networks ERS8600 running version 3.5.10.0 of Nortel's ethernet routing switch operating system.

In terms of the logical configuration of the network, IEEE802.1Q tagging was used on the links from routers R1-R3 to switch S4 so that two sub-interfaces could be configured on each router's physical g0/0 interface thereby allowing a separate routed connection to each of the other two routers.

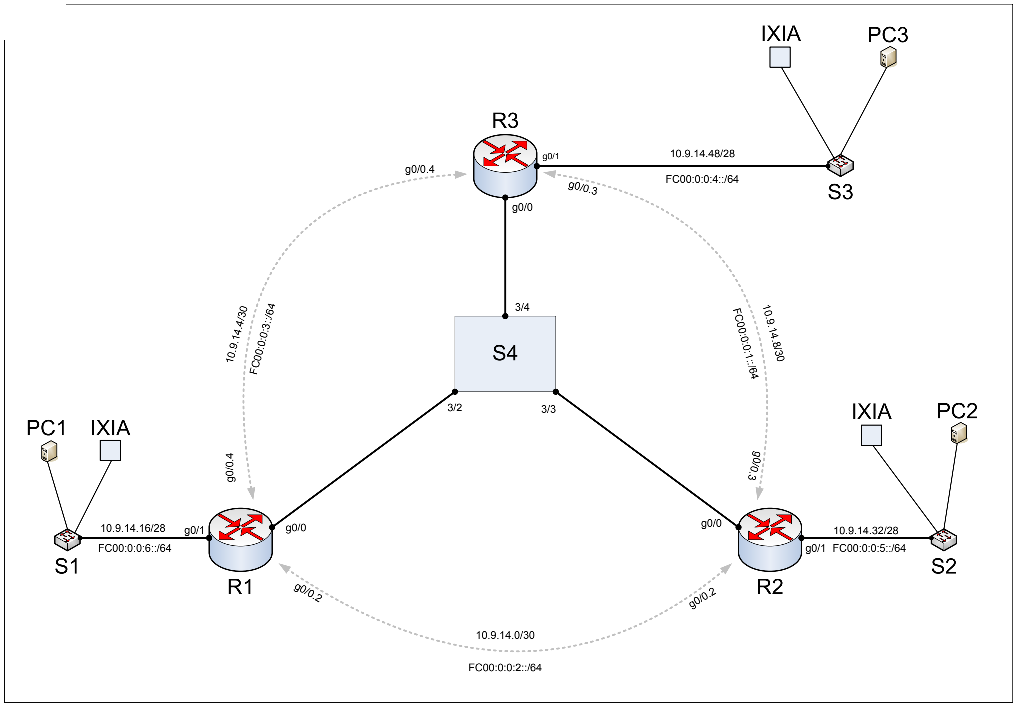

Fig. 1. IPv6 enabled testbed.

These interfaces are marked as g0/0.2, g0/0.3 and g0/0.4 in the diagram and the logical links are shown as dotted lines. The numbers 2, 3, and 4 are the 802.1Q tags used and are thus vlan numbers in switch S4. Switch S4 operated purely as a layer 2 device to facilitate the logical connectivity. In this way a triangle geometry was configured in the routed core providing more than one path between each PC and hence allowing the unicast routing protocol to calculate a shortest metric path and the multicast routing protocol to construct a shortest path distribution tree. Each router R1-R3 was configured with both IPv4 and IPv6 unicast and multicast routing enabled. The links between the routers had /30 IPv4 and /64 IPv6 subnets configured and the g0/1 interfaces to the access switches were configured with /28 IPv4 and /64 IPv6 subnets. For unicast routing, OSPFv2 for IPv4 and OSPFv3 [15] for IPv6 was used (OSPFv3 is based on OSPFv2 but with some significant enhancements specifically for routing IPv6) while for multicast routing PIM sparse mode was used for both IPv4 and IPv6 with SSM configured for specific multicast address ranges. The PIM boot strap router (BSR) method was used in both IPv4 and IPv6 to establish the RP (rendezvous point) address for each multicast group in the non-SSM range for forwarding of packets before the source tree is established. The full details of the configurations used in the test network have not been given since they are not relevant for the purposes of this paper. Switches S1-S3 operated in layer 2 mode only with the interfaces connected to the PC, the IXIA and the corresponding router in a vlan configured with both IGMPv3 and MLDv2 snooping enabled. This particular architecture and configuration enabled us to test all the relevant features of IPv4 and IPv6 multicast source filtering in a realistic deployment scenario.

B. Migrating the Application

A substantial amount time and effort was spent in porting the network libraries and associated test applications of the Virtual Tearoom^{TM} platform to IPv6. A video test application, with architectire shown in figure 2, was built on the Microsoft Windows DirectShow framework with the network libraries implemented as a pair of DirectShow filters. The two underlying libraries *transmitterlib* and *receiverlib* respectively encapsulate and extract DV video from an IP stream. These two libraries were originally IPv4 dependant and did not implement any form of multicast source filtering. Initial work on the migration to IPv6 involved rewriting IPv4 specific code to use protocol independent data structures and interfaces in accordance with RFC3493 [16]. The migration was relatively painless, with some issues encountered in removing the IPv4-specific parts of the code. This mainly involved the handling of addresses; rather than using length specific types to represent addresses we coded IPv6 addresses as strings, thereby avoiding problems with 128 bit values. Additionally some new socket calls were required to control multicast functionality in IPv6 that were not present in IPv4. As part of the code clean-up during migration, it was thought prudent to include multicast filtering techniques so the receiver could reject interleaving multicast packets from other (potentially malicious) sources. Three methods of multicast filtering were evaluated:

- **Application Filtered Multicast (AFM):** Packet filtering is performed at the application layer. The node will join an Any Source Multicast on the group, denoted by (*,G).

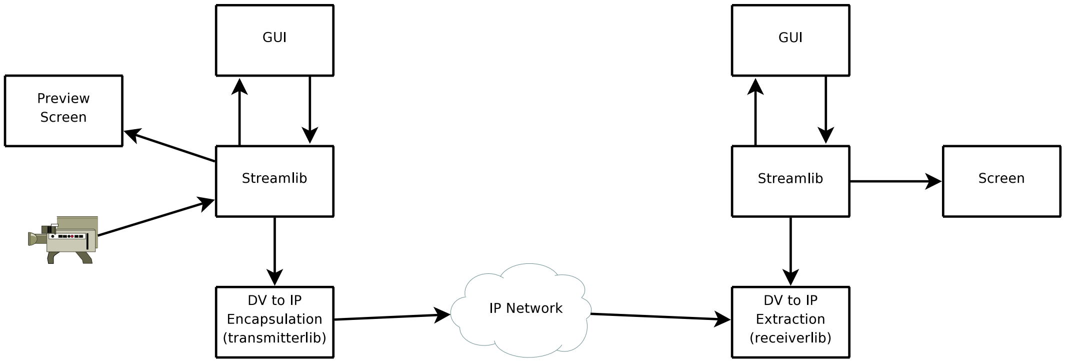
GUI
Preview Screen
Streamlib
DV to IP Encapsulation (transmitterlib)
IP Network
GUI
Streamlib
Screen
DV to IP Extraction (receiverlib)

Fig. 2. Test Application

Unsolicited packets will be inspected by the application. A packet will be forwarded to the DV filter only if it is from the desired source, otherwise the packet is dropped. This method of filtering can be expected to make inefficient use of bandwidth and processing resources as all traffic on the multicast group G is received by the node, passed from the kernel to the user space application and then dropped.

- **Source Filtered Multicast (SFM):** RFC3678 [17] defines an API by which an application can instruct the operating system kernel to filter incoming multicast packets based on the source address. Like Application Filtered Multicast, the node will join (*,G) and hence receive all packets addressed to that group, but the kernel will drop the multicast traffic from sources other than the expected one. Under this method unsolicited multicast packets still arrive to the host, but are dropped by the kernel instead of being processed by the application in user space, thereby reducing the processing load on the host.
- **Source Specific Multicast (SSM):** Source Specific Multicast [14] allows an application when joining a multicast group to specify a set of hosts from which it wishes to receive data on that multicast group. A host then signals desired source-specific group membership, ie (S,G), through IGMPv3 (for IPv4) or MLDv2 (for IPv6) to its first-hop router. Routers can exchange SSM information using the PIM-SM routing protocol. SSM thus allow packet filtering to be performed at routers in the network, thus reducing the bandwidth and processor resource requirements at end-host receivers. However, SSM support is not widely deployed today in end-hosts, and hence rarely used in IPv4-only networks.

Initial attempts to implement these filtering techniques were hindered by the discovery that Microsoft Windows XP does not support MLDv2 which is necessary for IPv6 SSM. The development was then shifted to the Windows Vista platform. This actually eased the implementation, since Vista implements the majority of the IPv6 sockets API (though some key interfaces are missing or not publically documented, and we had to resort to proprietary Microsoft interfaces). We feel that overall the migration improved the quality of the code.

C. *Test Strategy*

The test strategy was as follows. One PC transmitted a 30 Mb/s video stream to a particular multicast group address G using the Virtual Tearoom$^{\text{TM}}$ application. Simultaneously, the IXIA interfaces transmitted variable rate streams to the same group address G but with different, multiple source addresses thus simulating a multitude of hosts sending video/audio to one group address. One of the other PCs then joined multicast group G using each of the methods (AFM, SFM and SSM) described above via the Virtual Tearoom$^{\text{TM}}$ application. This scenario was done in both IPv4 and IPv6 and the unsolicited traffic from the IXIA was sent at rates of 10, 20 and 50 Mb/s.

So for example, in the IPv4 case, PC1 with source address 10.9.14.18/28 transmitted to group address 239.255.0.1 while the IXIA interface connected to S1 transmitted to 239.255.0.1 with source addresses 10.9.14.19/28 through to 10.9.14.30/28. The IXIA interface connected to S2 also transmitted to 239.255.0.1 with source addresses 10.9.14.35/28 through to 10.9.14.46/28. PC3 with source address 10.9.14.50/28 then joined (*,239.255.0.1) for AFM and SFM. The group address was then changed to 239.255.128.1 for all the transmitters since 239.255.128.0/17 was configured for SSM in the PIM configuration in routers R1-R3 and PC3 joined group (10.9.14.18,239.255.128.1) for SSM. In the IPv6 case, PC1 with source address fc00:0:0:6::/64 eui-64 (where eui-64 is the auto-configured 64-bit Interface ID derived from the MAC address) transmitted to group address ff05::1 for AFM and SFM and transmitted to group address ff05::1:1 for SSM while the IXIA used source addresses taken from fc00:0:0:6::/64 and fc00:0:0:5::/64. PC3 with source address fc00:0:0:4::/64 eui-64 then did joins to both (*,ff05::1) and (S_1,ff05::1:1) where S_1 denotes the source address of PC1. Note that all unicast and multicast addresses for both IPv4 and IPv6 have site-local scope only and are not globally routable.

IV. Results

In this section we report on the resource utilisation of the various multicast filtering techniques in the experimental set-up described above. Figure 3 shows for IPv4 the bandwidth usage on the multicast recipient's link when 10, 20, and 50 Mbps of unsolicited multicast traffic is injected into the network by the IXIA traffic generator. We observe that AFM and SFM behave identically – both rely on filtering unsolicited

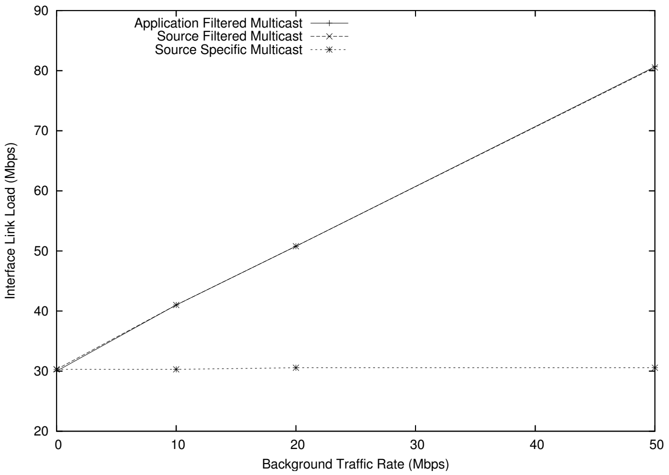

Fig. 3. IPv4 Network utilization

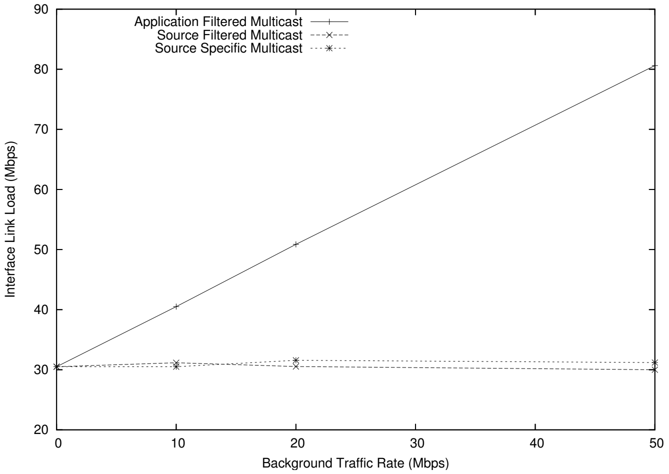

Fig. 4. IPv6 Network utilization

multicast traffic at the receiving host, and their bandwidth requirement increases linearly with the volume of unsolicited multicast traffic. SSM on the other hand is able to block unsolicited traffic within the network, thus protecting end-host resources from unwanted load.

Figure 4 shows the bandwidth utilisation in the case of IPv6. As expected AFM shows linearly increasing bandwidth requirement as the unsolicited traffic load increases, while SSM is as in the case of IPv4 invariant to such load since the network routers block all multicast traffic from unsolicited sources. A surprising observation however is that IPv6 SFM offers a saving in bandwidth much like SSM. Investigation showed that in Microsoft Windows Vista when an application instructs the kernel to insert multicast source filters as per [17], it actually performs a Source Specific Multicast join (S,G) rather than an Any Source Multicast (*,G) join, and thus bandwidth performance for Source Filtered Multicast and Source Specific Multicast was identical.

For both IPv4 and IPv6, we also measured the processor load on the receiving host (Dell Precision 390 with Intel Core 2 CPU 6400 @ 2.13 GHz, with 2GB of RAM) for each of the three multicast filtering schemes, and found that the CPU utilisation under SSM was fairly uniform at around 10%, while it increases linearly with multicast traffic rate in AFM. For IPv4, the CPU utilisation under SFM increased linearly with multicast traffic rate, albeit at a lower rate than AFM, while for IPv6 SFM behaved identically to SSM.

V. Conclusion

With the potential imminence of IPv6 in the public Internet, it is important to make timely preparations by migrating applications to IPv6, and in the process to understand and use the improved capabilities that IPv6 provides. In this paper we have demonstrated a successful migration of our high-performance multimedia platform to IPv6, and have found the process to be relatively smooth on Microsoft Windows Vista. As part of the migration we have profiled the performance of several multicast filtering techniques in a realistic network setting. We have found Source Specific Multicast (SSM) to be a promising way forward as it is supported well in emerging IPv6 platforms, and helps protect network and end-host resources in a scaleable way by filtering unsolicited traffic at the ingress points to the network. As part of our future work we will investage the use of IPv6 SSM in conjunction with stateful multicast firewalling techniques such as the one we developed in [18].

References

[1] C. Huitema, *"IPv6: The New Internet Protocol"*, 2nd Ed. Prentice Hall PTR, 1997.

[2] S. Deering and R. Hinden, *"RFC1883: Internet Protocol, Version 6 (IPv6) Specification"*, Dec 1995, http://www.ietf.org/rfc/rfc1883.txt.

[3] S. Deering and R. Hinden, *"RFC2460: Internet Protocol, Version 6 (IPv6) Specification"*, Dec 1998, http://www.ietf.org/rfc/rfc2460.txt.

[4] A. Miller, *"IPv6: Major Vendors Have Come Aboard"*, May 2007, http://www.enterprisenetworkingplanet.com/netsp/article.php/3676211.

[5] Hain T., *"A Pragmatic Report on IPv4 Address Space Consumption"*, http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html

[6] Geoff Huston, *"IPv4 - How long have we got?"*, Jul 2003, http://www.potaroo.net/ispcol/2003-08/ale.html.

[7] Geoff Huston, *"The End of the (IPv4) World is Nigher!"*, Jul 2007, http://www.potaroo.net/ispcol/2007-07/v4end.html.

[8] *"IPv4 Address Report"*, http://www.potaroo.net/tools/ipv4/index.html.

[9] S. King, R. Fax, D. Haskin, W. Ling, T. Meehan, R. Fink and C. E. Perkins, *"The Case for IPv6"*, Jul 1998, http://tools.ietf.org/id/draft-ietf-iab-case-for-ipv6-02.txt.

[10] *"Adopted Board Resolutions - San Juan, Puerto Rico"*, 29 Jun 2007, http://www.icann.org/minutes/resolutions-29jun07.htm.

[11] Ann Steward, Australian Govt. Chief Information Officer, *"Working Towards 2010 – establishing the way forward"*, Jul 2007, http://www.agimo.gov.au/media/speeches/2007/working_towards_2010_-_establishing_the_way_forward.

[12] S. Deering, W. Fenner and B. Haberman, *"RFC2710: Multicast Listener Discovery (MLD) for IPv6"*, http://www.ietf.org/rfc/rfc2710.txt, October 1999.

[13] R. Vida and L. Costa, *"RFC3810: Multicast Listener Discovery Version 2 (MLDv2) for IPv6"*, http://www.ietf.org/rfc/rfc3810.txt, June 2004.

[14] S. Bhattacharyya, *"RFC3569: An Overview of Source-Specific Multicast (SSM)"*, http://www.ietf.org/rfc/rfc3569.txt

[15] R. Colton, D. Furguson and J. Moy, *"RFC2740: OSPF for IPv6"*, http://www.ietf.org/rfc/rfc2740.txt

[16] R. Gilligan, S. Thomson, J. Bound, J. McCann and W. Stevens, *"Basic Socket Interface Extensions for IPv6"*, http://www.ietf.org/rfc/rfc3493.txt, February 2003.

[17] R. Thaler, B. Fenner and B. Quinn, *"RFC3678: Socket Interface Extensions for Multicast Source Filters"*, http://www.ietf.org/rfc/rfc3678.txt

[18] S. Li, V. Sivaraman, A. Krumm-Heller and C. Russell, *"A Dynamic Stateful Multicast Firewall"*, Proc. IEEE ICC, Glasgow, Scotland, Jun 2007.