AFRL-RI-RS-TR-2010-153

A FRESH LOOK AT INTERNET PROTOCOL VERSION 6 (IPv6) FOR DEPARTMENT OF DEFENSE (DoD) NETWORKS

University of Pennsylvania

August 2010

Sponsored By
Defense Advanced Research Projects Agency
Darpa Order No. BA77/00

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■**UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2010-153 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/
MARK K. WILLIAMS
Work Unit Manager

/s/
WARREN H. DEBANY, Jr.
Technical Advisor, Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* AUGUST 2010	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* June 2009 – March 2010

4. TITLE AND SUBTITLE A FRESH LOOK AT INTERNET PROTOCOL VERSION 6 (IPv6) FOR DEPARTMENT OF DEFENSE (DoD) NETWORKS	**5a. CONTRACT NUMBER** N/A
	5b. GRANT NUMBER FA8750-09-1-0205
	5c. PROGRAM ELEMENT NUMBER 62303E
6. AUTHOR(S) Jonathan M. Smith, W. David Sincoskie, and William A. Arbaugh	**5d. PROJECT NUMBER** IPV6
	5e. TASK NUMBER 00
	5f. WORK UNIT NUMBER 01
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Trustees of the University of Pennsylvania 3451 Walnut Street, Franklin Bldg., Suite P221 Philadelphia, PA 19104-6205	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency 3701 North Fairfax Drive Arlington, VA 22203-1714 AFRL/RIGB 525 Brooks Road Rome, NY 13441-4505	**10. SPONSOR/MONITOR'S ACRONYM(S)** N/A **11. SPONSORING/MONITORING AGENCY REPORT NUMBER** AFRL-RI-RS-TR-2010-153

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from Public Affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project examined the question of U.S. Department of Defense (DoD) adoption of the Internet Protocol, Version 6 (IPv6) and recommends actions by the Defense Advanced Projects Research Agency (DARPA) towards ensuring the integrity of DoD networks during and after the transition. We pay attention to infrastructure readiness, commercial carriers, security implications and tradeoffs and discuss timing implications and where DoD might rationally position itself on an S-adoption curve.

15. SUBJECT TERMS

Internet Protocol, Computer Networking, Network Security, Computer Security, Technology Deployment

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Mark K. Williams
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (*Include area code)***
U	U	U	UU	33	N/A

Table of Contents

List of Figures

List of Tables

1.0 Introduction

The Internet Protocol (IP) is genuinely ubiquitous, carrying chats, documents, imagery, voice and video across networks ranging from transoceanic fiber optic links to wireless tactical mobile ad-hoc networks (MANETs). The basic idea of the Internet Protocol is a uniform interoperability layer for diverse network technologies, as shown in Figure 1. If a network technology can encapsulate and transport IP packets it can be grafted onto the larger Internet and hosts connected to the network can be participants in the larger Internet. Upper layer protocols and applications can use any sub-network over which IP runs.

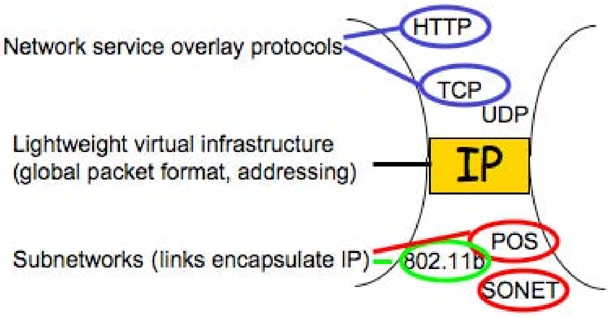
Network service overlay protocols
HTTP
TCP
UDP
Lightweight virtual infrastructure
(global packet format, addressing)
IP
Subnetworks (links encapsulate IP)
POS
802.11b
SONET

Figure 1: Internet Architecture "hourglass"

IP network participants must adopt a common packet format to allow routing amongst IP nodes, in particular devices called routers or gateways. Routers make decisions about best or preferred states using information available from neighbors in the IP connectivity graph, and forward packets from their source host to a destination host using information in the packets. The source and destination addresses and associated information are contained in a data structure called a header (since it is at the front of the packet), followed by a body that contains the data portion of the packet.

The dominant packet format in use today is Version 4 of the Internet Protocol, referred to as IPv4. IPv4 is characterized by 32-bit source and destination addresses. Various schemes are used to divide up this address space, including a hierarchy of classes with different allocations of the address bits to network and host, multicast and broadcast addresses, non-routable addresses and sub-netting support to further subdivide an address according to local needs. This 32-bit address space was foreseen to be inadequate in the early 1990s and a process began to standardize a new version of IP with 128-bit addresses to avoid version changes forced by address space run-out in the future. The designers addressed a number of other perceived shortcomings of IPv4 with new features such as integrating cryptographic security mechanisms and auto-configuration capabilities.

Adoption of IPv6 has not been as rapid as expected by its designers for three primary reasons. First, devices known as Network Address Translators (NATs) became widely available and deployed, arguably as a consumer reaction to ISPs charging per-address rates. This had the effect of reducing pressure on the IP address space, as many hosts connected to the NAT can share a single routable IP address. Second, many of the perceived shortcomings of IPv4 were addressed with additional functionality that was retrofitted, e.g., the Dynamic Host Configuration Protocol (DHCP), which carries out many of the necessary tasks for auto-configuration, including allocating available IP addresses to hosts and identifying domain service servers for hosts. DHCP, "leasing" addresses, also slightly reduces address space pressure, relative to static address allocations and their lesser potential for reuse. Third, a good business case has not yet developed; the transition is perceived as cost with little to no benefit, other than addresses, and a "flag day" transition is potentially risky and disruptive to Internet-based enterprises.

This study was undertaken to provide findings and recommendations on an appropriate stance towards IPv6 adoption by the U.S. Department of Defense. Here, when we use the word adoption, we mean that the protocol is the dominant protocol in actual use, carrying chats, charts, imagery and video. This is different than deployed, which can mean that the capability is present, but unused. A primary concern of DoD is the security of networks, as the move towards network-centricity has made networking an important part of U.S. Defense strategy and therefore an inviting target for U.S. military adversaries. Security analysis of IPv6 must include analyses of host software and training to understand points at which challenges might arise.

In carrying out this study, we drew on our own expertise, performed experiments, and consulted technical experts at router vendors, security appliance vendors, Internet service providers and web companies. There were several meetings with DoD elements that have bearing on the findings and recommendations.

The remaining report is organized as follows. Commercial adoption, which drives availability of cost-effective technology in the marketplace, is discussed in Section 2. A variety of DoD-specific challenges, such as information security, cyberwarfare and training are addressed in Section 3. Timing issues for DoD are important, and the adoption timing is a subtle choice, involving many elements of a complex information "ecosystem" – we address these issues in Section 4. Section 5 summarizes our findings and makes three recommendations, two short-term and one longer term. Section 6 concludes the report and Section 7 provides an annotated list of sources consulted in the study. Appendix A provides brief biographies of the authors, and Appendix B provides documentation of some configuration and software support for IPv6 in a consumer operating system (Apple Mac OS X, 10.6 – "Snow Leopard"). The key observation is that the management and configuration software is immature for this client platform.

2.0 Methods, Assumptions, and Procedures

The commercial Internet is organized as a federation of Internet Service Providers (ISPs), of various sizes and business models. All ISPs possess groups of routers and links of various capacities and geographic spans. Business models may include home access, business services, or long haul carrier businesses (e.g., transcontinental or international IP traffic carriage). To maximize use of real estate and infrastructure such as fibers, carriers route long-haul traffic using the highest capacity links (10-40 Gigabits per second), and switch as many of these at a single physical location as possible. Large carrier routing platforms have aggregate capacities of multiple terabits per second and are currently capable and equipped for operating IPv6 at more or less full performance. Multiple commercial carriers have been operating IPv6 internally for over two years. Broadband service providers such as Comcast have announced availability of IPv6 for their wholesale customers (see http://www.internetnews.com/infra/article.phpr/3825696/Comcast+Embraces+IPv6.htm), but are running dual-stack. Some commercial network providers, then, are ready to offer IPv6.

Turning our attention now to hosts, the key issue for a host that is attached to a network that can run IP is whether it has a software system that can interact with the version (or versions) of the IP protocol that other systems are using. For example, if all other systems attached to the network are running IPv4 software, and the routers support IPv4, and the host is running IPv6, then it has no systems with which it can exchange packets. If the network supports IPv6 but there are no remote hosts with software supporting IPv6, then there’s effectively a route to nowhere. If there are remote hosts that can operate IPv6, they can use the IPv4 network to carry traffic through a so-called tunnel. Therefore it is crucially important that host software support IPv6 if a computer network is to form. This software support is found in the operating system software of a computer host, such as Microsoft’s Windows XP, Windows Vista and Windows 7, Apple’s Mac OS, and open source operating systems such as Linux and OpenBSD.

2.1 DoD-Specific Issues

There are a variety of issues with IPv6 deployment peculiar to the U.S. Department of Defense, and the protocol path versus DoD needs.

2.1.1 Compatibility with Legacy Equipment

A first concern is compatibility with legacy equipment (a direct consequence of long acquisition lead times) as well as long shelf life. A subtly related issue is training and manpower – technical training takes time to develop and mature, and must encompass both legacy networks as well as newly deployed or deploying technologies. For example, familiarity with IPv4 and IPv6, as well as the configuration and management of dual-stack implementations must be addressed in technical training. A significant challenge to DoD may arise as a consequence of lack of trained people to configure and manage new deployed network technologies.

2.1.2 Compatibility with DoD Network

A second, related issue is compatibility with the DoD Network Centric Warfare ("netcentricity") visions. Examples include the overarching Global Information Grid and the service-specific netcentricity projects such as the Navy's FORCENet. A key characteristic of these architectures is that they are service oriented architectures (SOAs), which are characterized by their reliance on composition of distributed "services" such as those for information naming, retrieval and dissemination. SOAs, as overlays running over existing networks, should be able to run as well on IPv6 as IPv4; in principle. The implication is that any protocol transition should be transparent to applications that employ such as an architecture.

2.1.3 Costs and Overheads

A third important issue is money. For example, while IPv6 capabilities are mandated, there are many costs and overheads and it is not clear that these have been adequately budgeted for and funded. Costs include the training and staffing discussed earlier.

2.1.4 Types of Network

A fourth important issue is the types of networks in common use in the DoD. These include satellite communications networks (which are difficult to update once deployed and have long lead times) as well as tactical networks. Wireless links are a key part of many of these networks as many parts of the DoD are constantly "on the move" (consider, for example, Air Force networks) and only wireless communication supports these operations. For slow speed wireless links the overheads of IPv6 may prove significant.

2.1.5 Security

A final issue, and an issue of primary importance, is information assurance and the issue of network security more generally. Computer networking has become an important part of modern warfare, ranging from situational awareness in the battlespace to logistics. Disruptions to these networks impair warfighting capabilities, and if communications security is not maintained, can lead to information leakage and its negative consequences. As some actors have already signaled their intentions to employ cyberwarfare both alone and in concert with kinetic warfare, it is clear that U.S. networks must be secure, and offensive capabilities may be required as well. As many information assurance matters involve the National Security Agency (NSA), IPv6 plans must take into account NSA Certification and availability of approved devices such as the High Assurance IP Encryptor (HAIPE) for IPv6.

2.2 Timing

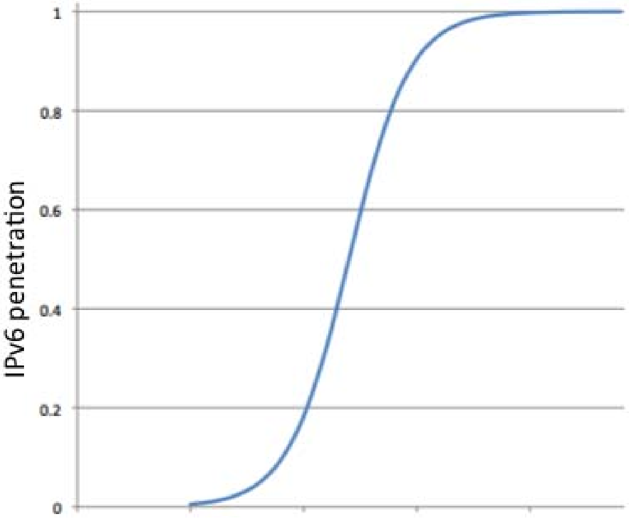

Figure 2: Deployment "S-curve"

Figure 2 illustrates an "S-curve", used to represent the fraction of users who have transitioned from IPv4 to IPv6. The key attribute of the S-curve is the time at which the sudden increase in adoption occurs. Before this point, adoption is slow, and after it, the remaining IPv4 users are a small and decreasing percentage of users. It is important to understand what IPv6 transition means: it is the point where the IPv6 protocol is the primary means used for computer networking. If mismeasured, counts might use metrics such as: (1) routers with IPv6 capability, (2) hosts with IPv6 stacks, or even (3) host operating systems designed to prefer IPv6 (if present). These would result in a gross overestimate of operational IPv6, as the IPv6 ecosystem (web services, IPv6 to the home, IPv6 wireless hotspots) has not been fully populated, and all of these are necessary for the majority of Internet users. Based on data from Google in their IPv6 deployment [11, 15], in early 2010 we remain on the flat, pre-increase portion of the curve. Based on Google's data, France is the furthest along in transition to IPv6, driven largely by a single early ISP adopter for the technology (Free).

Many elements are aligned for a sudden increase in IPv6 connectivity:

1. Windows 7 and Mac OS X turn on IPv6 by default, and prefer it at boot time if available (most users are unaware that it is turned on and might be surprised to find that it is active at turn-on; this may permit local networking that is unobserved).

2. Most universities have localized “islands” of IPv6, and there is interconnection of these islands via networks such as Internet2. Universities tend to be early adopters, with an important side effect: they set student expectations. As these students leave the universities and take positions in government and the commercial world they may expect IPv6 connectivity.

3. Major broadband providers such as Comcast have been operating IPv6 internally for several years, and have recently announced IPv6 availability for consumers, presumably to gain an advantage in the competitive broadband marketplace (cable versus DSL/FiOS).

4. Specialized ISPs such as Hurricane Electric have arisen to offer IPv6 services.

5. Some Web service providers, notably Google, have been strongly advocating transition to IPv6, and have put “skin in the game” by deploying IPv6 connectivity to their various web services. Unlike many other actors in the ecosystem, Google has financial incentives to transition to IPv6, since network address translators may make services more difficult to deploy and may interfere with the precision of targeting personalization and advertising.

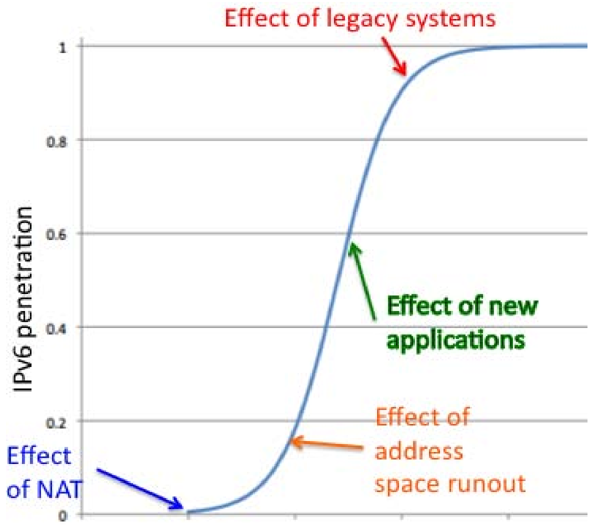

Figure 3: Key pressures leading to onset of rapid deployment

There are also some logistical barriers, in particular the large deployed infrastructure of IPv4 (e.g., for wireless access in hotels) and Network Address Translators, which are often combined with packet-filtering firewall and router functionality in home deployments. Figure 3 indicates the effects of some of these factors.

For DoD, the timing issues are complex. On the one hand, there is the natural desire to maintain technology leadership. On the other hand, as discussed earlier, security considerations are of greater concern in DoD than in the commercial world, not least due to the existence of well-funded nation-state adversaries as well as criminal elements that may or may not serve as proxies for nation-states.

This issue is important because DoD must gain maturity with security issues at least as fast as adversaries, in both the defensive and offensive domains, or be put at a disadvantage during and after the transition. In the next Section, 3.0 Results and Discussions and 4.0 Recommendations, we propose a strategy for preparing for transition.

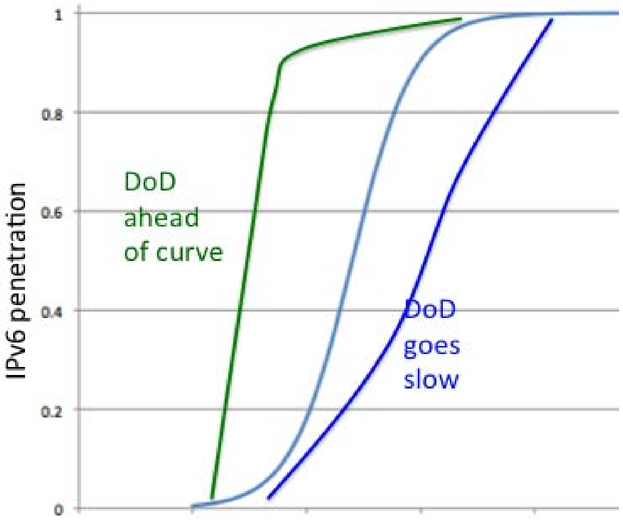

Figure 4: DoD Timing Strategies

There is risk, however, in wholesale transition before such preparation is complete. IPv6 addresses require new code in programs where connections are established and packet addresses are examined, because the data objects are of different size (16 bytes versus 4 bytes). Code can be modified to be IPv6-only or to support dual-stack (IPv4 and IPv6) but either dual-stack or IPv6-only code represent immature code paths, with potential opportunities for malice. These immature code paths affect any code that requires changes for IPv6, and will affect both application code and tools. This latter issue, that of tools, is of particular concern since system administration practices (such as the use of security appliances) depend heavily on tools for network management, diagnosis and protection. Administrators will require training and familiarity gained through experience to be effective with new tools, e.g., for configuring IPv6 addresses, specifying IPv6 filtering rules, and configuring temporary measures such as “6to4” IPv6 in IPv4 encapsulation (used to tunnel IPv6 over IPv4 – see RFC3056).

3.0 Results and Discussion

3.1 Finding 1

Router performance is not an issue. Industry is managing to keep performance high in spite of larger addresses and larger routing tables. Hardware capabilities such as Application-Specific Integrated Circuits (ASICs) coupled with specialized memory technologies appear to be adequate for core routers. The impacts of extensible headers are unclear but will only be an issue if such headers are widely used.

3.2 Finding 2

IPv6 does not address all IPv4 security issues and further, may introduce new IPv6 issues. Security issues can be formulated in a framework such as “CIA”, for Confidentiality, Integrity and Availability. Many of these issues are the same for both protocols, e.g., application layer vulnerabilities dominate those at the network layer, and reconnaissance will continue although methods will change. As noted by Vyncke[14] public servers will still need to be DNS reachable and to overcome the difficulty of remembering the long IPv6 addresses, administrators may use easy-to-remember addresses – Vyncke gives `::F00D` and `::C5C0` (he works for Cisco) as examples.

While IPv6 integrates the security features of IP Security (IPsec) into the standard protocol, neither is the use of these features mandated nor does it ease the use and configuration of the cryptographic protections IPv6 security offers. For example it does not overcome the deployment and operational challenges of public-key infrastructures. It is well known that cryptography is not security – it is instead a well-founded building block for protocols to protect confidentiality and check integrity. There are some other issues with securing all flows with IPsec, also noted by Vyncke – notably those endpoints and end-users must be trusted because firewalls and ACls are blinded, as is Netflow-based network telemetry.

As with IPv4, IPv6 security cannot protect against availability threats such as denial of service attacks (see for example the tools `6tunneldos`, `4to6ddos` and `ipv6f*ck`), and can make no guarantees about its own implementation (e.g., that an assumption of randomness is actually met). This latter point, implementation, is a particular challenge as IPv6 is a software system and therefore subject to all of the “bugs” characteristic of software systems, including some exploitable for attacks. As a somewhat interesting example, the security-focused OpenBSD open source operating system was penetrated by an IPv6 implementation bug in 2007.

3.3 Security Issues

Other important security issues are raised by dual-stack implementations and tunnels. Dual-stack implementations have the (unfortunate) property that they create an attack surface for applications that consists of both IPv6 and IPv4. The fact that IPv6 is enabled by default creates opportunities for an attacker sending Router Advertisements to configure your host to IPv6 and the attack surface is opened. In Appendix B, we have included documentation from the MacOS user manual that illustrates the state of IPv6 implementations (`inet6`), as well as two other tools, `gif` (used for tunneling) and `stf` (an interface to a more specific IPv6 over IP4 tunneling capability). While these tunneling solutions are intended as temporary measures, the deployment “S Curve” of Figure 2 suggests that these transition mechanisms will persist for a long time.

Configuration and management will introduce new risks, for example in IPv6 address allocation configurations using MAC addresses that allow attackers to deduce machine types and some network configuration information. There is no security mechanisms built into the discovery protocol.

3.4 Finding 3

Existing DARPA net-centric research programs are generally architecturally compatible with IPv6, although additional software may be needed. A survey of DARPA network-centric programs was performed for this study and is summarized in Table 1.

Table 1: DARPA Net-centric Programs

Program	DARPA Office	Compatible IPv4?	Compatible IPv6?
SAPIENT	IPTO	Yes	Yes[#]
Maingate	STO	Yes	Yes
Control Plane	STO	Yes	Yes[#]
IAMANET	STO	No*	No*
CORONET	STO	Yes	Yes
MNP	STO	Yes	Yes[#]
DTN	STO	Yes	Yes
Connectionless	STO	Yes	Yes[#]
WNaN	STO	Yes	Yes
LANdroids	IPTO	Yes	Yes

**# Architecturally compatible;
may require additional software**

*** Reachback via gateway**

DARPA programs generally address fundamental problems, and the impact of transitioning capabilities into military IPv6 networks will primarily be software modifications (if any are needed). For example, the DARPA/IPTO *SAPIENT* program might require an additional protocol module for IPv6, as would the DARPA/STO *Disruption Tolerant Networking* program. Other programs such as DARPA/STO *Wireless Network after Next (WNaN),* DARPA/IPTO *LANdroids* and DARPA/STO *CORONET* should be unaffected. The additional overhead of larger addresses and headers for IPv6 may affect performance in wireless networks such as Mobile Ad-hoc Networks (MANETs).

3.5 Finding 4

IPv6 penetration today is very low, with a slow rate of adoption. Data from Google [15] show that from September 2008 to September 2009, the year over year growth (Fig. 2 of [15]) was 35%, but based on Google’s metrics about 0.25% of users had working IPv6. Based on when connections were made to Google IPv6 web services, they also conclude (based on a higher number on weekends) that there is more available from home than in the workplace. The predominant connectivity type is `6to4`, followed by “`native/tunnel/unknown`”, with a tiny fraction of connectivity due to Teredo and ISATAP. The predominant operating systems are MacOS and Windows Vista. If we extrapolate their 35% growth rate to plot an exponential curve (rather than the “S curve” we believe will characterize deployment) it will take until 2028 for IPv6 to become the dominant protocol.

Some commercial firms, notably Google, have incentives to transition, such as improving the user experience (e.g., by interacting directly with a user machine rather than through a NAT). Major Internet Service Providers such as Comcast have announced an IPv6 deployment plan, and the latest versions of consumer operating system products (e.g., Microsoft Windows 7, Apple Mac OS 10.6 “Snow Leopard” and various instances of the open source operating systems such as Ubuntu Linux) incorporate IPv6 and, in fact, prefer routing by IPv6 if it is present. Other operating systems (Windows XP, NetBSD, OpenBSD, some Linuxes, FreeBSD) support IPv6, but not as the default option.

Even presuming an “S-curve” upsweep in IPv6 penetration occurs (e.g., in 2011 or 2012) there will be a huge installed base of IPv4-only equipment, including home routers, devices in small-office/home-office (SOHO) settings, and relatively recent deployments in settings accessed by consumers such as hotel rooms and coffee shops where the existing equipment works well enough – and produces enough revenue – that additional capital expenses would be hard to justify. For example, the Verizon FiOS wireless broadband routers do not support native IPv6 – tunneling over IPv4 is required. Also, no compelling IPv6-only applications have yet emerged, with the possible exception of Microsoft’s “MeetingSpace”.

3.6 Finding 5

IPv6 deployment appears to be proceeding more rapidly outside of the United States. Based on data from JTF-GNO, Table 2 shows the rank and percentage of currently assigned IPv6 address blocks for the top 10 allocations (about 96% of the total):

Table 2: Assigned IPv6 addresses

Country	%
Brazil	47.25
US	10.77
Germany	7.03
Japan	6.00
France	5.99
Australia	5.94
European Union	4.43
South Korea	3.74
Italy	3.00
Taiwan	1.66

Data from Google (interpreting Figures 6 and 7 from [15]) indicate that the top 10 countries by ratio of working IPv6 are Russia, France, Ukraine, China, US, Poland, Sweden, Canada, Netherlands and Japan. As a better indicator of working infrastructure, if the non-relayed (i.e., no `6to4` or Teredo, which could be deployed by users with no local network infrastructure, leaving "`native/tunnel/unknown`" and ISATAP) working IPv6s are extracted, the top 10 are France, China, Sweden, Netherlands, US, Japan, Poland, Russia, Canada and Ukraine. While most autonomous systems (ASes) with large IPv6 connectivity are universities or research institutions (3 in China, 2 in the US, according to Tables 1 and 2 of [15]) the Free AS (AS12322) in France generates a large percentage of French IPv6 connectivity as measured by Google's methodology.

3.7 Finding 6:

Neither IPv6 performance nor the interaction of IPv6 features with wireless network architectures are well understood in mobile, wireless networks. This topic is particularly important in tactical networks, which are almost exclusively mobile and wireless. For example, many proposed future military networks such as that of the Army's Future Combat System (FCS), are mobile ad hoc networks (MANETs) and yet there is limited practical experience with MANETs and their performance. Further, the interaction between IPv6 and mobility (e.g., "care of" addresses, etc.) is still undergoing study in the standardization process, with multiple Internet Engineering Task Force (IETF) study groups, e.g., 16ng for IEEE 802.16, trying to resolve technical issues. Some of these difficulties appear due to the IPv6 standard being developed prior to the increasing presence of mobile and wireless devices such as netbooks and smartphones, but these consumer devices might be considered representative of challenges to be faced in military tactical networks.

4.0 Recommendations

4.1 Recommendation 1

DARPA should consider initiating a program or series of programs focused on IPv6 security appliances, such as firewall/gateways and intrusion detection systems. A well-documented, open source, reference IPv6 or IPv6/IPv4 (dual-stack) firewall implementation engineered to the highest software engineering standards and red-teamed by multiple capable red teams to produce a definitive (and transitional) implementation (e.g., one which defines limits on continuation headers) will stimulate new products in the commercial world, either using the DARPA code base or augmenting it. It would find immediate application in NIPR/public Internet gateways. Advances possible in such a program would include highly usable policy expression languages and tools to translate such languages into low-level filtering and analysis specifications, coordination systems to ensure that a set of firewalls are enforcing the same policy, and automatic filter adaptation ("intelligent firewall") based on machine learning and feedback.

4.2 Recommendation 2

DARPA should consider initiating a breakaway effort in creating novel, highly usable and well-documented software tools for IPv6 diagnosis, configuration and management.

This programmatic thrust would include tools to:

- Automate IPv6 conversions to insure they end up with a safe default configuration;
- Use cognitive techniques to diagnose security problems and recommend fixes; and
- Automate new IPv6 configuration setups to avoid security flaws such as 48-bit host addresses based on Ethernet MACs (by, for example, initially assigning truly random host numbers).

Better tools would have the additional side effect of easing training and thus the challenges associated with a shortage of IPv6-trained personnel.

4.3 Recommendation 3

DARPA should consider initiating a program or series of programs focused on IPv6 mobility/IPv6 wireless. Particular issues to be addressed include exploratory performance studies, autoconfiguration overheads and autoconfiguration requirements. For example, IEEE 802.16 (WiMAX) Wireless MANs have problems with IPv6 autoconfiguration due to the 802.16 Medium Access Control definitions in particular its non-support for native multicast. Research is necessary on networks intended for military challenges not generally encountered in civilian settings, such as satellite networks and mobile ad-hoc networks.

5.0 Conclusions

DARPA should consider revolutionary programs to overcome difficulties with IPv6 through increased automation, leveraging DARPA advances in cognitive systems.

6.0 References and Sources

[1] *IPv6* (2nd Edition), Christian Huitema, 1997.

[2] *"A Filtering Strategy for Mobile IPv6"*, NSA IAD Report #I733-040R-2007

[3] *"Firewall Design Considerations for IPv6"*, NSA IAD Report #I733-041R-2007

[4] "*FORCEnet Implementation Strategy*" (2005), Naval Studies Board, National Academies Press.

[5] "*Information Assurance for Network-Centric Naval Forces*" (2009), National Academies Press.

[6] MacOS X 10.6 manual: inet6, ip6, stf

[7] Discussions with Cisco CRS-1 architects, Summer 2009.

[8] Dr. Vinton G. Cerf, Google – 9/25/09, Google Washington, DC.

[9] JTF-GNO briefing, Mr. Eric B. Gonzalez, Arlington, VA, 10/20/09.

[10] "carriers" (will need to get permission to cite I think)

[11] Dr. Lorenzo Colitti, Google – 12/11/09, Mountain View, CA

[12] IETF RFC 5154, "IP over IEEE 802.16 Problem Statement and Goals", April 2008.

[13] Internet Draft, "Routing Loops Using ISATAP and 6to4: Problem Statement and Proposed Solutions", http://www.ietf.org/id/draft-nakibly-v6ops-tunnel-loops-01.txt

[14] http://www.cisco.com/web/SG/learning/ipv6_seminar/files/02Eric_Vyncke_Security_Best_Practices.pdf

[15] "Evaluating IPv6 Adoption in the Internet", Lorenzo Colitti, Steinar H. Gunderson, Erik Kline, Tiziana Refice, preprint, December 2009.

List of Acronyms

DoD	Department of Defense
IPv4	Internet Protocol, Version 4
IPv6	Internet Protocol, Version 6
DARPA	Defense Advanced Projects Research Agency
IP	Internet Protocol
MANETs	Mobile Ad-hoc Network
NATs	Network Address Translators
DHCP	Dynamic Host Configuration Protocol
ISPs	Internet Service Providers
SOAs	Service Oriented Architectures
NSA	National Security Agency
HAIPE	High Assurance Internet Protocol Encryptor
ASICs	Application-Specific Integrated Circuits
“CIA”	Confidentiality, Integrity, and Availability
IPsec	Internet Protocol Security
WNaN	Wireless Network after Next
SOHO	Small-office/home-office
ASes	Autonomous Systems
FCS	Future Combat System
IETF	Internet Engineering Task Force
ISAT	Information Science and Technology
BAST	Board on Army Science and Technology

Appendix A

Biographies:

Jonathan M. Smith is the Olga and Alberico Pompa Professor of Engineering and Applied Science at the University of Pennsylvania to which he recently returned after almost three years at DARPA. He was awarded the OSD Medal for Exceptional Public Service in 2006 for his DARPA service. He is an IEEE Fellow. He was previously at Bell Telephone Laboratories and Bellcore which he joined at the AT&T divestiture. His current research interests range from programmable network infrastructures and cognitive radios to architectures for computer augmented immune response. Dr. Smith serves on the DARPA Information Science and Technology (ISAT) study group and the National Research Council's Board on Army Science and Technology (BAST).

William A. Arbaugh is an Associate Professor of Computer Science at the University of Maryland, College Park. He joined the Computer Science department at Maryland after spending sixteen years with the U.S. Department of Defense-first as a commissioned officer in the Army and then as a civilian. During the sixteen years, Prof. Arbaugh served in several leadership positions in diverse areas ranging from tactical communications to advanced research in information security and networking. Professor Arbaugh was also awarded one of the highest awards offered by DOD for technical achievement- the Louis W. Tordella award for technical leadership. He also briefed several members of congress and cabinet members on emerging security issues. In his last position at DOD, Prof. Arbaugh served as a senior technical advisor in an office of several hundred computer scientists, engineers, and mathematicians conducting advanced networking research and engineering.

While at Maryland, Professor Arbaugh founded and led as President and CTO - Komoku, Inc. Komoku focused on detecting host based zero day malware. Komoku's technology detected multiple instances of nation-state caliber malware that no other product (government or commercial) had previously been able to detect. Microsoft acquired Komoku in 2007, and the technology has been incorporated into their product line.

Prof. Arbaugh received a B.S. from the United States Military Academy at West Point, an M.S. in Computer Science from Columbia University in New York City, and a PhD in computer science from the University of Pennsylvania in Philadelphia. Prof. Arbaugh's research interests include information systems security and privacy with a focus on wireless networking, embedded systems, and configuration management. He also has served on the editorial boards of IEEE Computer where he edited a bi-monthly column on Information Security, and IEEE Security and Privacy.

W. David Sincoskie, a member of the National Academy of Engineering, is Director of the Center for Information and Communications Sciences and Professor of Electrical and Computer Engineering at the University of Delaware. From 1996 – 2008, he was Group Senior Vice President of the Network Systems Research Laboratory at Telcordia. The laboratory consists of over 100 researchers involved in many aspects of Internet and broadband networking. Major areas of activity in the lab are Internet network management, mobile and ad-hoc Internet, wireless communications, and optical network management. Prior service includes the Defense Advanced Research Projects Agency’s Information Science and Technology committee and the Internet Architecture Board.

Dr. Sincoskie was Executive Director of the Computer Networking Research Department at Telcordia (formerly Bellcore) from 1990 through 1996. He managed a group working on the AURORA gigabit testbed, IPv6, IP over ATM, NSFNET, and broadband service control. Dr. Sincoskie is a Fellow of the IEEE. He was inducted to the University of Delaware’s Alumni Wall of Fame in 2006, and received the Distinguished Electrical Engineering Alumnus award in 1994. He received the Bellcore President’s award in 1993. In 2003, he received the IEEE Fred W. Ellersick prize paper award. He was an Adjunct Professor of Computer and Information Science at the University of Pennsylvania from 1989-2008. Dr. Sincoskie served on the National Academies Board on Army Science and Technology from 2003-2008 and was a member of the BAST Committee on Strategies for Network Science, Technology, and Experimentation.

Appendix B
IPv6 Software Documentation

NAME

inet6 — Internet protocol version 6 family

SYNOPSIS

```
#include <sys/types.h>
#include <netinet/in.h>
```

DESCRIPTION

The **inet6** family is an updated version of inet(4) family. While inet(4) implements Internet Protocol version 4, **inet6** implements Internet Protocol version 6.

inet6 is a collection of protocols layered atop the *Internet Protocol version 6* (IPv6) transport layer, and utilizing the IPv6 address format. The **inet6** family provides protocol support for the SOCK_STREAM, SOCK_DGRAM, and SOCK_RAW socket types; the SOCK_RAW interface provides access to the IPv6 protocol.

ADDRESSING

IPv6 addresses are 16 byte quantities, stored in network standard byteorder. The include file ⟨netinet/in.h⟩ defines this address as a discriminated union.

Sockets bound to the **inet6** family utilize the following addressing structure:

```
struct sockaddr_in6 {
        u_int8_t       sin6_len;
        u_int8_t       sin6_family;
        u_int16_t      sin6_port;
        u_int32_t      sin6_flowinfo;
        struct in6_addr        sin6_addr;
        u_int32_t      sin6_scope_id;
};
```

Sockets may be created with the local address “::” (which is equal to IPv6 address 0:0:0:0:0:0:0:0) to affect “wildcard” matching on incoming messages.

The IPv6 specification defines scoped addresses, like link-local or site-local addresses. A scoped address is ambiguous to the kernel, if it is specified without a scope identifier. To manipulate scoped addresses properly from the userland, programs must use the advanced API defined in RFC2292. A compact description of the advanced API is available in ip6(4). If a scoped address is specified without an explicit scope, the kernel may raise an error. Note that scoped addresses are not for daily use at this moment, both from a specification and an implementation point of view.

The KAME implementation supports an extended numeric IPv6 address notation for link-local addresses, like “fe80::1%de0” to specify “fe80::1 on de0 interface”. This notation is supported by getaddrinfo(3) and getnameinfo(3). Some of normal userland programs, such as telnet(1) or ftp(1), are able to use this notation. With special programs like ping6(8), you can specify the outgoing interface by an extra command line option to disambiguate scoped addresses.

Scoped addresses are handled specially in the kernel. In kernel structures like routing tables or interface structures, a scoped address will have its interface index embedded into the address. Therefore, the address in some kernel structures is not the same as that on the wire. The embedded index will become visible through a PF_ROUTE socket, kernel memory accesses via kvm(3) and on some other occasions. HOWEVER, users should never use the embedded form. For details please consult IMPLEMENTATION supplied with KAME kit.

PROTOCOLS

The **`inet6`** family is comprised of the IPv6 network protocol, Internet Control Message Protocol version 6 (ICMPv6), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). TCP is used to support the `SOCK_STREAM` abstraction while UDP is used to support the `SOCK_DGRAM` abstraction. Note that TCP and UDP are common to `inet`(4) and **`inet6`**. A raw interface to IPv6 is available by creating an Internet socket of type `SOCK_RAW`. The ICMPv6 message protocol is accessible from a raw socket.

MIB Variables

A number of variables are implemented in the net.inet6 branch of the `sysctl`(3) MIB. In addition to the variables supported by the transport protocols (for which the respective manual pages may be consulted), the following general variables are defined:

`IPV6CTL_FORWARDING` (ip6.forwarding) Boolean: enable/disable forwarding of IPv6 packets. Also, identify if the node is acting as a router. Defaults to off.

`IPV6CTL_SENDREDIRECTS` (ip6.redirect) Boolean: enable/disable sending of ICMPv6 redirects in response to unforwardable IPv6 packets. This option is ignored unless the node is routing IPv6 packets, and should normally be enabled on all systems. Defaults to on.

`IPV6CTL_DEFHLIM` (ip6.hlim) Integer: default hop limit value to use for outgoing IPv6 packets. This value applies to all the transport protocols on top of IPv6. There are APIs to override the value.

`IPV6CTL_MAXFRAGPACKETS` (ip6.maxfragpackets) Integer: default maximum number of fragmented packets the node will accept. 0 means that the node will not accept any fragmented packets. -1 means that the node will accept as many fragmented packets as it receives. The flag is provided basically for avoiding possible DoS attacks.

`IPV6CTL_ACCEPT_RTADV` (ip6.accept_rtadv) Boolean: enable/disable receiving of ICMPv6 router advertisement packets, and autoconfiguration of address prefixes and default routers. The node must be a host (not a router) for the option to be meaningful. Defaults to off.

`IPV6CTL_KEEPFAITH` (ip6.keepfaith) Boolean: enable/disable “FAITH” TCP relay IPv6-to-IPv4 translator code in the kernel. Refer `faith`(4) and `faithd`(8) for detail. Defaults to off.

`IPV6CTL_LOG_INTERVAL` (ip6.log_interval) Integer: default interval between IPv6 packet forwarding engine log output (in seconds).

`IPV6CTL_HDRNESTLIMIT` (ip6.hdrnestlimit) Integer: default number of the maximum IPv6 extension headers permitted on incoming IPv6 packets. If set to 0, the node will accept as many extension headers as possible.

`IPV6CTL_DAD_COUNT` (ip6.dad_count) Integer: default number of IPv6 DAD (duplicated address detection) probe packets. The packets will be generated when IPv6 interface addresses are configured.

`IPV6CTL_AUTO_FLOWLABEL` (ip6.auto_flowlabel) Boolean: enable/disable automatic filling of IPv6 flowlabel field, for outstanding connected transport protocol packets. The field might be used by intermediate routers to identify packet flows. Defaults to on.

`IPV6CTL_DEFMCASTHLIM` (ip6.defmcasthlim) Integer: default hop limit value for an IPv6 multicast packet sourced by the node. This value applies to all the transport protocols on top of IPv6. There are APIs to override the value as documented in `ip6`(4).

`IPV6CTL_GIF_HLIM` (ip6.gifhlim) Integer: default maximum hop limit value for an IPv6 packet generated by `gif`(4) tunnel interface.

`IPV6CTL_KAME_VERSION` (ip6.kame_version) String: identifies the version of KAME IPv6 stack implemented in the kernel.

`IPV6CTL_USE_DEPRECATED` (ip6.use_deprecated) Boolean: enable/disable use of deprecated address, specified in RFC2462 5.5.4. Defaults to on.

`IPV6CTL_RR_PRUNE` (ip6.rr_prune) Integer: default interval between IPv6 router renumbering prefix babysitting, in seconds.

`IPV6CTL_MAPPED_ADDR` (ip6.mapped_addr) Boolean: enable/disable use of IPv4 mapped address on `AF_INET6` sockets. Defaults to on.

`IPV6CTL_RTEXPIRE` (ip6.rtexpire) Integer: lifetime in seconds of protocol-cloned IP routes after the last reference drops (default one hour).

`IPV6CTL_RTMINEXPIRE` (ip6.rtminexpire) Integer: minimum value of ip.rtexpire (default ten seconds).

`IPV6CTL_RTMAXCACHE` (ip6.rtmaxcache) Integer: trigger level of cached, unreferenced, protocol-cloned routes which initiates dynamic adaptation (default 128).

Interaction between IPv4/v6 sockets

The behavior of `AF_INET6` TCP/UDP socket is documented in RFC2553. Basically, it says this:

- A specific bind on an `AF_INET6` socket (`bind`(2) with an address specified) should accept IPv6 traffic to that address only.
- If you perform a wildcard bind on an `AF_INET6` socket (`bind`(2) to IPv6 address `::`), and there is no wildcard bind `AF_INET` socket on that TCP/UDP port, IPv6 traffic as well as IPv4 traffic should be routed to that `AF_INET6` socket. IPv4 traffic should be seen as if it came from an IPv6 address like `::ffff:10.1.1.1`. This is called an IPv4 mapped address.
- If there are both a wildcard bind `AF_INET` socket and a wildcard bind `AF_INET6` socket on one TCP/UDP port, they should behave separately. IPv4 traffic should be routed to the `AF_INET` socket and IPv6 should be routed to the `AF_INET6` socket.

However, RFC2553 does not define the ordering constraint between calls to `bind`(2), nor how IPv4 TCP/UDP port numbers and IPv6 TCP/UDP port numbers relate to each other (should they be integrated or separated). Implemented behavior is very different from kernel to kernel. Therefore, it is unwise to rely too much upon the behavior of `AF_INET6` wildcard bind sockets. It is recommended to listen to two sockets, one for `AF_INET` and another for `AF_INET6`, when you would like to accept both IPv4 and IPv6 traffic.

It should also be noted that malicious parties can take advantage of the complexity presented above, and are able to bypass access control, if the target node routes IPv4 traffic to `AF_INET6` socket. Users are advised to take care handling connections from IPv4 mapped address to `AF_INET6` sockets.

SEE ALSO

`ioctl`(2), `socket`(2), `sysctl`(3), `icmp6`(4), `intro`(4), `ip6`(4), `tcp`(4), `ttcp`(4), `udp`(4)

STANDARDS

Tatsuya Jinmei and Atsushi Onoe, *An Extension of Format for IPv6 Scoped Addresses*, internet draft, draft-ietf-ipngwg-scopedaddr-format-02.txt, June 2000, work in progress material.

HISTORY

The **inet6** protocol interfaces are defined in RFC2553 and RFC2292. The implementation described herein appeared in the WIDE/KAME project.

BUGS

The IPv6 support is subject to change as the Internet protocols develop. Users should not depend on details of the current implementation, but rather the services exported.

Users are suggested to implement "version independent" code as much as possible, as you will need to support both inet(4) and **inet6**.

NAME

gif — generic tunnel interface

SYNOPSIS

pseudo-device gif

DESCRIPTION

The **gif** interface is a generic tunnelling pseudo device for IPv4 and IPv6. It can tunnel IPv[46] traffic over IPv[46]. Therefore, there can be four possible configurations. The behavior of **gif** is mainly based on RFC2893 IPv6-over-IPv4 configured tunnel. On NetBSD, **gif** can also tunnel ISO traffic over IPv[46] using EON encapsulation.

Each **gif** interface is created at runtime using interface cloning. This is most easily done with the `ifconfig`(8) **create** command or using the *gifconfig_*⟨`interface`⟩ variable in `rc.conf`(5).

To use **gif**, administrator needs to configure protocol and addresses used for the outer header. This can be done by using `gifconfig`(8), or `SIOCSIFPHYADDR` ioctl. Also, administrator needs to configure protocol and addresses used for the inner header, by using `ifconfig`(8). Note that IPv6 link-local address (those start with `fe80::`) will be automatically configured whenever possible. You may need to remove IPv6 link-local address manually using `ifconfig`(8), when you would like to disable the use of IPv6 as inner header (like when you need pure IPv4-over-IPv6 tunnel). Finally, use routing table to route the packets toward **gif** interface.

gif can be configured to be ECN friendly. This can be configured by `IFF_LINK1`.

ECN friendly behavior

gif can be configured to be ECN friendly, as described in `draft-ietf-ipsec-ecn-02.txt`. This is turned off by default, and can be turned on by `IFF_LINK1` interface flag.

Without `IFF_LINK1`, **gif** will show a normal behavior, like described in RFC2893. This can be summarized as follows:

Ingress Set outer TOS bit to `0`.

Egress Drop outer TOS bit.

With `IFF_LINK1`, **gif** will copy ECN bits (`0x02` and `0x01` on IPv4 TOS byte or IPv6 traffic class byte) on egress and ingress, as follows:

Ingress Copy TOS bits except for ECN CE (masked with `0xfe`) from inner to outer. Set ECN CE bit to `0`.

Egress Use inner TOS bits with some change. If outer ECN CE bit is `1`, enable ECN CE bit on the inner.

Note that the ECN friendly behavior violates RFC2893. This should be used in mutual agreement with the peer.

Security

Malicious party may try to circumvent security filters by using tunnelled packets. For better protection, **gif** performs martian filter and ingress filter against outer source address, on egress. Note that martian/ingress filters are no way complete. You may want to secure your node by using packet filters. Ingress filter can be turned off by `IFF_LINK2` bit.

Miscellaneous

By default, **gif** tunnels may not be nested. This behavior may be modified at runtime by setting the sysctl(8) variable *net.link.gif.max_nesting* to the desired level of nesting. Additionally, **gif** tunnels are restricted to one per pair of end points. Parallel tunnels may be enabled by setting the sysctl(8) variable *net.link.gif.parallel_tunnels* to 1.

SEE ALSO

inet(4), inet6(4), gifconfig(8)

R. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", *RFC2893*, August 2000, ftp://ftp.isi.edu/in-notes/rfc2893.txt.

Sally Floyd, David L. Black, and K. K. Ramakrishnan, *IPsec Interactions with ECN*, December 1999, draft-ietf-ipsec-ecn-02.txt.

HISTORY

The **gif** device first appeared in WIDE hydrangea IPv6 kit.

BUGS

There are many tunnelling protocol specifications, defined differently from each other. **gif** may not interoperate with peers which are based on different specifications, and are picky about outer header fields. For example, you cannot usually use **gif** to talk with IPsec devices that use IPsec tunnel mode.

The current code does not check if the ingress address (outer source address) configured to **gif** makes sense. Make sure to configure an address which belongs to your node. Otherwise, your node will not be able to receive packets from the peer, and your node will generate packets with a spoofed source address.

If the outer protocol is IPv4, **gif** does not try to perform path MTU discovery for the encapsulated packet (DF bit is set to 0).

If the outer protocol is IPv6, path MTU discovery for encapsulated packet may affect communication over the interface. The first bigger-than-pmtu packet may be lost. To avoid the problem, you may want to set the interface MTU for **gif** to 1240 or smaller, when outer header is IPv6 and inner header is IPv4.

gif does not translate ICMP messages for outer header into inner header.

In the past, **gif** had a multi-destination behavior, configurable via IFF_LINK0 flag. The behavior was obsoleted and is no longer supported.

It is thought that this is not actually a bug in gif, but rather lies somewhere around a manipulation of an IPv6 routing table.

NAME

stf — 6to4 tunnel interface

SYNOPSIS

pseudo-device stf

DESCRIPTION

The **stf** interface supports “6to4” IPv6 in IPv4 encapsulation. It can tunnel IPv6 traffic over IPv4, as specified in RFC3056.

For ordinary nodes in 6to4 site, you do not need **stf** interface. The **stf** interface is necessary for site border router (called “6to4 router” in the specification).

Due to the way 6to4 protocol is specified, **stf** interface requires certain configuration to work properly. Single (no more than 1) valid 6to4 address needs to be configured to the interface. “A valid 6to4 address” is an address which has the following properties. If any of the following properties are not satisfied, **stf** raises runtime error on packet transmission. Read the specification for more details.

- matches 2002:xxyy:zzuu::/48 where xxyy:zzuu is a hexadecimal notation of an IPv4 address for the node. IPv4 address can be taken from any of interfaces your node has. Since the specification forbids the use of IPv4 private address, the address needs to be a global IPv4 address.
- Subnet identifier portion (48th to 63rd bit) and interface identifier portion (lower 64 bits) are properly filled to avoid address collisions.

If you would like the node to behave as a relay router, the prefix length for the IPv6 interface address needs to be 16 so that the node would consider any 6to4 destination as “on-link”. If you would like to restrict 6to4 peers to be inside certain IPv4 prefix, you may want to configure IPv6 prefix length as “16 + IPv4 prefix length”. **stf** interface will check the IPv4 source address on packets, if the IPv6 prefix length is larger than 16.

stf can be configured to be ECN friendly. This can be configured by IFF_LINK1. See gif(4) for details.

Please note that 6to4 specification is written as “accept tunnelled packet from everyone” tunnelling device. By enabling **stf** device, you are making it much easier for malicious parties to inject fabricated IPv6 packet to your node. Also, malicious party can inject an IPv6 packet with fabricated source address to make your node generate improper tunnelled packet. Administrators must take caution when enabling the interface. To prevent possible attacks, **stf** interface filters out the following packets. Note that the checks are no way complete:

- Packets with IPv4 unspecified addrss as outer IPv4 source/destination (0.0.0.0/8)
- Packets with loopback address as outer IPv4 source/destination (127.0.0.0/8)
- Packets with IPv4 multicast address as outer IPv4 source/destination (224.0.0.0/4)
- Packets with limited broadcast address as outer IPv4 source/destination (255.0.0.0/8)
- Packets with subnet broadcast address as outer IPv4 source/destination. The check is made against subnet broadcast addresses for all of the directly connected subnets.
- Packets that does not pass ingress filtering. Outer IPv4 source address must meet the IPv4 topology on the routing table. Ingress filter can be turned off by IFF_LINK2 bit.
- The same set of rules are appplied against the IPv4 address embedded into inner IPv6 address, if the IPv6 address matches 6to4 prefix.

It is recommended to filter/audit incoming IPv4 packet with IP protocol number 41, as necessary. It is also recommended to filter/audit encapsulated IPv6 packets as well. You may also want to run normal ingress filter against inner IPv6 address to avoid spoofing.

By setting the IFF_LINK0 flag on the **stf** interface, it is possible to disable the input path, making the direct attacks from the outside impossible. Note, however, there are other security risks exist. If you wish to use the configuration, you must not advertise your 6to4 address to others.

EXAMPLES

Note that 8504:0506 is equal to 133.4.5.6, written in hexadecimals.

```
# ifconfig ne0 inet 133.4.5.6 netmask 0xffffff00
# ifconfig stf0 inet6 2002:8504:0506:0000:a00:5aff:fe38:6f86 \
        prefixlen 16 alias
```

The following configuration accepts packets from IPv4 source 9.1.0.0/16 only. It emits 6to4 packet only for IPv6 destination 2002:0901::/32 (IPv4 destination will match 9.1.0.0/16).

```
# ifconfig ne0 inet 9.1.2.3 netmask 0xffff0000
# ifconfig stf0 inet6 2002:0901:0203:0000:a00:5aff:fe38:6f86 \
        prefixlen 32 alias
```

The following configuration uses the **stf** interface as an output-only device. You need to have alternative IPv6 connectivity (other than 6to4) to use this configuration. For outbound traffic, you can reach other 6to4 networks efficiently via **stf**. For inbound traffic, you will not receive any 6to4-tunneled packets (less security drawbacks). Be careful not to advertise your 6to4 prefix to others (2002:8504:0506::/48), and not to use your 6to4 prefix as a source.

```
# ifconfig ne0 inet 133.4.5.6 netmask 0xffffff00
# ifconfig stf0 inet6 2002:8504:0506:0000:a00:5aff:fe38:6f86 \
        prefixlen 16 alias deprecated link0
# route add -inet6 2002:: -prefixlen 16 ::1
# route change -inet6 2002:: -prefixlen 16 ::1 -ifp stf0
```

SEE ALSO

gif(4), inet(4), inet6(4)

http://www.6bone.net/6bone_6to4.html

Brian Carpenter and Keith Moore, *Connection of IPv6 Domains via IPv4 Clouds*, RFC, 3056, February 2001.

Jun-ichiro itojun Hagino, *Possible abuse against IPv6 transition technologies*, draft-itojun-ipv6-transition-abuse-01.txt, July 2000, work in progress.

HISTORY

The **stf** device first appeared in WIDE/KAME IPv6 stack.

BUGS

No more than one **stf** interface is allowed for a node, and no more than one IPv6 interface address is allowed for an **stf** interface. It is to avoid source address selection conflicts between IPv6 layer and IPv4 layer, and to cope with ingress filtering rule on the other side. This is a feature to make **stf** work right for all occasions.