WIDE Technical-Report in 2012

IPv6 only Network 構築と検証実験

wide-tr-hazeyama-ipv6-only-network-00.pdf

WIDE Project : http://www.wide.ad.jp/

If you have any comments on WIDE documents, please contact to board@wide.ad.jp

Title: IPv6 only Network 構築と検証実験

Author(s): 櫨山寛章 (hiroa-ha@is.naist.jp), 上野幸杜 (eden@wide.ad.jp), 佐藤 弘崇 (satu@wide.ad.jp), 石橋 尚武 (take@hongo.wide.ad.jp), 横石 雄大 (dokan@wide.ad.jp), 山岸 祐大 (yummy@sfc.wide.ad.jp), 石原 知洋 (sho@c.u-tokyo.ac.jp)

Date: 2012-1-25

IPv6 only Network 構築と検証実験

櫨山寛章 (hiroa-ha@is.naist.jp)
上野幸杜 (eden@wide.ad.jp)
佐藤 弘崇 (satu@wide.ad.jp)
石橋 尚武 (tak@hongo.wide.ad.jp)
横石 雄大 (dokan@wide.ad.jp)
山岸 祐大 (yummy@sfc.wide.ad.jp)
石原 知洋 (sho@c.u-tokyo.ac.jp)

2012 年 1 月 17 日

概要

2011 年 9 月 WIDE 合宿の実験のメインテーマとして、参加者に DHCP6 と DNS64/NAT64 のアドレス変換による IPv6 only アクセス環境のみを基本的に提供し、ユーザが普段利用している OS やアプリケーションの IPv6 対応状況の確認や、IPv6 のみを有効にする設定や IPv6 only アクセス環境で快適に生活するための知見、NAT64/DNS64 を実際に利用した際の問題点の洗い出しを行った。また、IPv4 コネクティビティの提供方法として SA46T による 464 トンネルを用いて WIDE バックボーンの IPv4 グローバルアドレスを MAC アドレス登録制の DHCP4 で提供し、あわせて、IIJ で開発している SEIL 4RD ルータと vyatta の 4RD 拡張を用いて 4RD による IPv4 プライベートアドレスの提供実験を行った。2 つの IPv4 提供実験はともにアドレス変換、トンネリング技術である。また、対外線を IPoE 方式で提供される IPv6 インターネット接続サービスのみを利用し、慶應義塾大学 上野によって作成された IPv6 用 LT2P を用いて WIDE バックボーンへの L2 トンネルを作成し WIDE バックボーンからのルーティングをおこなうため、多重カプセリングの影響を把握する検証も参加者を交えて実施した。

1 はじめに

本章では、2011 年 9 月 6 日から 9 月 9 日の 4 日間に開催された WIDE プロジェクト 2011 年 9 月合宿で実施された IPv6 only アクセスネットワーク利用実験に関する報告を行う。WIDE プロジェクトでは、WIDE 合宿の実験ネットワークや WIDE バックボーンにて IPv4/IPv6 のデュアルスタック環境の運用を長年行ってきた。また、2011 年 6 月に実施された World IPv6 day の前後で多くの商用 ISP が IPv6 接続サービスを提供開始した。一方で、実際に IPv6 だけ提供されるアクセスネットワークとアドレス変換サービスを利用した場合、現在提供されている IPv4 もしくはデュアルスタックによる接続サービスと比較して、どの程度利用できるのかに関しての知見は、世界的に見てあまり蓄積されていない。そこで、IPv6 のみを利用してどこまでサービスネットワークとして構築し、提供できるかという点と、IPv6 しかない環境ではユーザの利用においてどのような問題が生じるのかを洗い出すために、WIDE 合宿ネットワーク全体のテーマとして IPv6 only network に着目して合宿ネットワークを設計し、構築、運用した。

2 The Camp 1109 ネットワークの設計

合宿ネットワーク全体のテーマとして、IPv6 only network を構築し、参加者全員が IPv6 only network で生活するという方針は、2011 年 5 月に北陸先端科学技術大学院大学で実施された WIDE 研究会の研究発表にて議論され、決定した。しかしながら、WIDE 合宿参加者の多くは、特に企業からの参加者は、社内ネットワークサービスが IPv6 に対応していないことが多く、IPv6 接続性のみを提供しても日常の業務（メールの閲覧など）を実施することができない。そこで、合宿ネットワーク準備委員会（以降 Net PC）では、WIDE 合宿の実験ネットワーク（camp-net）にて、参加者が EMOBILE などの商用モバイルインターネットサービスに退避せずに 4 日間生活できる、現実的なサービスとして IPv6 only network 環境の提供方法の模索を始めた。本節では、合宿ネットワーク NOC チームで検討し、いくつかの事前検証を通して設計した camp-net の

概要、実験趣旨および技術解説を行う。

2.1 実験概要

まず、NOC チームにより、1）DHCP6 の利用実験、2）NAT64 / DNS64 による IPv6 / IPv4 アドレス変換の利用実験、3）SA46T による IPv6 バックボーンを介した IPv4 ネットワーク接続性提供実験、4）WPA2 EAP-TLS による無線 LAN アクセス認証、の 4 つの実験を実施した。これに合わせて、慶應義塾大学、IIJ、NTT 東日本、インターネットマルチフィードによって構成された実験チームによる 4RD による IPv6 および IPv4 ネットワーク接続性提供実験が実施された。

2.2 対外接続

図 1 に示すように、今回の camp-net では IPv6 による 2 つの対外線を用意した。ひとつは松代ロイヤルホテルと慶応大学湘南藤沢キャンパス（SFC）を結ぶ衛星回線であり、もうひとつは NTT 東日本により提供された FTTH 回線である。

まず衛星回線に関して説明する。衛星回線は 1.5GHz 帯を用いた回線で下り 1.5Mbps, 上り 512 Kbps の回線利用申請を行った。衛星回線は合宿会期中通して安定していた。衛星回線を挟み SFC 側のルータと camp-net のコアルータとの間で VLAN を構築した。

次に、FTTH 回線について説明する。FTTH 回線では NTT 東日本をアクセスキャリア、インターネットマルチフィードを VNE （Virtual Network Enabler）、IIJ を IPv6 ISP として構成された 2 種類の商用 IPv6 アクセスサービスを検証した。設営に当たる 2011 年 9 月 5 日から 9 月 6 日午後 8 時まではフレッツ光ネクストの IPv6 オプション付きの契約を行い利用した。9 月 6 日午後 8 時からは、4RD の検証を実施するために、さらに光電話オプションをつけて再契約およびオプション変更に伴う工事を実施し利用した。光電話オプションのあり・なしの差異は、光電話オプションなしでは IPv6 は RA にて/64 の IPv6 アドレスが割り当てられ、一方、光電話オプションありの場合は DHCP6 により/48 の IPv6 が割り当てられる。FTTH 回線の変更を図示すると図 1(b) のようになる。

IPv6 インターネットへの接続性は IIJ mio FiberAccess/NF for IPv6 ネイティブサービスを契約し利用した。9 月 6 日午後 8 時までは、図 1(a) に示すように、camp-net で用意した L2TP ゲートウェイに FTTH 回線上の RA で割り当てられる/64 プレフィックス長の IPv6 アドレスを設定した。一方、9 月 6 日午後 8 時以降は、図 1(b) に示すように、IIJ が研究開発している SEIL ホームルータの WAN 側インターフェースに FTTH 回線上の DHCP6 で割り当てられる/48 プレフィックス長の IPv6 アドレスを割り当てて、4RD のための prefix delegation を実施できる構成に設定した。

また、WIDE プロジェクトで運用管理する IPv6 アドレスブロックを合宿ネットワークで利用するために、松代ロイヤルホテルと慶応 SFC の間を結ぶ L2TP トンネルを FTTH 回線上に構築した。L2TP ゲートウェイは Linux Debian squeeze (kernel 2.6.32) で構築したサーバ上に NOC の一人である上野が作成した IPv6 用 L2TP 実装 (v6tun [1]）を用いた。筑波大学で開発されているオープンソース VPN ソフトウェアである ut-vpn [2] と比較した事前検証では、v6tun は TCP で 719 Mbps、UDP で 738 Mbps のスループットを記録し、他方、ut-vpn [2] は TCP で 428 Mbps、UDP で 410 Mbps のスループットであったため、L2TP として v6tun を採用した。

2.3 NAT64 / DNS64

今回の camp-net ではユーザへの IPv6 アドレス割り当てを ISC DHCP6 実装 [3] を用いて提供した。一方、多くのネットワークが World IPv6 day を経験した後でも IPv6 に未対応であるため、6to4 アドレス変換技術を camp-net の IPv6 only 接続性実験に組み入れることとした。そこで、camp-net の要求を満たす、2011 年 9 月現在で構築可能な最良の NAT64 [4,5] と DNS64 [6] を用いた 6to40 アドレス変換の実装を検証した。

NOC チームの NAT64 および DNS64 への技術要求事項は以下のとおりである。

実装はオープンソースソフトウェアであること
: 理由は、何らかのトラブルが発生した場合にデバッグが NOC チームによって行えるようにするためである。

正常に動作する **DNS64** 実装であること
: 理由は、正常に動作しない実装をサービスに用いることはできないためである。

表 1: 無線 LAN 設定と接続性、アドレス変換技術の一覧

ESSID	Accounting	Channel	Address version	Address scope	DNS	Address allocation	Trans. / Encap.
widecamp	WPA2 EAP-TLS	11b/g/n	v6	global	DNS64	DHCP6 (automatic)	NAT64
			v4	global	DNS64	DHCP4 (registration)	SA46T
widecamp-a	WPA2 EAP-TLS	11a	v6	global	DNS64	DHCP6 (automatic)	NAT64
			v4	global	DNS64	DHCP4 (registration)	SA46T
widecamp-sat	WPA2 EAP-TLS	11b/g/n	v6	global	DNS64	DHCP6 (automatic)	NAT64
			v4	N/A	N/A	N/A	N/A
widecamp-nowep (hidden)	MAC addr. Auth.	11b/g/n	v6	global	DNS64	DHCP6 (automatic)	NAT64
			v4	global	DNS64	DHCP4 (registration)	SA46T
widecamp-iij	WPA2 EAP-TLS	11b/g/n	v6	global	N/A	RA from SEIL (automatic)	N/A
			v4	N/A	N/A	N/A	4RD
widecamp-4rd	WPA2 EAP-TLS	11b/g/n	v6	global	N/A	RA from SEIL (automatic)	N/A
			v4	private	Proxy resolver	DHCP4 from SEIL (automatic)	4RD

正常に動作する **NAT64** 実装であること
: DNS64 と同様に正常に動作しない NAT64 実装をサービスに用いることはできないためである。特に変換によってペイロードを破壊しない事が必須である。

他の **NAT44** とのカスケード接続を利用しないこと
: 理由としては、NAT64 のみの不具合を正しく把握し、カスケード接続によって生じる可能性のあるトラブルを回避するためである。

DNS64 と NAT64 の実装の評価はプレホットステージ期間である 7 月 1 日から開始した。まず、DNS64 の実装に関する評価について説明する。評価対象とした DNS64 の実装は ISC bind 9.8 p4 [7]、NLnet labs unbound [8] および Viagénie ecdysis [9] の 3 種類を検証した。このうち、2011 年 7 月の段階で正常に動作したのは bind のみであったため、camp-net で用いる実装として bind を採用した。

次に NAT64 の実装に対する評価イについて説明する。評価した NAT64 の実装は linuxnat64 [10]、tayga [11] および ecdysis [9] の 3 種類である。

linuxnat64 と ecdysis は ステートフル NAT64 [5] の実装であり、一方、tayga はステートレス NAT64 [4] の実装である。評価の結果、camp-net では linuxnat64 を採用した。採用した理由は、まず、tayga のステートレス NAT64 は収容するユーザ数だけ IPv4 アドレスを必要とするが、NAT44 のカスケードを利用しないという要求事項と、camp-net で参加者全員を収容できるほど十分なグローバル IPv4 アドレスを保持していないことから、検討段階で tayga ステートレス NAT64 は要求事項を満たせなかった。次に、linuxnat64 と ecdysis を実際に動作させて評価したところ、2011 年 7 月の段階では、ecdysis の実装では TCP ペイロードを破壊する挙動が確認されたため、特に不具合もなく正常に動作した linuxnat64 を camp-net では採用した。bind と linuxnat64 の設定例は 7 章に付録として記載しておく。

2.4 IPv4 over IPv6 カプセル化技術

6to4 アドレス変換とは別に、camp-net では IPv6 未対応の OS やアプリケーションを利用する参加者のために、464 カプセル化技術の検討も行った。camp-net NOC が公式サービスとして用意する 464 カプセル化技術としては SA46T [12–15] のソフトウェア実装を採用した。この SA46T ソフトウェア実装は慶應義塾大学と富士通との共同研究で研究開発され、過去の WIDE 合宿での実験や JGN-X と ThaiSARN との間でのビデオストリームを用いた検証 [16] などによって実績があるため採用した。

また、追加実験として 4RD [17] の検証も実施した。4RD 検証実験は慶應義塾大学、NTT 東日本、インターネットマルチフィードおよび IIJ によって構成された

4RD 検証チームから 9 月 1 日に急遽提案され、camp-net でこの実験を受け入れたことによって実施が決まった。4RD 検証実験では IIJ が開発している SEIL ホームルータ上での 4RD 実装を 4RD-CE (4RD Customer Edge Router) として合宿地に設置し、vyatta の 4RD 実装 [18] を 4RD-BR（4RD Boarder Router) として WIDE バックボーン藤沢 NOC 内に設置した。4RD-CE で IPv6 パケットにカプセリングされた IPv4 パケットは NTT 東日本のフレッツ網からインターネットマルチフィードのバックボーンを経由して IIJ の IPv6 網に入り、そこから WIDE バックボーンにルーティングされ 4RD-BR で IPv4 パケットに戻されることになる。

2.5 WiFi アクセスとアドレス割り当て

ユーザアクセス周りのトポロジーは図 1(c) に示すような構成とした。また、表 1 では図 1(c) に対応した無線 LAN のチャネル、認証方式、VLAN、利用 IP アドレスタイプ、IP アドレスの割り当て方、アドレス変換方式の一覧である。基本的に、無線 LAN アクセスは WIDE 個人証明書を用いた WPA2 EAP-TLS により提供した。WIDE 個人証明書を用いた WPA2 EAP-TLS による無線 LAN アクセス認証は 2008 年 3 月合宿から camp-net NOC により継続して行われている実験である。また、ESSID **widecamp-nowep** を、WPA2 EAP-TLS が行えないデバイスや個人証明書をインストールし忘れたユーザのためのバックアップとして隠し ESSID に設定して用意した。ESSID **widecamp-nowep** での認証は文献 [19] で説明されている radius を用いた MAC アドレスと接続 AP の対応によるレイヤ 2 レベルの認証（アカウンティング）により行った。

IP アドレス自動割り当てと名前解決用のリゾルバの自動設定は ISC DHCP 実装 [3] を用いて DHCP4 と DHCP6 を提供した。ただし、ESSID **widecamp-4rd** では 4RD-CE として動作している SEIL ホームルータが RA、NAT44、DHCP4 および name proxy として動作しているため、SEIL ホームルータによって IP アドレスと名前解決用のリゾルバの自動設定を行った。

2.6 物理機材およびクラウド資源の活用

松代ロイヤルホテルでの設営作業を省力化するために、2011 年 9 月合宿ではほとんどのサーバをクラウド環境に構築した。利用したクラウド環境は、StarBED [20] 上の CISCO UCS サーバを 6 台と WIDE 藤沢 NOC に設置した WIDE Cloud Controller (WCC) [21] のクローンサーバを用いて構築した。藤沢の WCC サーバはクラウド環境のコントローラおよび NFS サーバとして利用し、StarBED 上のサーバは仮想マシンを配置するスレイブノードとして利用した。仮想マシンとしては qemu-kvm 0.14.1、Linux Kernel 3.0.4 を用い、libvirt 0.9.4 を用いて仮想マシンの操作を行った。また、StarBED と藤沢の WCC サーバ間は WIDE バックボーン上に広域 VLAN を設定して同一 L2 セグメントに収容できるようにした。詳細は、第 4 部「クラウドコンピューティング基盤の構築と運用」や第 28 部「大規模な仮設ネットワークテストベッドの設計・構築とその運用」を参照してほしい。

3 実験

本節では、実験結果について報告する。前節までで述べたように camp-net では「WPA2 EAP-TLS によるアクセス認証の検証」、「NAT64/DNS64 による IPv6 only アクセス環境の検証」、「SA46T による 464 カプセル化の検証」、「4RD による 464 カプセル化の検証」の 4 つの実験が行われた。

3.1 実験概要とタイムライン

表 2 は合宿期間中に発生したイベントのタイムラインを示す。「WPA2 EAP-TLS によるアクセス認証の検証」と「NAT64/DNS64 による IPv6 only アクセス環境の検証」は 9 月 6 日の午前 10 時から開始した。初日（9 月 6 日）の段階では、**widecamp-nowep** および SA46T を通した IPv4 グローバルアドレスによるアクセスを行うための MAC アドレス登録ページは隠した状態で、IPv6 のみのアクセス環境を参加者に強制的に利用させるようにした。WIDE 個人証明書の入れ忘れなどで IPv4 アドレスで構築された社内網にアクセスしなければならないと言った IPv4 アドレスが必要な参加者には、NOC にヘルプデスクを用意し個別対応を行った。

また、EMOBILE などの商用モバイルインターネットと個人用 WiFi ルータを持ちこむ参加者は昨今増え

表 2: Time line of experiments

Date	Events
2:00 PM, Sep. 5th	設営開始
4:00 PM, Sep. 5th	FTTH 回線上への IPv6 L2TP トンネルの設定終了
4:50 PM, Sep. 5th	衛星回線の設定終了
3:00 AM, Sep. 6th	DHCP ヘルパーの動作不良（チェックサムエラー）を特定、camp-net トポロジーから削除
9:00 AM, Sep. 6th	衛星回線の動作検証終了
10:00 AM, Sep. 6th	started the WPA2 EAP-TLS の検証と DNS64/NAT64 による IPv6 only アクセス環境の検証を開始
8:00 PM, Sep. 6th	4RD 検証実験のための FTTH 回線の設定変更を開始
9:00 PM, Sep. 6th	4RD 検証実験のための FTTH 回線の設定変更が終了
10:00 PM, Sep. 6th	4RD 検証実験のための設定終了と NOC チームでの事前検証開始
1:30 PM, Sep. 7th	SA46T による IPv4 アクセス環境用の MAC アドレス登録ページを開設し、SA46T による IPv4 アクセス環境を参加者に解放
3:15 PM, Sep. 7th	合宿参加者を交えた 4RD 検証実験の開始
4:15 PM, Sep. 7th	fixed the mis-configuration of firewall on MAC アドレス登録ウェブサーバ上の firewall の設定ミスを特定し、修正
5:47 PM, Sep. 7th	radius サーバの設定ミスを特定し、修正
10:00 PM, Sep. 7th	ほぼすべての参加者が WPA2 EAP-TLS または widecamp-nowep で ESSID のいづれかに接続し、検証実験に参加できる状態になったことを確認
1:42 PM, Sep. 8th	DNS64 サーバ上で DNSSEC を有効にする
4:30 PM, Sep. 8th	有志による SA46T および 4RD のパフォーマンス測定開始
5:30 PM, Sep. 8th	ns.wide 上の設定ミス（lame delegation）を特定、修正
6:00 PM, Sep. 8th	負荷向上により、DNS64 サーバを WIDE クラウド上から独立した物理サーバへ移行
8:00 PM, Sep. 8th	有志による SA46T と 4RD の比較検証開始
10:00 PM Sep. 8th	FTTH 回線のパフォーマンスチューニングを開始
11:00 PM Sep. 8th	FTTH 回線のパフォーマンスチューニング終了
11:00 AM Sep. 9th	camp-net を停止、撤収開始
2:30 PM Sep. 9th	撤収終了

ているため、初日は NOC チームで Fox Hunting チームを結成し、個人用 WiFi ルータによる rouge AP の取り締まりを実施した。rouge AP を用いて IPv4 ネットワークにアクセスしていた参加者に対しては、ホテルの宿泊部屋で松代ロイヤルホテルが用意している IPv4 ネットワークを使うか、rouge AP として実験に影響しない程度に会場から遠く離れて利用する、または NOC のヘルプデスクにて有線で提供しているの SA46T 経由の IPv4 ネットワークを利用するように促し、できる限り IPv6 only アクセス環境の検証実験に参加するように取り組んだ。

表 2 に示すように、9 月 7 日午後 1 時半に SA46T によるグローバル IPv4 アクセスを利用するための MAC アドレス登録ウェブページを開設し、参加者にグローバル IPv4 アクセスの利用を開放した。また、9 月 7 日午後 3 時 15 分から 4RD による NAT44 を介した IPv4 アクセスの利用を参加者に開放した。上記のタイムラインから、ほとんどの参加者は少なくとも丸 1 日は IPv6 のみのアクセス環境で会場では過ごしたことになる。

3.2 アンケート

ここでは実験に対して参加者に行ったアンケート結果を報告する。アンケート結果は WIDE プロジェクトのメンバーであれば [22] で閲覧することができる。2011 年 9 月合宿参加者は総数 153 名でそのうち 2011 年 9 月 17 日の午後 1 時半までにアンケートに回答した参加者は 110 名、回答率 71.9% であった。図 2 は参加者が利用したアクセスネットワークの組み合わせの割合、および NOC が実施した実験（IPv6 only および SA46T）に対する回答の集計結果である。一方、図 3 は 4RD 検証実験に対するアンケートの集計結果である。

図 2(a) は参加者が利用したアクセスネットワークの組み合わせの割合を集計した結果でさる。驚くことに 30 名（19.6%）の参加者が合宿期間中を通して NAT64 / DNS64 によるアドレス変換を伴う IPv6 only のアクセスネットワークだけで生活したと回答した。IPv6 と SA46T による IPv4 アクセスネットワークを利用したと申告した参加者は 7 名いた。解析したところ、初日の WPA2 EAP-TLS の設定または IPv6 only 設定に失敗し、IPv4 アドレス利用が開放された段階で SA46T による IPv4 アクセスネットワークに逃げ込んだ参加者が多かった。33 名の参加者は IPv6 と 4RD 環境を利用したと申告した。これらのユーザは MAC アドレス登録をしてまで IPv4 アクセスを利用しようとは思わなかったユーザではないかと推測できる。34 名の参加者は全ての組み合わせを試し、その多くは 4RD と SA46T の比較実験に参加した参加者であった。図 2(c) および図 2(d) から 90 名の参加者（58.8%）は今回 camp-net で提供した IPv6 only アクセスの品質や SA46T で提供した IPv4 アクセスの品質に満足したようであった。

SA46T や 4RD などの IPv4 アクセスを利用した主な理由を以下に列挙する。

理由 1) 持ってきた PC に WIDE 個人証明書を入れ忘れたが、メールサーバが IPv4 のみで DNS 登録もさ

れていないため、ssh でログインするために IPv4 アクセスを利用した。

理由 **2)** 利用している OS（Windows XP や Mac OS X 10.5.8 など）で IPv6 設定が行えなかった、もしくは設定の仕方がわからずあきらめた。

理由 **3)** Lenovo ThinkPad' の無線 LAN 設定では IPv6 only の設定がうまく設定できなかったため、IPv4 アクセスを利用した。

理由 **4)** skype や windows live messanger などの IPv6 only 環境に未対応のアプリケーションを仕事で利用するために、IPv4 アクセスを利用した。

理由 **5)** IPv4 アドレスしか設定されていない会社の VPN サーバ（IPsec VPN または PPTP）に接続するために、IPv4 アクセスを利用した。

理由 **6)** Andoroid OS で WPA2 EAP-TLS を設定しようとしたが、名前解決を IPv4 で行っている挙動を示したので IPv4 アクセスを利用した。

理由 **7)** あるウェブページが AAAA での名前解決時に ServFAIL エラーを返し、閲覧できなかったため、IPv4 アクセスを利用した。

理由 **8)** DNS64 の応答速度が遅くなったため、IPv4 アクセスを利用した。

理由 **9)** VMWare の NAT から外部にアクセスできなくなったために、IPv4 アクセスを利用した。

理由 **10)** SA46T と 4RD の比較検証実験に参加するために IPv4 アクセスを利用した。

図 3(b)、図 3(c)、図 3(d)、図 3(e) および 図 3(f) は 4RD 検証結果に対するアンケート結果をまとめたものである。ほとんどの参加者は提供された 4RD 環境に満足した。しかしながら、一部の参加者から 4RD 環境での不具合が報告された。報告された不具合に関しては次節 3.3 に記載する。

3.3 報告されたトラブル

本節では NOC や合宿参加者メーリングリストに報告されたトラブルをまとめる。

3.3.1 WPA2 EAP-TLS に関するトラブル

不幸なことに、9 月 6 日から 9 月 7 日かけて最も多かったトラブルは WPA2 EAP-TLS に関するトラブルであった。

WPA2 EAP-TLS に関するトラブルで最も多かった項目は「WIDE 個人証明書の入れ忘れて来てしまったが、メールサーバが IPv4 のみなのでどうしたらよいのか?」という相談であった。NOC ヘルプデスクに直接相談に来た参加者には MAC アドレスの登録を手動で行い **widecamp-nowep** もしくは有線接続で SA46T による IPv4 アクセスを利用し、NOC のサポートで WPA2 EAP-TLS の設定を行った。しかしながら、NOC ヘルプデスクに相談しにくる参加者は少なく、相談しに来なかった参加者は自分の部屋で松代ロイヤルホテルが提供する IPv4 アクセスを利用するか、商用モバイルインターネットサービスを使い、rouge AP を立ち上げていることを NOC 注意されるという参加者が多かった。

また、9 月 7 日まで radius サーバの設定ミスがあり、認証が不安定であったこともトラブルとして上がった。設定ミスは WIDE moca ワーキンググループの木村泰司氏らの手助けにより、9 月 7 日中に解決された。

3.3.2 IPv6 対応に関するトラブル

参加者からは次のような IPv6 対応に関するトラブルが報告された。

- フォールバックルーチンが長すぎる

  IPv6 only アクセスに接続したほとんどの参加者がこのトラブルを報告した。このトラブルは Windows ユーザや Mac OS ユーザから多く報告された。このトラブルの原因は、Windows 7 や Mac OS X では IPv4 プロパティが有効になっている場合、接続時に IPv4 の対外接続性の確認も行っているためで IPv4 接続確認のタイムアウトで 1 分から 2 分待たされるためである。NOC チームからの推奨設定として、IPv6 only アクセス環境で IPv4 を使わない場合は IPv4 プロパティを無効にすることをアナウンスした。実際に設定した参加者の多くは「IPv4 の設定を無効にするのは衝撃的だが、実際、非常に快適になった。」と回答した。

- **DHCP6** 未対応によるトラブル

  Windows XP や Mac OS X 10.6 (Snow Leopard) 以前の OS では DHCP6 に対応していないため、DNS リゾルバの設定の仕方がわからないという参加者が多く存在した。NOC チームでは、DHCP6 未対応の OS と手動で設定する DNS サーバのアドレスをアナウンスし、設定方法がわからない参加者は NOC ヘルプデスクにて対応した。

  一方で、Windows 7 や Mac OS 10.7 (Lion）を利用していた参加者からは DHCP6 や DNS リゾルバ設定に関するトラブル報告はほとんどなかった。例外として、Think Pad などの無線 LAN アクセス設定アプリケーションを利用していた参加者からは設定がうまく反映されないとの報告を受けた。

- **OS** で **IPv6** のみの設定ができないことによるトラブル

  Windows XP や Android OS (バージョン 2.x のデバイス）では IPv4 プロパティを無効にできない。Windows XP の場合、DNS 名前解決でローカルプロキシを 127.0.0.1 で立ち上げて利用しているためである。Android OS に関しては詳しい原因調査を行えなかったが、参加者からは「おそらく名前解決は IPv4 パケットを使っているようだ」という報告が挙げられた。

- **GUI**、デバイスの **IPv6** 未対応によるトラブル

  いくつかの GUI やハードウェアで IPv6 未対応に関連するトラブルが報告された。初期型の Mac Book Air で Snow Leopard を用いていた参加者から、GUI で設定した DNS64 の設定がいつの間にか消えてしまうというトラブルが報告された。NOC で調査したところ、“**sudo networksetup -setdnsservers ”AirPort” 2001:200:0:ff80::5**” をコマンドラインから実行すると設定が消えない事が判明したため、これを回避方法として指定した。また、参加者が持ち込んだ Apple 社の USB イーサネットアダプタは IPv6 に未対応であることや、古いデバイス・古い OS を利用している参加者からデバイスドライバの再インストールを頻繁に行ったという報告を受けた。

3.3.3 アプリケーションに関するトラブル

ここでは、参加者から報告された、特定のアプリケーションに関するトラブルを記載する。

- アプリケーションの **IPv6** 未対応に関するトラブル

  Arkko がインターネットドラフト [23] で指摘しているように、インスタントメッセンジャーや VoIP アプリケーションの多くはやはり IPv6 未対応のものが多く、利用できなかった。

  また、CVSNT や NOD32 のウィルスデータベースアップデートなど、おそらく IPv4 ソケットしか利用していない Windows 上のアプリケーションがいくつか確認された。

  Mac OX X では、Cocoa フレームワークで実装されたアプリケーションは IPv6 only 環境で動作し、Cocoa フレームワークを利用していないアプリケーションの多くは IPv6 only 環境で利用できないとの報告を参加者から受けた。

  また、HTTP ベース、XMPP ベースのアプリケーションの多くは利用できた。利用できない HTTP ベース、XMPP ベースのアプリケーションはサーバへのアクセスに対し IPv4 アドレス表記をアプリケーション内で埋め込まれているようであった。

- **MTU** ブラックホールによるトラブル

  Path MTU ディスカバリが正常に動作しないことに起因する MTU サイズの不整合でパケットが破棄される問題を MTU ブラックホールと一般的に呼称する。この MTU ブラックホールに関するトラブルもいくつか発生した。

  まず、9 月 5 日の設営時に WIDE クラウド上を経由するパケットが軒並み破棄されるというトラブルに遭遇した。これは WIDE クラウドの MTU サイズが 9000 に設定されていることによるもので、MTU サイズを 1500 に設定することで回避した。

  次に、UDP を用いたアプリケーションで L2TP 上を経由しているパケットで MTU ブラックホールと思われるトラブルが参加者から報告された。残念ながら NOC チームではこのトラブルの原因を合宿中に解析できなかった。

- アドレス・プロトコル変換へプロトコル仕様上対応していないことによるトラブル
  Open VPN や Apple Mobile Me IPSec / PKI ベースの通信など IPSec を用いているアプリケーションはその仕様上 NAT64/DNS64 のアドレス変換には対応できない。そのため、多くの参加者から IPv6 only アクセス環境で会社への VPN が利用できないというトラブルの報告を受けた。また、4RD の環境でも、ある参加者から Apple Mobile Me IPSec / PKI ベースの通信が行えないというトラブルの報告を受けた。こちらの原因特定は残念ながら期間中に実施できなかった。

3.3.4 名前解決に関するトラブル

ここでは名前解決に関するトラブルを報告する。

- **IPv4** アドレス表記に関連するトラブル
  Arkko のインターネットドラフト [23] でも指摘されているように、多くの参加者から IPv6 only アクセス環境から IPv4 サーバに対し IPv4 アドレス表記で通信できないというトラブルの報告が挙げられた。これは、IPv4 アドレス表記を用いると、たとえ getaddrinfo のような IPv6 対応ソケットを用いたとしても、NAT64/DNS64 の仕様上、DNS64 により指定された IPv6 mapped IPv4 アドレスでユーザ側のソケットを作成しないためである。HTTP、SSH、IMAP、SMTP、POPFile などのアプリケーションの設定で指定するサーバを IPv4 表記で書いていたり、サーバのアドレスが IPv4 表記で組み込まれていたりする場合に発生する。設定変更できるものに関しては、対象となる IPv4 アドレスを DNS に登録したうえで、アプリケーション側の設定を FQDN で設定すれば動作するが、参加者の多くが DNS 管理者ではないため、DNS に登録されていない IPv4 アドレスに対応できないケースが散見された。

  また、VNC、PPTP、IPSec などのアプリケーションはこの IPv4 アドレス表記による問題とともに、プロトコル仕様上の問題や MTU ブラックホールの問題も重なり、切り分けが非常に困難であった。

- **AAAA** 逆引きが設定されていないことによるトラブル
  ホットステージ期間中の検証で、一部の商用ウェブサービスが AAAA の逆引きを要求してくることが明らかとなった。そのため、NOC チームでは `camp.wide.ad.jp` ドメインで利用する全ての IPv6 アドレスに対し逆引きを設定した。

- **lame delegation** によるトラブル
  合宿期間中に、ある参加者から一部の商用ウェブサービスが AAAA の逆引きを要求してきているが逆引きができていないという指摘を受けた。NOC チームで調査した結果、`camp.wide.ad.jp` ドメインの上位ドメインである `wide.ad.jp` ドメインを管理する ns.wide.ad.jp 上の設定ミスで lame delegation が発生していることが明らかとなった。設定ミスを修正することでこの問題は解消された。

- **DNS64** の負荷による応答速度劣化によるトラブル
  DNSSEC を DNS64 サーバで有効にしたあと、DNSSEC の検証や参加者のアクセス増加により DNS64 の負荷が向上し応答速度劣化によるトラブルが発生した。当初 DNS64 は WIDE クラウド上の仮想マシンとして動作させていたが、急遽独立した物理サーバとして再構築し、負荷の解消を行った。物理サーバに移行した後は、特に負荷による応答速度の劣化は生じなかった。

- 不適切な **AAAA** 応答を返す権威サーバに関連するトラブル
  合宿期間中、一部のウェブページで NAT64/DNS64 で名前解決が失敗し閲覧できないというトラブルが多くの参加者から報告された。この現象は国内外問わず、航空券・ホテル検索ページで頻繁に発生していた。NOC チームと有志により解析したところ、この原因は RFC 4074 [24] で指摘されている IPv6 移行期に発生する可能性のある不適切な AAAA 応答を返す権威サーバにより DNS64 が AAAA レコードの問い合わせから A レコードの問い合わせにフォールバックせずにエラーコードをクライアントに返してしまう現象により発生していることが明らかとなった。エラーとしては RFC 4074 [24] の 4.2 節で述べられている “Return Name Error”、4.3 節で述べられている “Return Other Erroneous Codes”、4.4 節で述べられている “Return a Broken

Response” を実際に観測した。これらのエラーは権威サーバの実装に起因するものなので NOC チーム側では解決できないトラブルであった。

4 考察

4.1 IPv6 環境への移行における提言

ここでは、合宿ネットワークでの実験で得た知見を通した、NOC チームからの IPv6 環境への移行における提言を記載する。

- **IPv6** 対応が可能であれば外部からアクセスのあるサーバは **IPv6** 対応すべきである
  理由としては、今回の実験から NAT や NAPT を挟んだ環境では容易に MTU ブラックホール問題やスループット低下など、様々な問題が生じ、切り分けが難しいため、デュアルスタックなどで IPv6 対応できるのであれば、今のうちに対応してしまうのが得策である。

- 最新バージョンの **OS** に移行すべきである
  理由としては、Window 7 や Mac OS X 10.7 (Lion) からのトラブル報告はほとんどなく、一方で古いバージョンの OS を利用した参加者からは様々なトラブルの報告を受けたため、運用コスト上は最新バージョンの OS に移行したほうがよい。

- **AAAA** の逆引き設定をすべきである
  理由は AAAA の逆引きを認証に利用する商用ウェブサービスが存在するためである。

- **DNS** サーバやウェブアプライアンスの **DNS** 応答が **RFC** に則っているか否かの検証をしたほうがよい
  理由としては、不適切な DNS 応答はクライアント側では対応できないためである。RFC に則った応答を返せばウェブサービスが IPv6 対応していなくても DNS64/NAT64 のアドレス変換によって IPv6 only ネットワークからもウェブサービスを利用できる。

- **MTU** サイズに注意を払う
  理由は Path MTU ディスカバリが動作しないことが多いためである。

4.2 研究課題

ここでは、今後の研究課題として広く議論しなければいけない項目を述べる。

4.2.1 PMTUD、MTU ブラックホール問題

今回の実験では MTU ブラックホール問題はさまざま個所で発生した。主原因は Path MTU ディスカバリが運用上の理由やアドレス変換・カプセリングが挟まった段階で正常に動作しないことに起因している。一方で VPN などの多くのトンネル・カプセル化技術では PMTUD が正常に動作していることを前提に実装されているものが多い。

可能性のある解決方法としては、RFC 規約違反であるが、DF bit が設定されていてもルータや VPN ゲートウェイでフラグメントを行うように実装することが考えられる。また、MTU ブラックホールが発生している個所を効率よく特性する手法が現在確立していないこともあり、そのような検証ツールの研究開発も必要である。

4.2.2 RFC 4074 で指摘されている不適切な DNS 応答

3.3.4 節で説明したように、RFC 4074 [24] で指摘されている不適切な DNS 応答にどのように対応するかも研究課題の一つである。理想的には全ての DNS リゾルバ実装が RFC に則った実装に改修されることであるが、非常にコストが高い。

可能性のある解決方法としては、DNS64 のフォールバックの仕様を次のように変更することである。

- NXDOMAIN や ServFail が AAAA 応答で返されても A 要求を出すように DNS64 の仕様を変更する。

- DNS64 側でアクセスされる可能性のある A レコードをあらかじめ探索しキャッシュしておく。DNS 応答が NOERROR の場合、DNS64 リゾルバは AAAA 要求を発行する。仮に AAAA レコードが存在すれば DNS64 リゾルバはクライアントに対し AAAA レコードを返し、そうでなければキャッシュされている A レコードを返す。

4.2.3 プロトコル仕様上変換できないプロトコルに対するアドレス変換

IPSec や FTP など、いくつかのプロトコルは、その仕様上 IPv4/IPv6 変換が行えない。しかしながら、これらの問題はサーバ側が IPv6 対応すれば簡単に解決する問題なので、無理に変換する仕様を研究したり RFC で標準化する必要はない。

5 おわりに

ここでは、2011 年 9 月 WIDE 合宿にて実施した IPv6 only アクセス環境の検証実験に関して報告した。検証結果は AWFIT2011 [25] での発表論文やや IETF 82 TAIPEI [26] でのインターネットドラフトで文書としてまとめ発表し、Global IPv6 summit TAIPEI や Internet Week の BoF で口頭発表を行った。発表した結果、WIDE プロジェクト内外から追検証を行ってほしいという要望が多く上がったため、2012 年 3 月 WIDE 合宿で “Life with IPv6 ワークショップ” としてワークショップを開催し、より工学的な観点から 2011 年 9 月 WIDE 合宿で実施した実験の追検証を実施する運びとなった。2012 年 3 月 WIDE 合宿での “Life with IPv6 ワークショップ” における検証結果は来年度の WIDE 報告書や国内外の会議、標準化会議などで発表したい。

6 謝辞

WIDE プロジェクト 2011 年 9 月合宿に参加し実験にご協力頂いた全ての参加者、および実験環境の一部として StarBED[3]・JGN-X の環境から資源提供していただいた情報通信研究機構テストベッド研究開発推進センターおよび北陸 StarBED 技術センターに感謝の意をここに示す。

7 付録: ISC bind 9.8 と linuxnat64 を用いた DNS64 / NAT64 の設定例

ここでは、付録として、camp-net における ISC bind 9.8 と linuxnat64 を用いた DNS64 / NAT64 の設定例を記載する。英語による設定例は文献 [27] を参照していただきたい。

camp-net では Linux Debian squeeze (kernel 2.6.32) 64bit を用いて構築した。以下、Linux Debian squeeze (kernel 2.6.32) 64bit での設定である。

7.1 NAT64 implementation (linuxnat64) の入手

`http://sourceforge.jp/projects/sfnet_linuxnat64/` から図 7.1 に示すコマンドで入手できる。

7.2 linuxnat64 のコンパイル

図 7.2 に示すように、linux kernel source を apt-get コマンドで入手し、次に linuxnat64 の作業ディレクトリに移り make コマンドでコンパイルを実行する。

7.3 linuxnat46 の設定

まず IPv6 および IPv4 prefix を単一の物理インターフェース（例えば eth0）に設定する。camp-net ではロードバランスを行うため nat64 を 3 台用意した。ここでは、3 台のうちの一つを例として記載する。NAT64 の物理インターフェース（eth0）には以下のアドレスを割り当てた。

```
2001:200:0:ff81::37/64
203.178.158.37/26
```

次に DNS64 が A 応答で入手した IPv4 アドレスを IPv6 アドレスに格納してユーザに通知するために使われるアドレス変換用のプレフィックスを用意する。camp-net では次のプレフィックスををアドレス変換用プレフィックスとして用いた。

```
2001:200:0:ff99::/96
203.178.159.24/30
```

そして、図 7.3 に示すように linuxnat64 のカーネルモジュールを起動し、仮想インターフェース（nat64）に next hop の設定を設定する。

最後に、バックボーンルータ（ここでは cisco ルータ）に図 7.3 に示すような形式でアドレス変換プレフィックスを NAT64 へ向けるスタティックルートの設定を投

入する。この設定は rc.local 等に記載して再起動時に自動的に設定されるようにしておく。

7.4 ISC bind の入手とコンパイル

bind 9.8 以降のソースコードを入手し、bind のマニュアルに沿ってコンパイルする。apt から入手できる bind のバージョンが 9.8 以降であれば apt でバイナリを入手しても構わない。

7.5 bind の設定

まず、bind のマニュアルに沿って設定ファイルを作成する。次に、named.conf に次の設定を追加する。**dns64** の次に記入している IPv6 アドレスプレフィックスはアドレス変換用アドレスプレフィックスである。設定するネットワーク環境に合わせて変更して欲しい。

```
dns64 2001:200:0:ff99::/96 {

clients { any; };
mapped { any; };
suffix ::;
recursive-only yes;

};
```

そして、**named-checkconf** コマンドで named.conf の syntax error を確認する。syntax error がないことが確認できたら **rndc** によって bind の再起動を行う。再起動後、動作確認を行う。DNS64 をリゾルバとして設定し、適当な IPv4 アドレスのみ設定されたサーバの URL を名前解決した結果がアドレス変換プレフィックスの IPv6 アドレスで返されると成功である。上記の設定の場合、**2001:200:0:ff99::xxxx:yyyy** が返されることになる。名前解決によって返されたアドレスに対し ping6 で応答が返ってきた場合、NAT64 も動作している事が確認できる。

参考文献

[1] Y. Ueno. v6tun. `git://quina.sfc.wide.ad.jp/git/v6tun.git`.

[2] University of Tsukuba and SoftEther Corporation. UT-VPN. `http://utvpn.tsukuba.ac.jp/`.

[3] Internet Systems Consortium. DHCP. `http://www.isc.org/software/dhcp`.

[4] X. Li, C. Bao, and F. Baker. IP/ICMP Translation Algorithm. RFC 6145 (Proposed Standard), April 2011.

[5] M. Bagnulo, P. Matthews, and I. van Beijnum. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. RFC 6146 (Proposed Standard), April 2011.

[6] M. Bagnulo, A. Sullivan, P. Matthews, and I. van Beijnum. DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers. RFC 6147 (Proposed Standard), April 2011.

[7] Internet Systems Consortium. BIND. `http://www.isc.org/software/bind`.

[8] NLnet Labs. Unbound. `http://unbound.net/`.

[9] Viagénie. Ecdysis: open-source implementation of a NAT64 gateway. `http://ecdysis.viagenie.ca/index.html`.

[10] Geeknet, Inc. Linux NAT64 implementation. `http://linuxnat64.sourceforge.net/`.

[11] N. Lutchansky. TAYGA Simple, no-fuss NAT64 for Linux. `http://www.litech.org/tayga/`.

[12] N. Matsuhira. Motivation for developing Stateless Automatic IPv4 over IPv6 Tunneling (SA46T), Jul. 2011. individual draft `draft-matsuhira-sa46t-motivation-00.txt`.

[13] N. Matsuhira. Applicability of Stateless automatic IPv4 over IPv6 Tunneling (SA46T), Jul. 2011. individual draft draft-matsuhira-sa46t-applicability-02.txt.

[14] N. Matsuhira. Stateless Automatic IPv4 over IPv6 Tunneling: Global SA46T Address Format, Jul. 2011. individual draft draft-matsuhira-sa46t-gaddr-03.txt.

[15] N. Matsuhira. Stateless Automatic IPv4 over IPv6 Tunneling: Specification, Jul. 2011. individual draft draft-matsuhira-sa46t-spec-03.txt.

[16] C. Charnsripiny, P. Tantatsanawong, and T. Sribuddee. Research Network Infrastructure to Support Future Internet Technology in Thailand, Aug. 2011. Presentation in APAN 32nd meeting http://www.apan.net/meetings/India2011/Session/Slides/fit/3-2.pdf.

[17] Ed. R. Despres, S. Matsushima, T. Murakami, and O. Troan. IPv4 Residual Deployment across IPv6-Service networks (4rd) ISP-NAT's made optional, Mar. 2011. individual draft draft-despres-intarea-4rd-01.txt.

[18] masakazu's Weblog. Vyatta で 4rd 環境を構築しよう. http://bougaidenpa.org/masakazu/archives/176.

[19] M. Oe, H. Hazeyama, S. Yamamoto, and S. Shirahata. An implementation and verification of ieee 802.11 wireless network management system. *Electronics and Communications in Japan (Part I: Communications)*, Vol. 88, No. 12, pp. 20–28, June 2005.

[20] StarBED Project. http://www.starbed.org/.

[21] WIDE Cloud Working Group. WIDE Cloud Controller. http://wcc.wide.ad.jp/.

[22] WIDE Project. 2011年9月 WIDE 合宿 アンケート集計結果. https://member.wide.ad.jp/wide-confidential/camp/11autumn/enquete/result.cgi.

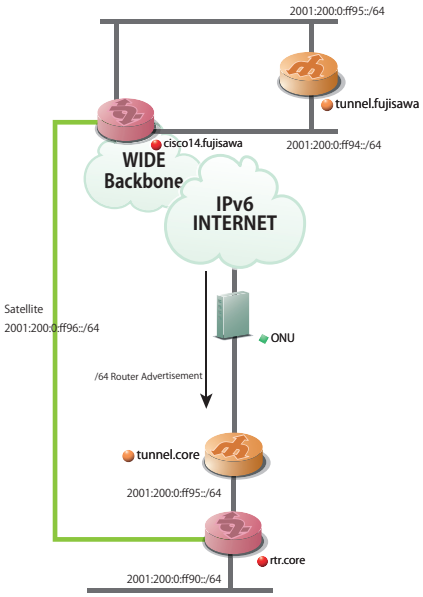
[23] J. Arkko and A. Keranen. Experiences from an IPv6-Only Network, Apr. 2011. individual draft draft-arkko-ipv6-only-experience-03.txt.

[24] Y. Morishita and T. Jinmei. Common Misbehavior Against DNS Queries for IPv6 Addresses. RFC 4074 (Informational), May 2005.

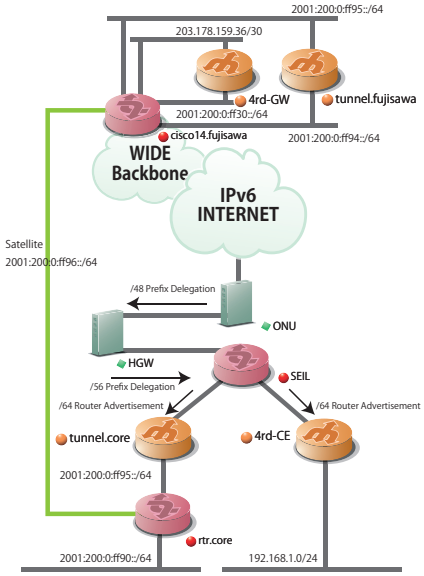
[25] H. Hazeyama, Y. Ueno, H. Sato, Y. Yamagishi, T. Yokoishi, and H. Ishibashi. How much can we survive on an IPv6 network? - Experience on the IPv6 only connectivity with NAT64/DNS64 at WIDE camp 2011 autumn. In *Proceedings of Asia Workshop on Future Internet Technologies (AWFIT2011)*, November 2011.

[26] H. Hazeyama and Y. Ueno. *Experiences from an IPv6-Only Network in the WIDE Camp Autumn 2011*, November 2011. http://tools.ietf.org/html/draft-hazeyama-widecamp-ipv6-only-experience-00.

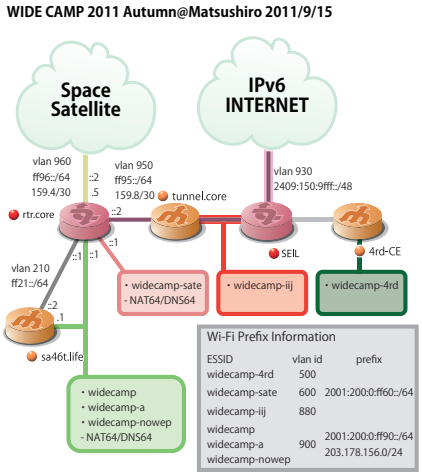
[27] Y. Ueno and H. Hazeyama. How to setup NAT64 and DNS64 on ubuntu. wide memo https://member.wide.ad.jp/wide-confidential/memo/wide-memo-camp1109-hack-eden-nat64-dns64-settings-00.txt.



(a) 合宿ネットワークバックボーン構成（9 月 6 日 午後 8 時まで）

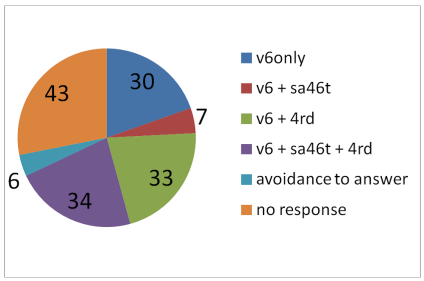


(b) 合宿ネットワークバックボーン構成（9 月 6 日 午後 8 時以降）

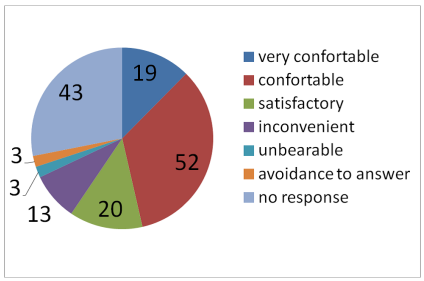


(c) 合宿ネットワークアクセス周りの構成

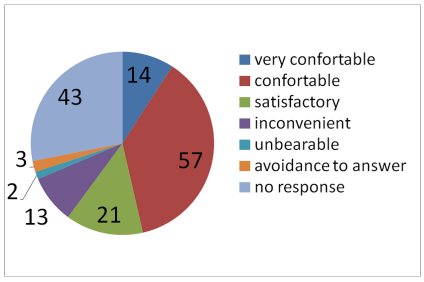
図 1: 合宿ネットワークの構成



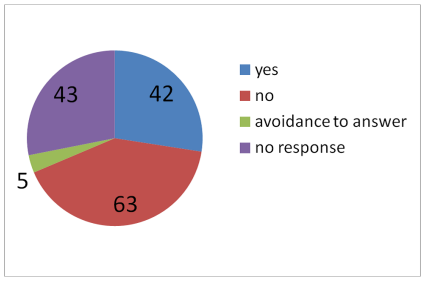
(a) 参加者が利用したと申告したユーザアクセスの割合



(b) Q1: 今回の合宿では主に IPv6 のみを提供しましたがいかがでしたか？

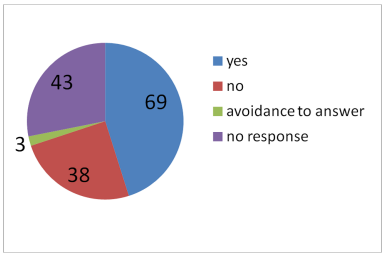


(c) Q2: 今回のネットワークの品質はいかがでしたか？

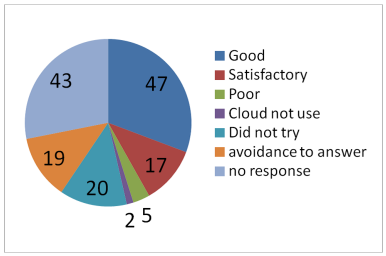


(d) Q3: SSID:widecamp において IPv4 の認証サービスを使いましたか?

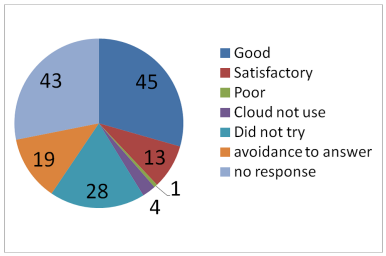
図 2: NOC による実験のアンケート結果



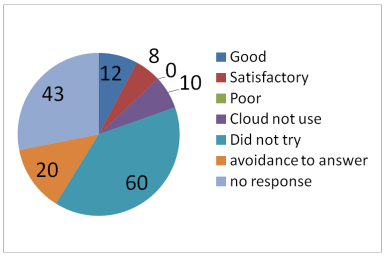
(a) Q4: SSID: widecamp-4rd を使いましたか？



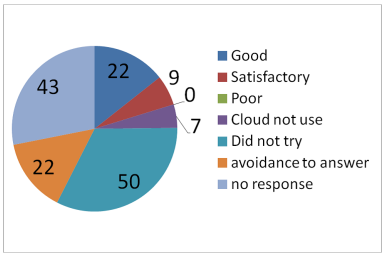
(b) Q5: widecamp-4rd の通信環境上での通信はいかがだったでしょうか? (Web)



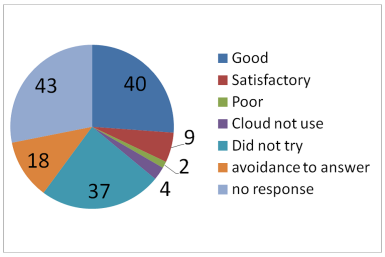
(c) Q6: widecamp-4rd の通信環境上での通信はいかがだったでしょうか? (Mail)



(d) Q7: widecamp-4rd の通信環境上での通信はいかがだったでしょうか? (VPN)



(e) Q8: widecamp-4rd の通信環境上での通信はいかがだったでしょうか? (Messenger)



(f) Q9: widecamp-4rd の通信環境上での通信はいかがだったでしょうか? (SSH)

図 3: 4RD 検証実験のアンケート結果

```
#git clone git://linuxnat64.git.sourceforge.net/gitroot/linuxnat64/linuxnat64
```

図 4: git による linuxnat64 の入手

```
#apt-get install linux-headers-'uname -r'

#cd <your_working_dir>/linuxnat64/modules/

#make
```

図 5: linuxnat64 のコンパイル

```
# insmod /path/to/nat64.ko ipv4_address=203.178.159.25 \
  prefix_address=2001:200:0:ff99::
# ip link set nat64 up
# ip -6 route add 2001:200:0:ff99::/96 dev nat64
# ip route add 203.178.159.24/30 dev nat64
```

図 6: linuxnat64 の起動と設定

```
ipv6 route 2001:200:0:FF99::/96 2001:200:0:FF81::37
ip route 203.178.159.24 255.255.255.252 203.178.158.37
```

図 7: cisco ルータでの NAT64 へのスタティック経路の設定