Network Diagrams of WIDE Backbone

櫨山寛章 (hiroa-ha@is.naist.jp)
堀場勝広 (qoo@sfc.wide.ad.jp)
上野幸杜 (eden@sfc.wide.ad.jp)
岡本裕子 (okayu@wide.ad.jp)
岡村 耕二 (oka@ec.kyushu-u.ac.jp)
竹内奏吾 (sohgo@wide.ad.jp)
井上博之 (hinoue@hiroshima-cu.ac.jp)
宇多 仁 (zin@jaist.ac.jp)
明石邦夫 (k_akashi@jaist.ac.jp)
小林和真 (kazu-k@cs.kusa.ac.jp)
岡本慶大 (yoshihiro-o@is.naist.jp)
寺本泰大 (teramoto@net.ist.i.kyoto-u.ac.jp)
Glenn Mansfield Keeni (glenn@cysols.com)
齋藤武夫 (saito@cysols.com)　土井一夫 (kazuo@cysols.com)
高橋航平 (flast@tsukuba.wide.ad.jp)
畠山元也 (genyakun@tsukuba.wide.ad.jp)
厚谷 有輝 (atie@inl.ics.keio.ac.jp)
安藤 大佑 (mackey@inl.ics.keio.ac.jp)
井出 幹 (shallot@inl.ics.keio.ac.jp)
関口 貴久 (arc@inl.ics.keio.ac.jp)
近藤 賢郎 (latte@inl.ics.keio.ac.jp)
宮下 山斗 (mine@inl.ics.keio.ac.jp)
田原 裕市郎 (ash@sfc.wide.ad.jp)
山本 成一 (yama@wide.ad.jp)
関谷 勇司 (sekiya@wide.ad.jp)
長谷部 克幸 (hasebe@wide.ad.jp)

中村 遼 (upa@wide.ad.jp)
石原 知洋 (sho@c.u-tokyo.ac.jp)
藤原 和典 (fujiwara@wide.ad.jp)
遠藤 正仁 (masaxmasa@wide.ad.jp)

平成 24 年 12 月 31 日

本ドキュメントでは、WIDE backbone と各 NOC の現状について述べる。

1　はじめに

WIDE バックボーンネットワークは国内はもとより San Fransico, Losangels, Bangkok など海外にも拠点（NOC, Network Operation Center）を持つ広大なレイヤー 2 およびレイヤー 3 ネットワークである．WIDE バックボーンネットワークは各接続組織の対外接続ネットワークとして活用されるだけではなく，インターネットの新技術を開発している研究者，開発者らの新技術の運用実験の場としても頻繁に活用されている．

WIDE バックボーンネットワークの運用は Two ワーキンググループに参加する各 NOC の運用者による定常的な運用に支えられている．本年度の Two ワーキンググループの活動報告として，WIDE バックボーンネットワークの運用報告を行い，合わせて，Two サーバの WIDE クラウドへの移設作業や，DNSSEC の運用実験に関しても報告する．最後に今後の WIDE バックボーン運用についての展望を述べる．

2　WIDE バックボーンの運用

本節では，WIDE バックボーンの各拠点での 2011 年 12 月 31 日から 2012 年 12 月 31 日までの運用報告と 2012 年 12 月 31 日現在の WIDE バックボーンのネットワーク構成を報告する．図 1 は 2012 年 12 月 31 日現在の WIDE バックボーンの概略図である．

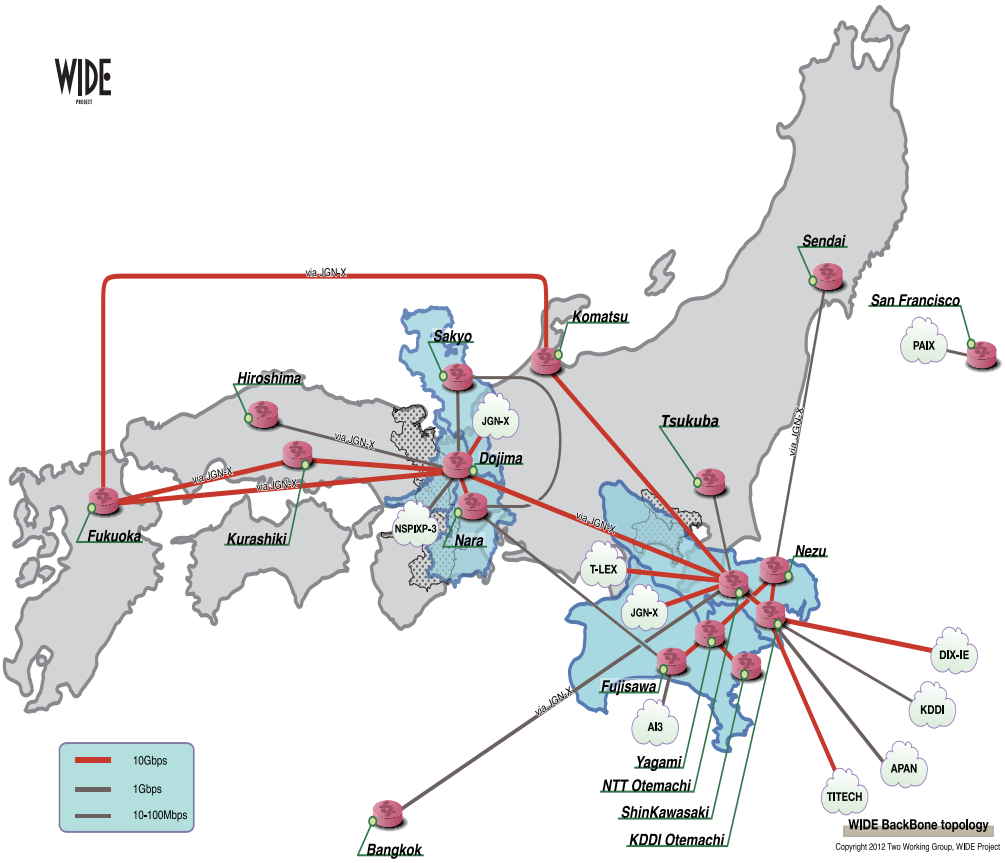


図 1: WIDE バックボーントポロジ

2.1 San Francisco

サンフランシスコ NOC(sanfrancisco) は，2004 年 4 月からそれまでの sanjose に代わり稼働した新しい NOC で，Los Angeles から OC-3 により接続されていた．その後 OC-3 から 100M Ethernet に変更された．主な接続先は，PAIX や ISC である．

2010 年 9 月の Los Angeles NOC 撤収にともない，2010 年 10 月に Los Angeles と San Francisco 間の回線も廃止され，専用線による接続の無い独立 NOC として存在する．

2011 年ならびに 2012 年は特に構成変更は無く，M-ROOT に関する機材やクラウド WG のための実験機材が存在する．

- (2012/07/27) pc5.sfo ディスク障害により HDD 交換
- (2011/10/27) pc5.sfo RAID コントローラ故障

As of 2012/12/18

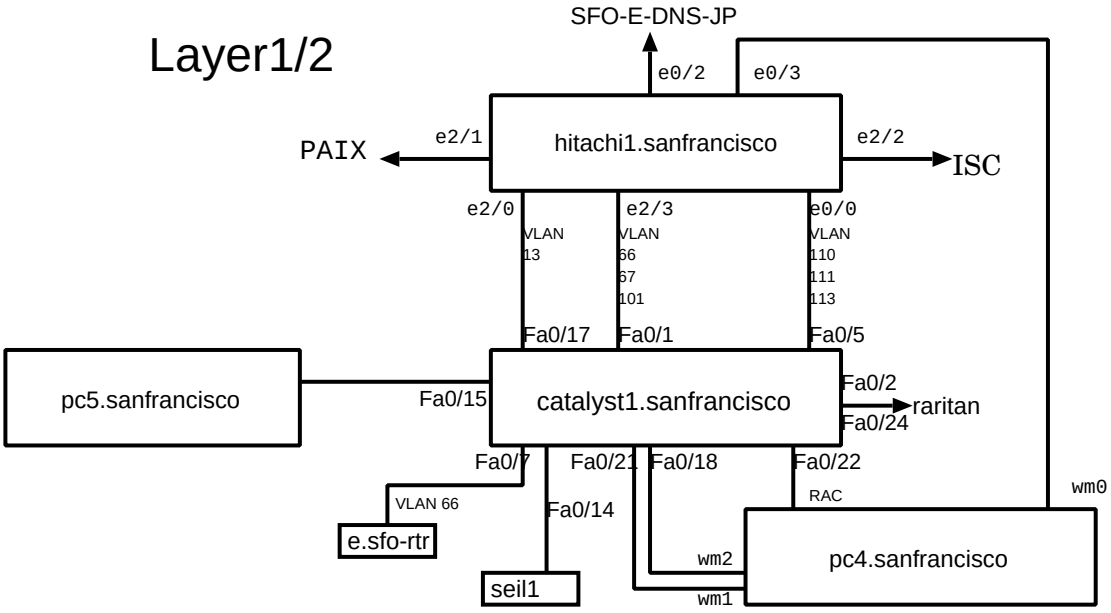


図 2: San Francisco NOC

2.2 仙台

仙台 NOC は仙台周辺の拠点を収容する NOC として運用されている．接続回線の障害，および停電の影響による他は安定して運用された．トポロジー，および構成機器に変更はなかった．

- (2012/08/02–) JGN-X アクセスポイント東北-2 との接続回線（広域イーサネット専用回線）障害による断続的接続断
- (2012/08/07) 専用回線側設備の故障復旧により接続回復
- (2012/12/01) 法定電気設備点検による計画停電

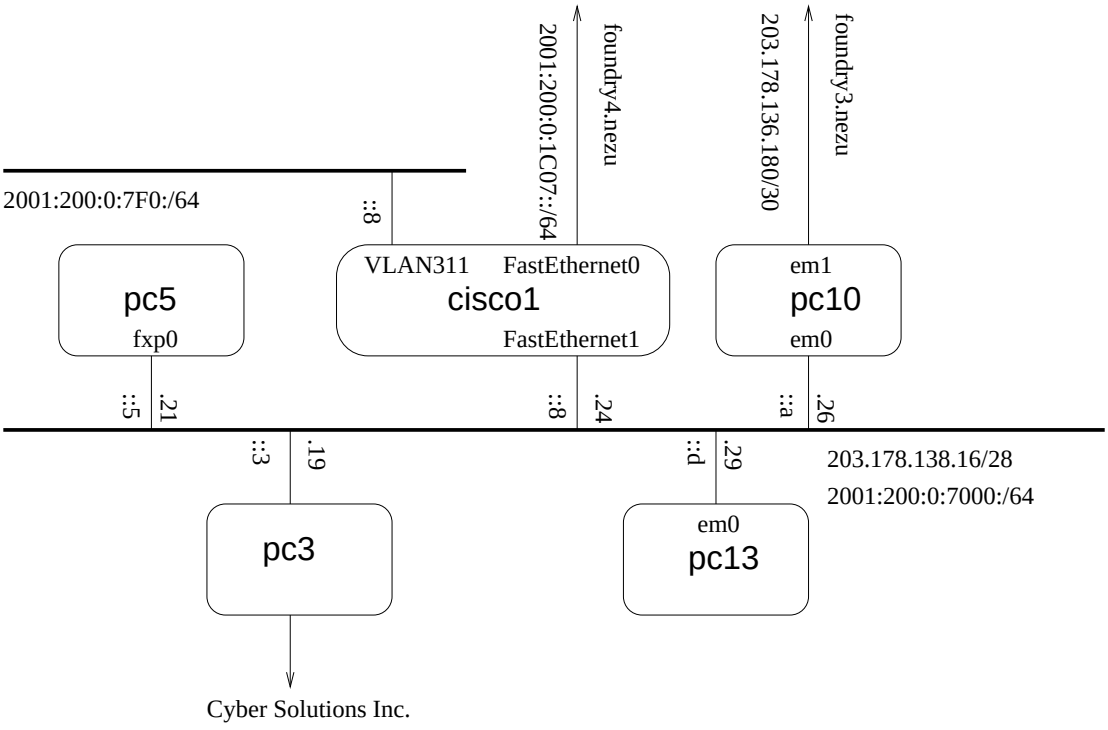


図 3: 仙台 NOC

2.3 筑波

筑波 NOC は，2009 年 3 月に筑波大学学術情報メディアセンター内に新たに設置された NOC で，システム情報工学研究科産学間連携推進室をはじめとする周辺の研究組織を収容している．

- (2012/05/19) FTP サーバのハードウェア交換
- (2012/05/20) 同上
- (2012/07/07) ネットワーク構成更新
- (2012/07/08) 同上
- (2012/07/08) グローバル・固定 IPv6 アドレス割当型トンネル接続実験サービスに於いて DNS64/NAT64 サービスの広告開始
- (2012/10/27) 電気事業法に基づく電気設備の定期点検のため停止
- (2012/10/28) 同上
- (2012/12/03) 学術情報メディアセンター分電盤検査

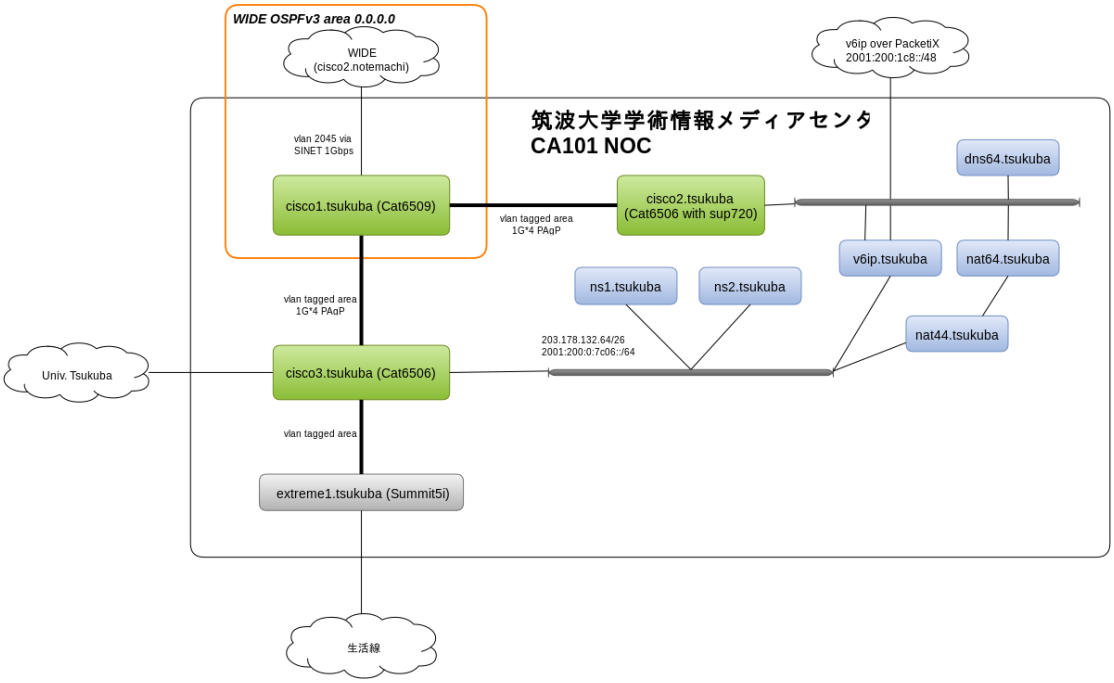


図 4: 筑波 NOC

2.4 根津

根津 NOC は，WIDE 関東地区の重要な接続拠点として，東京大学，JGN-X 等との接続を行っている．また WIDE クラウドの拠点としても重要な機器が設置されている．

- (2012/12/20) nezu-kote 線回線借用
- (2012/12/20) nezu-yagami 線回線借用

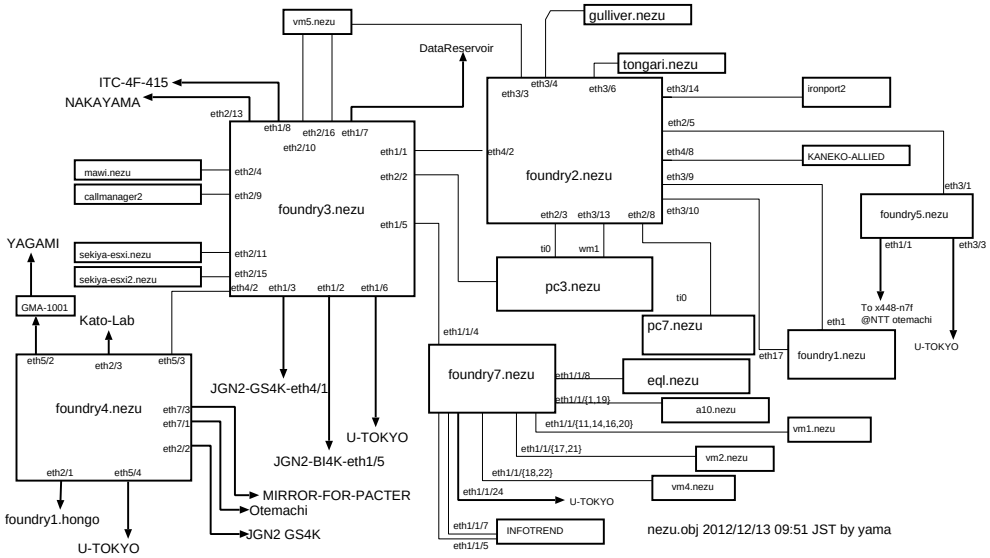


図 5: 根津 NOC

2.5 NTT 大手町

NTT 大手町 NOC(notemachi) は，1999 年終りから稼働した比較的新しい NOC で，現在，関西方面，北陸方面への L2 網，JGN-X ，APAN-JP の接続拠点として重要な立場にある．また，日本のインターネットトラフィック交換の 1 拠点として，DIX-IE ， T-LEX を設置し ISP および学術研究 NW を収容している．

- (2012/03) 東阪間接続変更
- (2012/06) Interop 2012 Tokyo と接続
- (2012/11) ORF2012 (慶大 SFC Open Research Forum 2012) 用接続
- (2012/11) 東京大学 DR チーム，sc12 でのデータ並列転送実験他を実施/JGN-X 日米回線
- (2012/11) tokyo6to4 接続解除

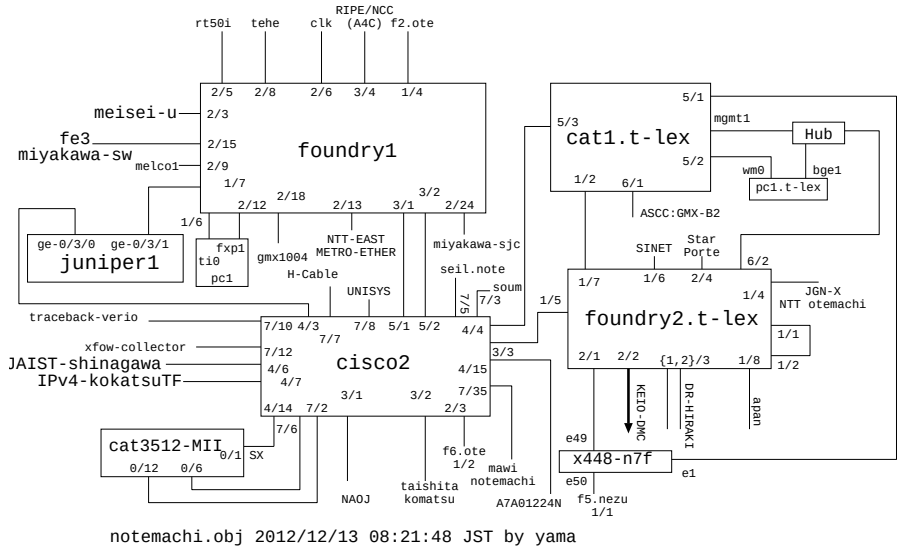


図 6: NTT 大手町 NOC

2.6 KDDI 大手町

KDDI 大手町 NOC は WIDE バックボーンの中でも中核を担う重要な NOC となっており，外部組織接続が最も多い NOC となっている．10GbE によるバックボーンが導入され，NTT 大手町 NOC との連携がより強まり，WIDE から DIX-IE への接続拠点となっている．

- (2012/03) alaxala1.otemach firmware update
- (2012/04) foundry6.otemachi 設定変更のため再起動
- (2012/06) juniper1.otemachi 古くなったため機能を他機器に移して撤去
- (2012/11) Tokyo6to4 の kotemachi ノードを撤去
- (2012/12) kote-nezu 線回線借用

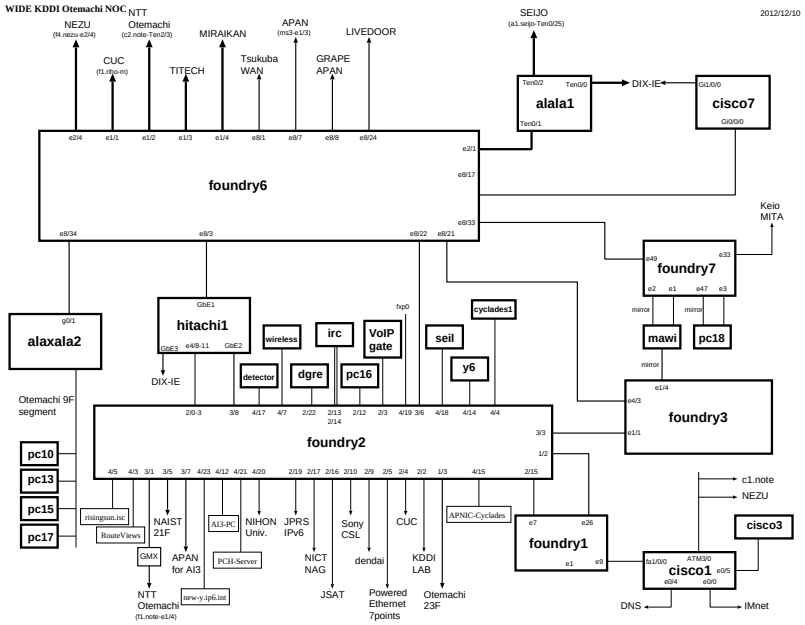


図 7: KDDI 大手町 NOC

2.7 八王子

八王子 NOC は本年度をもって廃止となった．本節では，八王子 NOC の経緯と本年度の活動について記述する．

2.7.1 八王子 NOC の経緯

八王子 NOC は，八王子周辺の組織を収容するために 1994 年 7 月に東京工科大学内に設置され，東京 NOC と 192kbps で接続した．八王子 NOC 設置から本年度までに，東京工科大学，拓殖大学，明星大学，職業能力開発大学校 (現：職業訓練開発総合大学校)，東京工業高等専門学校，津田塾大学，東京薬科大学，東京造形大学が八王子 NOC に接続した．時の経過とともに，八王子 NOC に接続していた組織は減少していった．WIDE プロジェクトから脱退したり，インターネットへの接続手段が大きく変化するにつれ，Point-to-Point で八王子 NOC と接続する形態から広域イーサネットサービスを用いた形態へと変化するにつれ，さらに減少した．ここ数年は東京工科大学のみであった．本年度，東京工科大学と WIDE プロジェクトとの接続が終了したため，八王子 NOC は廃止となった．

八王子 NOC では，NetNews サーバ，DNS サーバ，メールサーバ，Web キャッシュサーバ，IPv6 ルータ，マルチキャストルータ等の運用を行った．八王子 NOC の初期は，各組織および上流への接続が狭帯域であり，NetNews サーバ，Web キャッシュサーバが有効な役割を果たした．DNS サーバは，各組織のセカンダリ DNS サーバとして運用した．

2.7.2 2012 年度の活動

- (2012/06/11) 東京工科大学と WIDE 間の BGP 接続を切断
- (2012/07/22) 全サーバマシンを停止
- (2012/07/23) 八王子 NOC と WIDE 大手町 NOC 間回線終端装置を撤去
  サーバマシン・ルータをサーバ室から撤去

2.8 矢上

矢上 NOC は慶應義塾大学理工学部矢上キャンパス構内にあり，同大学理工学部情報工学科および周辺の研究組織を収容すると共に慶應 DMC を介して JGN-X, CineGrid との接続を行っている．現在，WIDE-Cloud の矢上 NOC セグメントの導入を進めており，導入完了次第矢上 NOC 内各種サーバの仮想化を行う予定である．

- (2012/08/17) 定期保安点検による停電

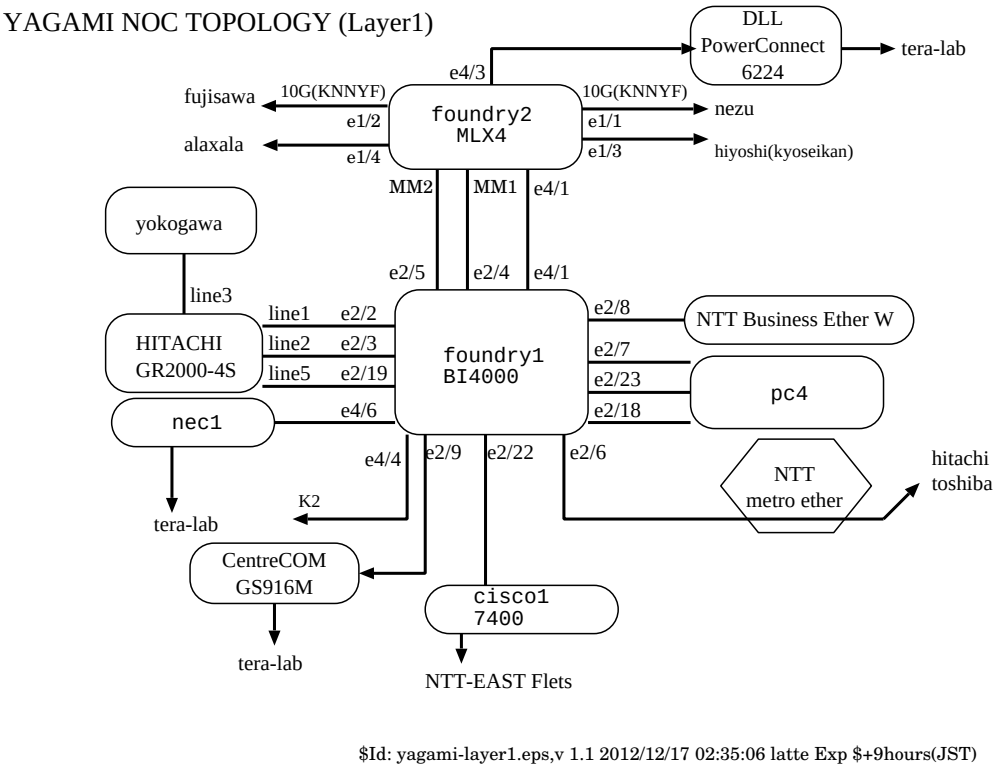


図 8: 矢上 NOC Layer-1 トポロジ

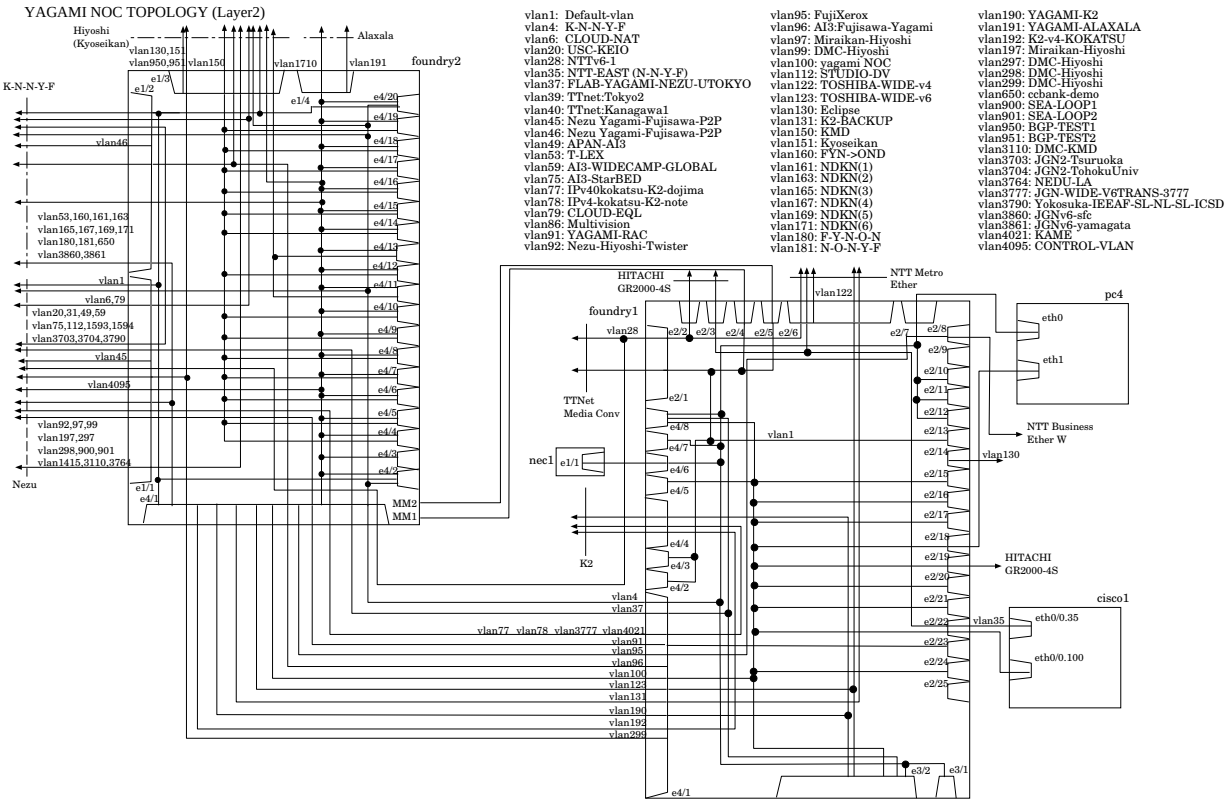


図 9: 矢上 NOC Layer-2 トポロジ

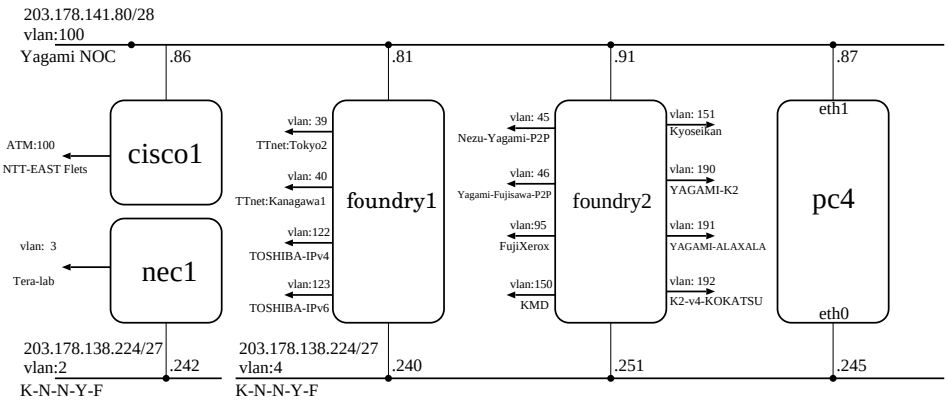


図 10: 矢上 NOC Layer-3 トポロジ

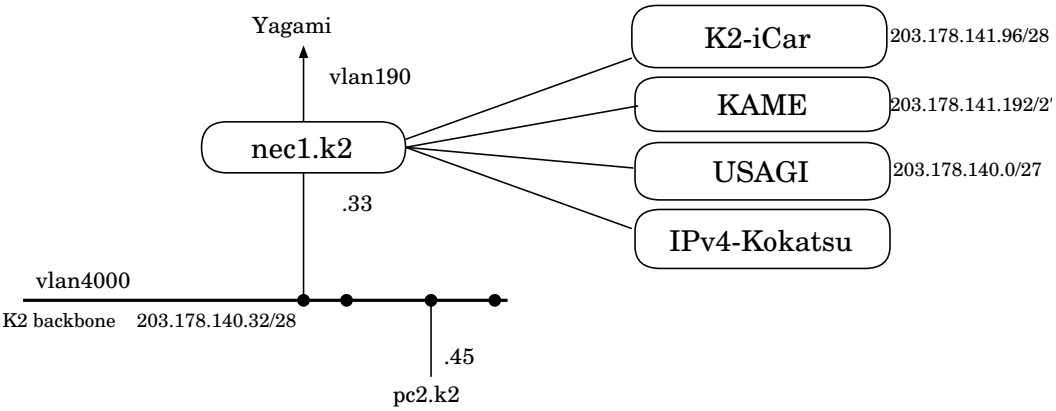
2.9 新川崎

新川崎 NOC は，K2 タウンキャンパス内の村井研究室を拠点とした NOC である．K2 タウンキャンパス村井研究室はこれまで矢上 NOC の下部組織として運用されてきたが，リーフ組織への回線提供を行うため，2005 年後半より NOC として運用していた．2008 年度の構成変更にて，リーフ組織であったアラクサラの接続先が矢上 NOC へと変更となったため，一時 NOC ではなくなったが，2009 年度に IPv4 枯渇 TF がリーフ組織として接続したため，再び NOC としての機能を担うこととなった．

- (2012/02/15) 新川崎 K2 タウンキャンパス付近共同溝工事によるファイバ借用発生
- (2012/02/16) 新川崎 K2 タウンキャンパス付近共同溝工事によるファイバ借用発生

As of 2012/12/28

Shinkawasaki NOC topology Map(Layer3)



Shinkawasaki NOC topology Map(Layer2)

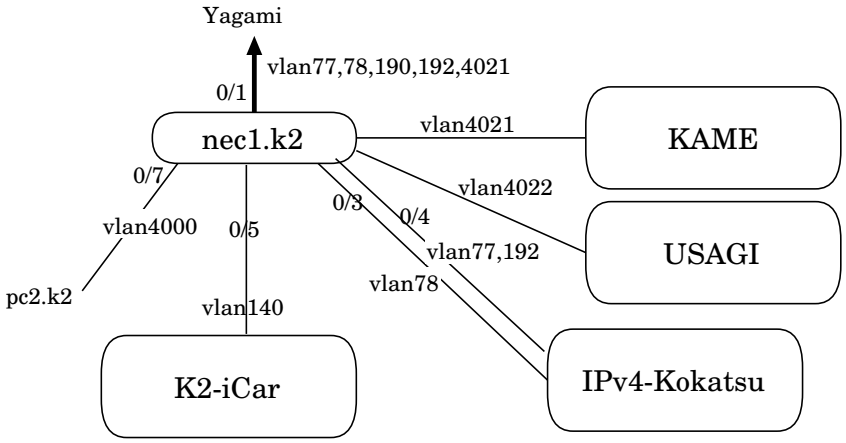
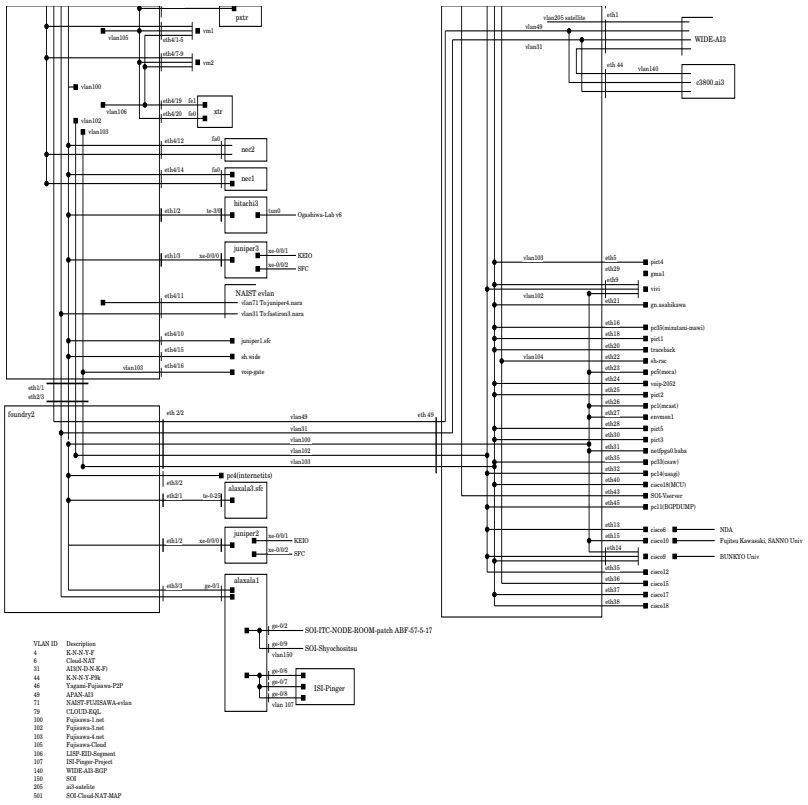


図 11: 新川崎 NOC

2.10 藤沢

藤沢 NOC は慶應義塾大学湘南藤沢キャンパス内にあり，慶應義塾大学や村井研究室の他，周辺の研究組織を収容している．同時に XCAST や AI3 との接続，VoIP 関連サービス (CallManager, VoiceGateway) などを行っている．

- (2012/03/01) Powered Ether 回線撤去工事
- (2012/03/26) 奈良・左京・藤沢間専用線サービス切り替え工事
- (2012/08/22-28) `www.wide.ad.jp` 映像配信サーバ移行工事
- (2012/10/01) `nec2.fujisawa` 設置
- (2012/12/02) SFC 構内全域の変電設備の定期保安点検による構内停電



VLAN ID	Description
4	K-N-N-Y-F
6	Cloud-NAT
31	AHN-B-N-K-F)
44	K-N-N-Y-F9h
46	Yagami-Fujisawa-P2P
49	APAN-AI3
71	NAIST-FUJISAWA-evlan
79	CLOUD-EQL
100	Fujisawa-1.net
102	Fujisawa-3.net
103	Fujisawa-4.net
105	Fujisawa-Cloud
106	LISP-EID-Segment
107	ISI-Pinger-Project
140	WIDE-AI3-BGP
150	SOI
205	ai3-satellite
501	SOI-Cloud-NAT-MAP

図 12: 藤沢 NOC Layer-2 トポロジ図

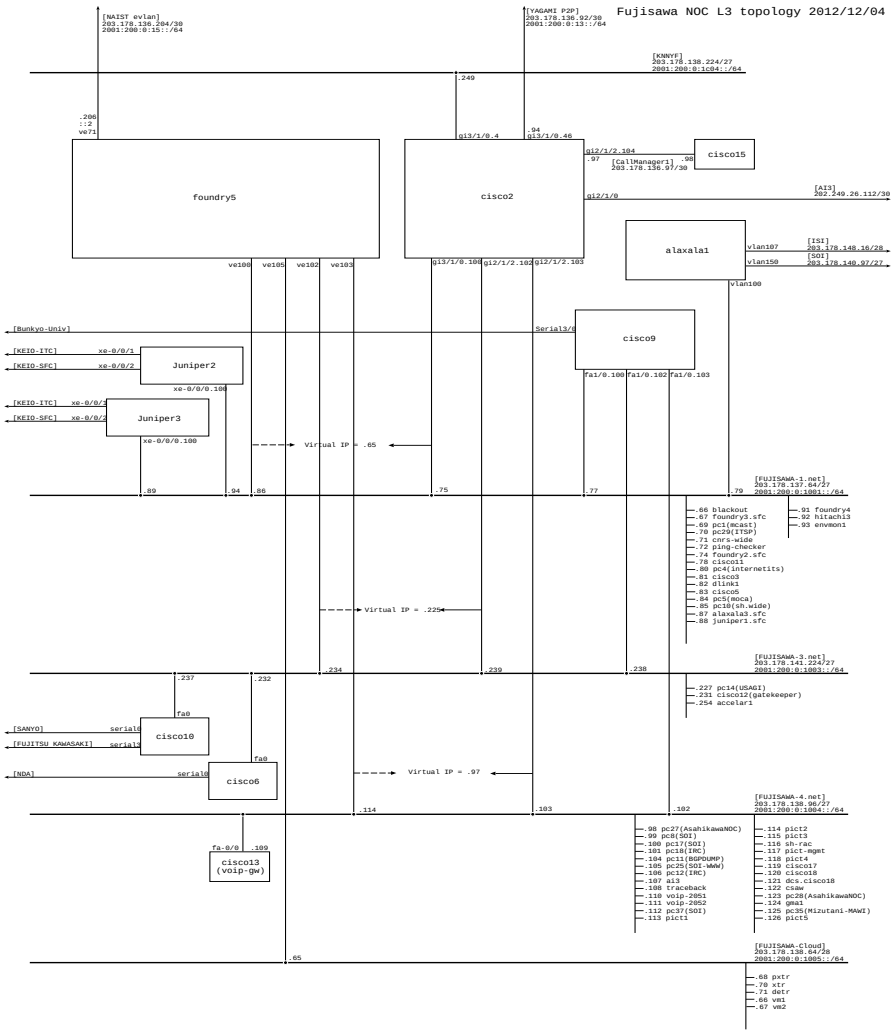


図 13: 藤沢 NOC Layer-3 トポロジ図

2.11 小松

小松 NOC は北陸先端科学技術大学院大学 (JAIST / 石川県能美市) 内に設置された NOC であり，同大学，NICT 北陸 StarBED 技術センター (通称: StarBED) 等への接続を収容している．NOC 間接続として関東および関西方面に対し複数のリンクを持ち，東阪間リンク障害時の迂回経路としての役割も担っている．

- (2012/03/17) 08:00–17:00 JAIST 全学停電に伴うサービス停止.

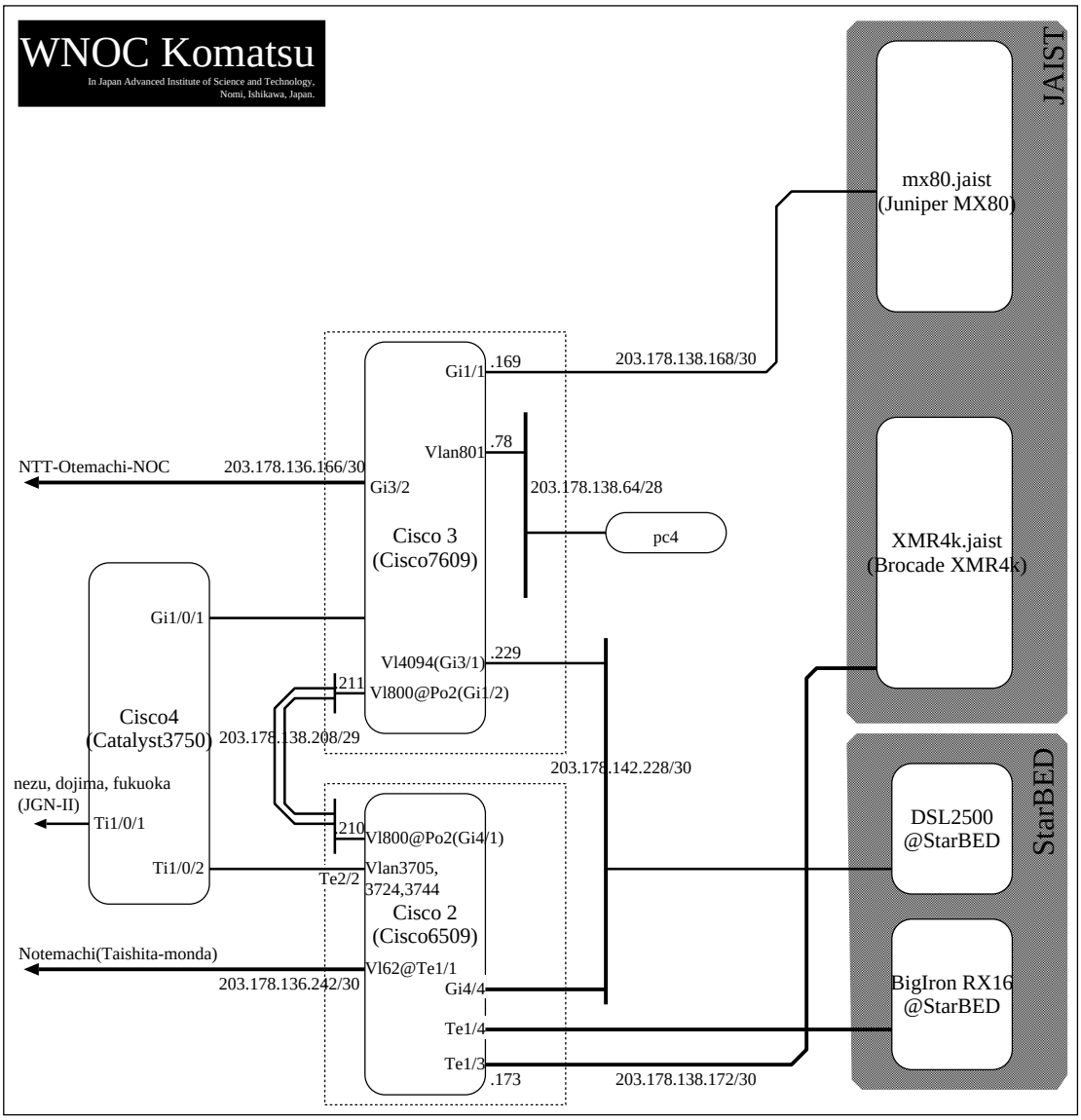
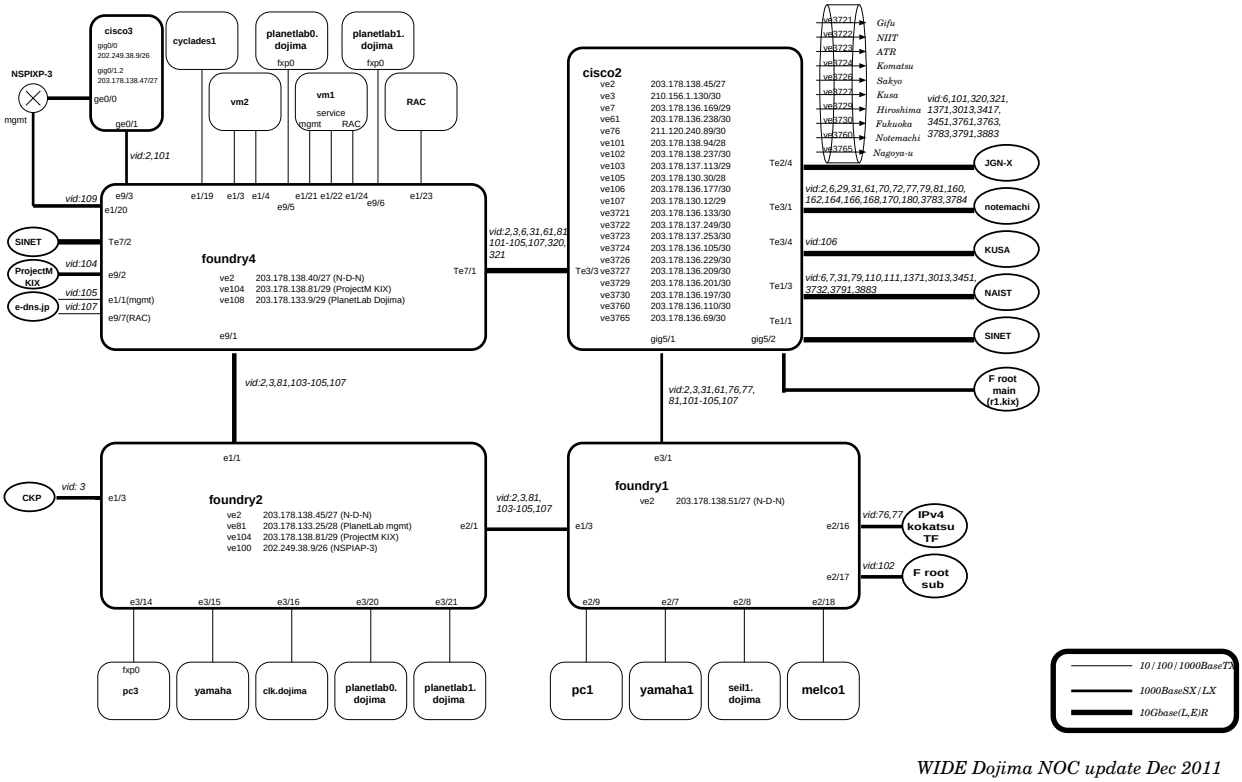


図 14: 小松 NOC

2.12 堂島

堂島 NOC は，WIDE プロジェクトのネットワークにおける西日本のコア拠点となっている．NTT テレパーク堂島第 1 ビルと第 3 ビルに拠点を構え，NTT 大手町 NOC とともに 10GigabitEthernet バックボーンの 1 点を担ったり，大阪における学術 IX(NSPIXP3) 拠点を担ったりしている NOC である．また，第 3 ビル内において JGN や SINET とも接続し，西日本方面の多数の NOC とリーフサイトを収容している．

- (2012/3) DIX-IE - NSPIXP3 東阪接続に伴う構成変更作業
- (2012/8) `pc1.dojima` 故障に伴う機器交換
- (2012/10) `vm4.dojima` 設置
- (2012/11) `cisco2.dojima` モジュールトラブル (抜去により回避)
- (2012/12) IPv4 アドレス枯渇対応タスクフォース機器撤去



WIDE Dojima NOC update Dec 2011

図 15: 堂島 NOC トポロジ

2.13 奈良

奈良 NOC は奈良先端科学技術大学院大学内にあり，大学および NOC 周辺の研究組織を収容するとともに AIII と接続している．また，Debian JP 等の公式ミラーを始めとする 10 以上のミラーを提供する FTP ミラー (`ftp.nara.wide.ad.jp`) をサービスしている．

- (2012/3/29) 奈良・左京・藤沢線を NTTcom e-VLAN から，K-Opticom イーサネット VPN に切り替え
- (2012/5/29) 落雷に伴い，15 分程度の停電が発生．WIDE クラウドノードの一部がダウン．
- (2012/8/14) 落雷に伴い，10 分程度の停電が発生．
- (2012/9/21, 25) 奈良-堂島線回線借用 (ファイバルート変更) による 3 分程度の回線断
- (2012/10/28) NAIST 全学停電による NOC 機能停止．旧 two サーバ twomine を撤去

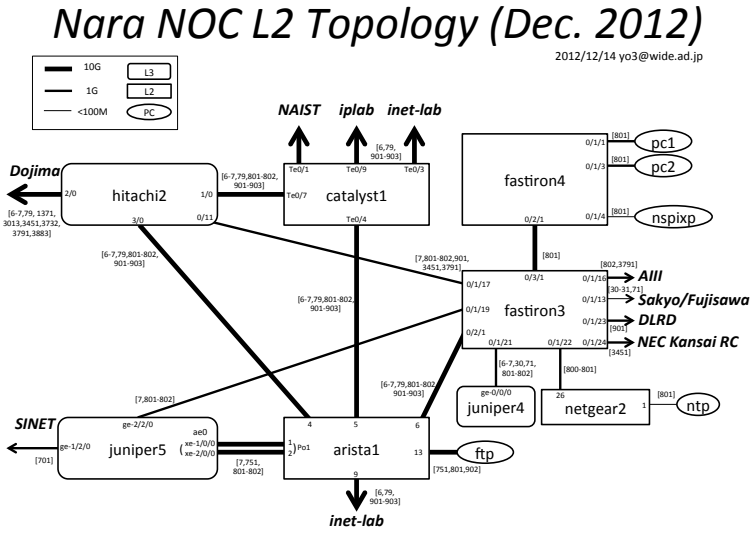


図 16: 奈良 NOC Layer-2 トポロジ

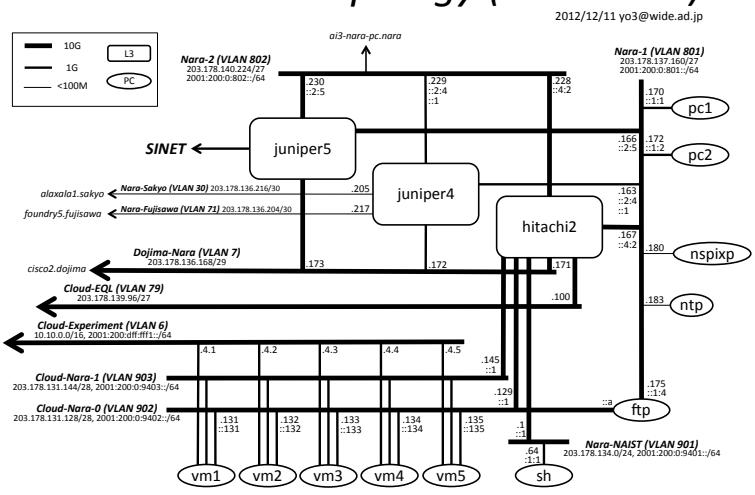


図 17: 奈良 NOC Layer-3 トポロジ

2.14 左京

左京 NOC は京都およびその周辺に存在する組織に対する接続拠点であり京都大学に設置されている．また，遠隔講義実施のためにキャンパスプラザ京都および広島市立大学向けの IPv6 接続も提供している．予定されていた NOC 設備の学術メディアセンター南館から電話庁舎への移行を 2011 年 12 月に行った．また京都大学との BGP 接続を 2012 年 3 月に開始した．

- (2011/12/20) L3 装置 (IP8800/S3630) を学術情報メディアセンター南館から電話庁舎に移設
- (2012/3/7) 京都大学との BGP 開始

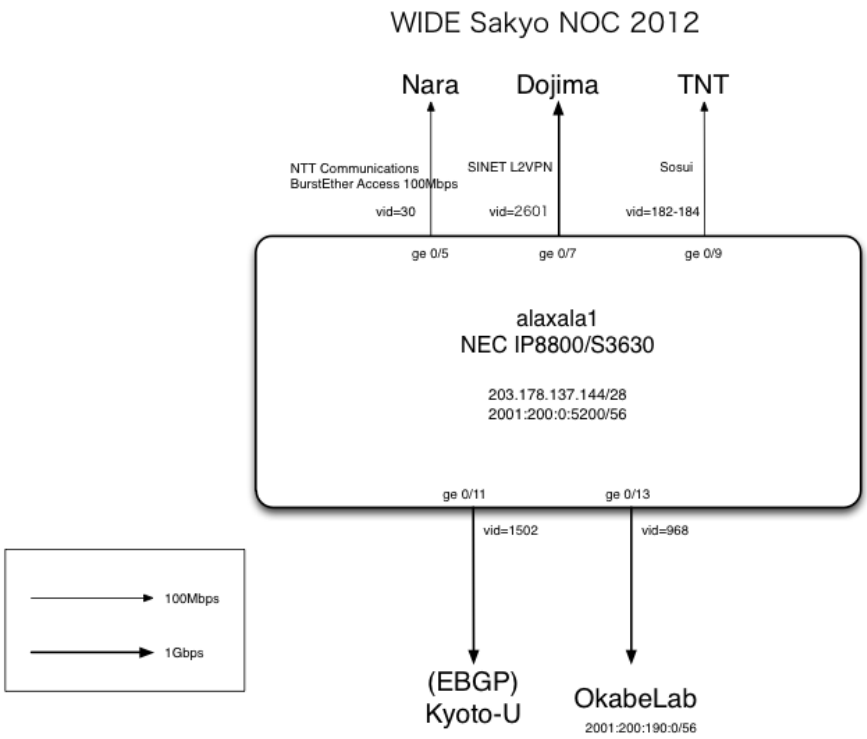


図 18: 左京 NOC

2.15 倉敷

倉敷 NOC は，平成 24 年度に学内ネットワーク機器の更新にともない NOC 機器の更新など全体構成の変更を実施した．外部接続回線を収容していた GS4K は ASR9K に更新し，対外的な L3 ルータのうち GR4K を Juniper MX80 に更新した．基本的な接続設定は，従前の装置の設定を踏襲している．また，倉敷 NOC 機器については，死活監視ツールで監視を行っているが特に大きな問題もなく運用されている．

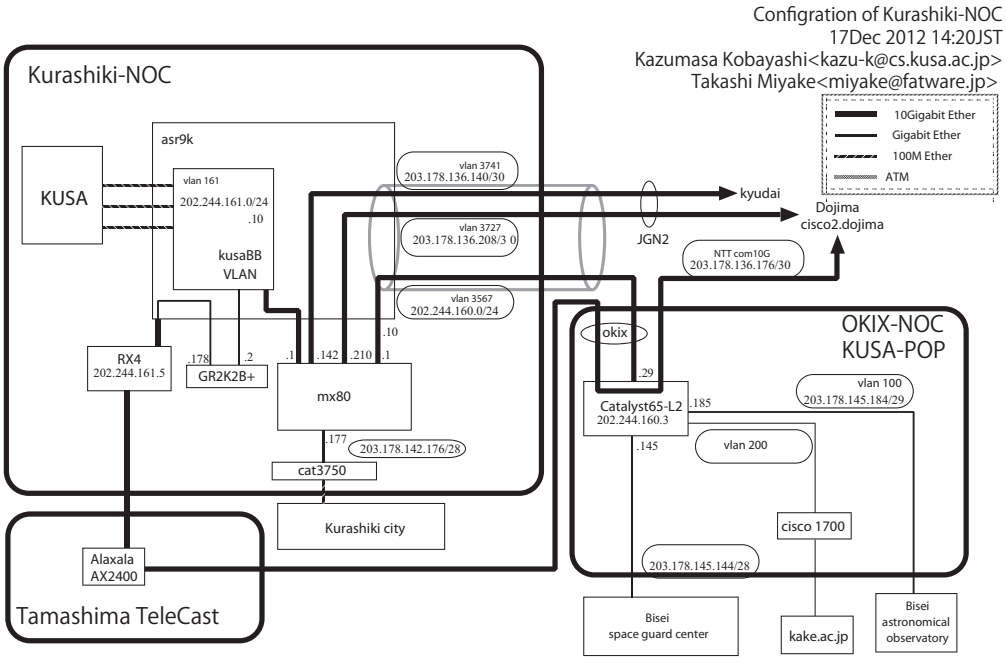


図 19: 倉敷 NOC

2.16 広島

広島 NOC は，2011 年 3 月末日の JGN2plus から JGN-X への移行に伴い接続性が失われていたが，広島大学（東広島市）から広島市立大学（広島市）に NOC の設置場所や管理担当者を変更し，2012 年 4 月から利用可能になっている．

移設に伴い，老朽化したルータは利用を停止し，XenServer 上で動作する VM(Virtual Machine) を使ったソフトウェアルータを導入した．ローカルな Linux のサーバも同様に VM 上で実現している．ソフトウェアルータとしては Vyatta Router VC6.3 を使用し，OSPF および OSPFv3 の設定を行っている．コンフィグレーションについては，Cisco の文法と似ていることや TAB による補完も行われることから，Vyatta を使った初めての設定にも関わらず大きく戸惑うことはなかった．また，ソフトウェアルータを使用することにより危惧された，安定性やパフォーマンスについても特に問題なかった．

なお，福岡 NOC が停止したままであることから，他サイトとの接続が冗長化されていないという問題がある．福岡 NOC の今後の見通しによっては，他のサイトとの接続を早急に検討し，構成変更を行う予定である．

- (2012/04/10) NOC 構築．JGN-X のパス変更
- (2012/04/20) NOC 構築．IPv4 による接続
- (2012/05/27) IPv6 による接続．NOC 移設完了
- (2012/09/02) 法令点検による計画停電

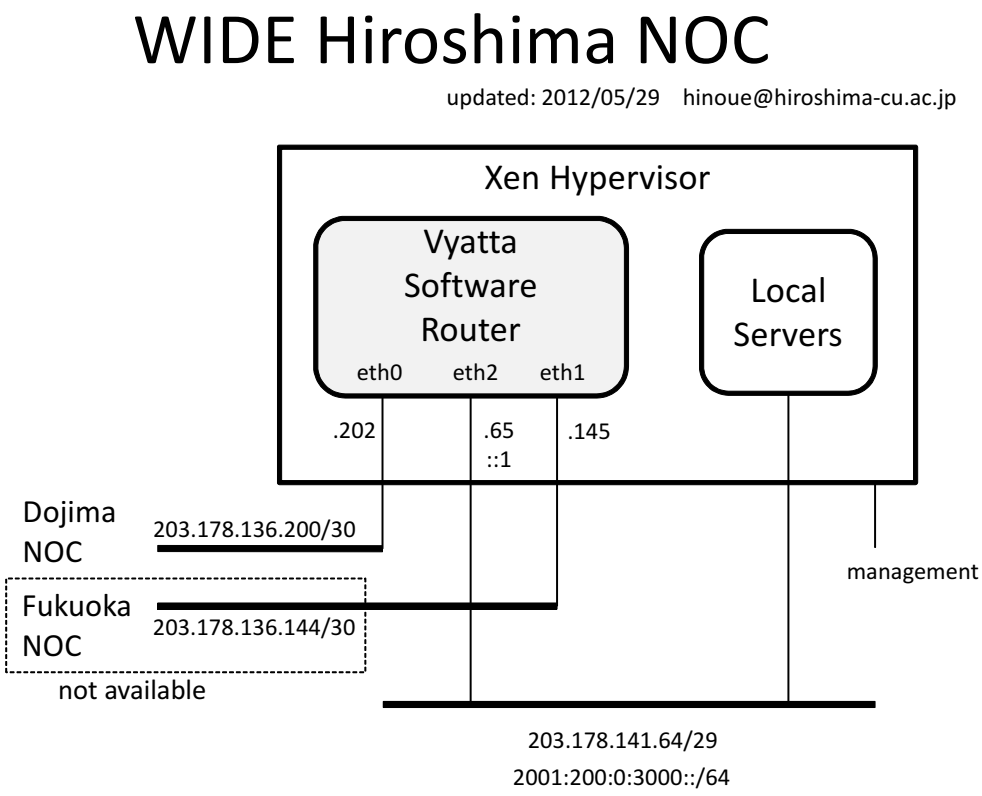

WIDE Hiroshima NOC
updated: 2012/05/29 hinoue@hiroshima-cu.ac.jp
Xen Hypervisor
Vyatta
Software
Router
eth0
eth2
eth1
Local
Servers
.202
.65
::1
.145
Dojima
NOC
203.178.136.200/30
management
Fukuoka
NOC
203.178.136.144/30
not available
203.178.141.64/29
2001:200:0:3000::/64

図 20: 広島 NOC

3 福岡

福岡 NOC では，日立 GR2000 をコアルータとして運用を行なっている．支線は 2 つあり，それぞれ帯域を必要としないローカル実験用の 100Mbps のセグメントと，グローバル実験用の 1Gbps のセグメントである．ローカル実験用の経路情報は現在，インターネットには広告していない．

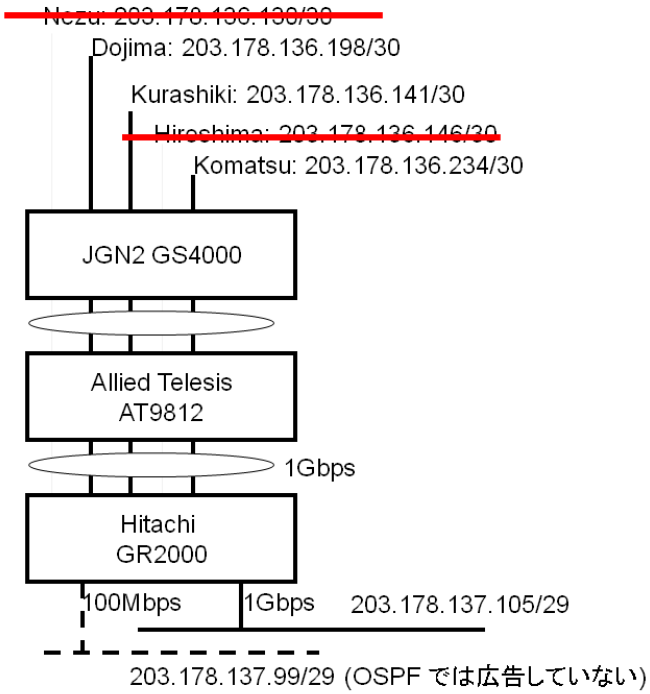


図 21: 福岡 NOC

3.1 バンコク

2007 年 5 月 15 日に設置されたバンコク NOC は，NECTEC や UniNET といったタイの学術研究組織との研究活動強化を目的に設立された．今年度も引き続き，WIDE プロジェクトとしての独自の回線は存在しないが，JGN-X の回線を利用し，VLAN を用いて WIDE インターネットをバンコクまで延長し，IPv4，および IPv6 の接続性を提供している．バンコク NOC は，JGN-X バンコク回線を収容している NECTEC と同じ建物に存在し，そこから UTP ケーブルを延伸し，バンコク NOC が存在する部屋にネットワークを引いている．バンコク NOC の主な利用者は，バンコクを中心に活動している SOI Asia プロジェクトのメンバーである Patcharee Basu，および関係者になる．

今年度も昨年度同様，SOI Asia プロジェクトで遠隔講義，講演をするための環境が整えられ，様々な授業やイベントへ参加した．

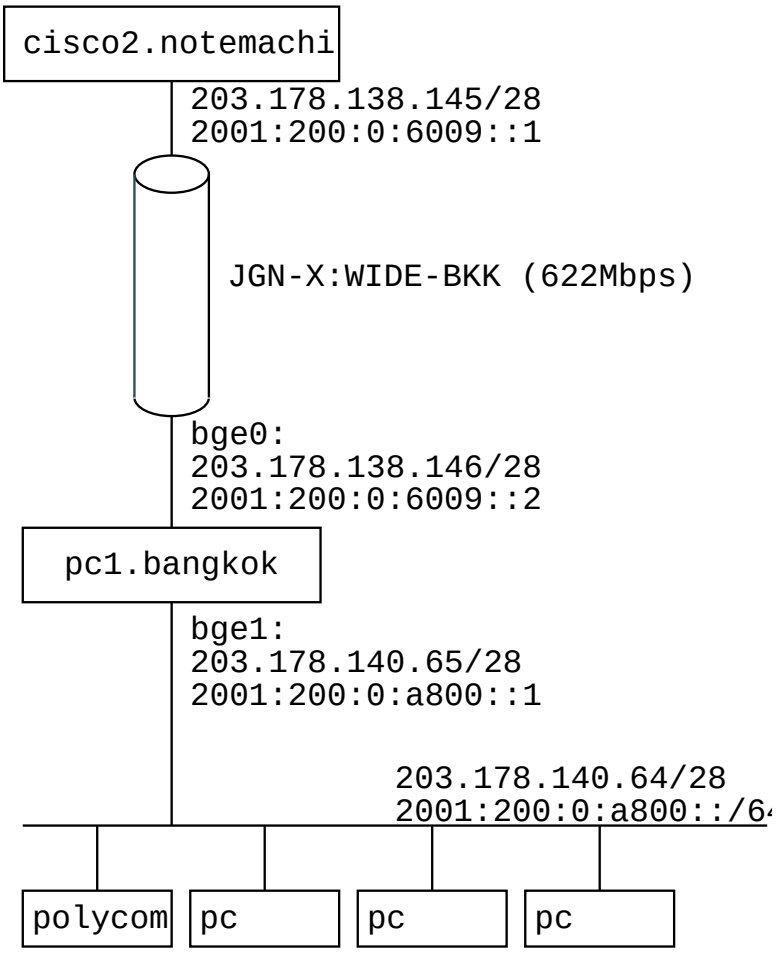


図 22: Bangkok NOC

4　two サーバの移行

本節では，two WG が WIDE インターネットの運用管理用に利用してるサーバ two.wide.ad.jp（以後 two サーバ） を WIDE クラウド上に移行した作業について報告する．

4.1　背景と概要

two サーバは 2002 年ごろから WIDE インターネットの運用管理に用いられてきた．そのため，ハードウェアの陳腐化によるスペック不足が問題になっていた．two WG では，two サーバの移行計画を検討し cloud WG で運用している WIDE クラウド環境へのサーバ移行を実施することとなった．当初の移行計画では，現行の two サーバシステムを可能な限りそのままクラウド環境へ移することを志向したが，two サーバで OS として利用している NetBSD が WIDE クラウド上で動作不具合が起きる既知の問題があり断念した．次に，WIDE クラウド上で動作するように修正した NetBSD を WIDE クラウド上でクローンし，two サーバのボリュームを dump と restore で移行することを計画した．しかしクローンした NetBSD サーバは HDD ボリュームサイズが小さく，WIDE クラウドには HDD ボリュームを追加する機能が存在しないため，この計画も断念することとなった．最終的に，OS まで含めた移行は断念し two サーバ上のデータのみコピーすることにした．WIDE クラウド上に，Ubuntu Linux のサーバをセットアップし two サーバ上のコンテンツとアカウント情報をコピーした．

4.2　移行と構築

two サーバは two WG メンバーのみが利用するホストであり，公開サービスなどは行なっていないためインフラとしての要素が低く，可能な限り WIDE クラウド環境の実験に協力する構成とした．インターネットへのアクセスは固定の IP アドレスを割り振るのではなく，WIDE クラウドが自動でホストに割り振るアドレスを利用した．EUI-64 で付与される IPv6 グローバルアドレスと DHCP で付与される IPv4 プライベートアドレスを用いている．IPv6 では，このアドレスを DNS の AAAA レコードに登録を行った．IPv4 は WIDE クラウドの map646 機構を用いて，IPv4 グローバルアドレスと IPv6 アドレスのマッピングを実施した．この機構により，two サーバへの IPv4 アクセスは，サーバ側では IPv6 アクセスとして認識される．また，この構成を採ることにより WIDE クラウド上での仮想サーバマイグレーションが可能となる．

4.3 web ページ生成システムの再構築

two サーバの web ページは，各 NOC ごとのページや two WG メンバーが運用するために必要な情報をまとめる場所としての役割を担っている．しかし，今まで利用していた web のシステムが古く移行の障害になったこと，またメンテナンスの可用性を考慮し，今回の two サーバの移行に伴って web ページの再構築を行った．

旧来の two サーバにおける web ページでは，コンテンツは各ページや NOC 毎に CVS によってレポジトリ管理され，php でページを生成していた．しかし，CVS では機能が少ないために柔軟な管理が難しく，またシステム全体が php で構築されていたため，脆弱性の問題や移行が難しかった．そこで，今回の再構築にあたって，リビジョン管理システムには Git を，Web ページの生成には python を用いて，すべてのシステムを作り直した．リビジョン管理に Git を用いることによって，hook などの多くの機能を用いて柔軟な制御が可能になり，今後新しく機能を追加する際に幅をもたせることができる．また，各レポジトリ内のコンテンツは今までのものをそのまま再利用できるように web ページを生成するシステムを構築した．そのため，現在の two サーバはレポジトリが Git 管理になったこと，外見が変更された意外には変更点はなく，今までと同様にコンテンツを表示することができるようになっている．

two サーバの重要な機能の 1 つとして，アドレス割り当て表がある．アドレス割り当て表は，設定ファイルに記載されたアドレスの割り当てと，実際の WIDE バックボーンの OSPF 網に流れる経路をつき合わせた結果を two サーバ上の web ページに表示する，アドレス割り当ての可視化ツールである．two WG では，これを用いて実際に流れている経路や割り当てた経路の確認を行っている．今回のサーバの移行に伴って，このアドレス割り当て可視化ツールも php から python を用いて書き直した．

また，この可視化ツールを使用するには，現在流れている経路を知るために two サーバ自身が OSPF 網に参加する必要がある．しかし，サーバを WIDE クラウド上に設置したことによって，OSPF 網に参加することができなくなってしまった．WIDE クラウドでは全てのユーザの VM は同一の L2 セグメントに収容される．そのため，このセグメントに対して OSPF Hello を流すことが現実的では無いためである．そこで，cloud WG の成果の 1 つである VXLAN を用いてセグメントを分ける構成をとった．この概要を図 23 に示す．WIDE クラウドの VM セグメントを収容する map646 ルータと two サーバの両方に実装した VXLAN を追加し，VM 用セグメント上に VXLAN を用いたオーバーレイによって仮想的に分離されたセグメントを構築した．この VXLAN セグメント上で map646 ルータと two サーバが OSPF によって経路交換することによって，two サーバはアドレス割り当て可視化のための経路情報を受け取ることが可能になった．

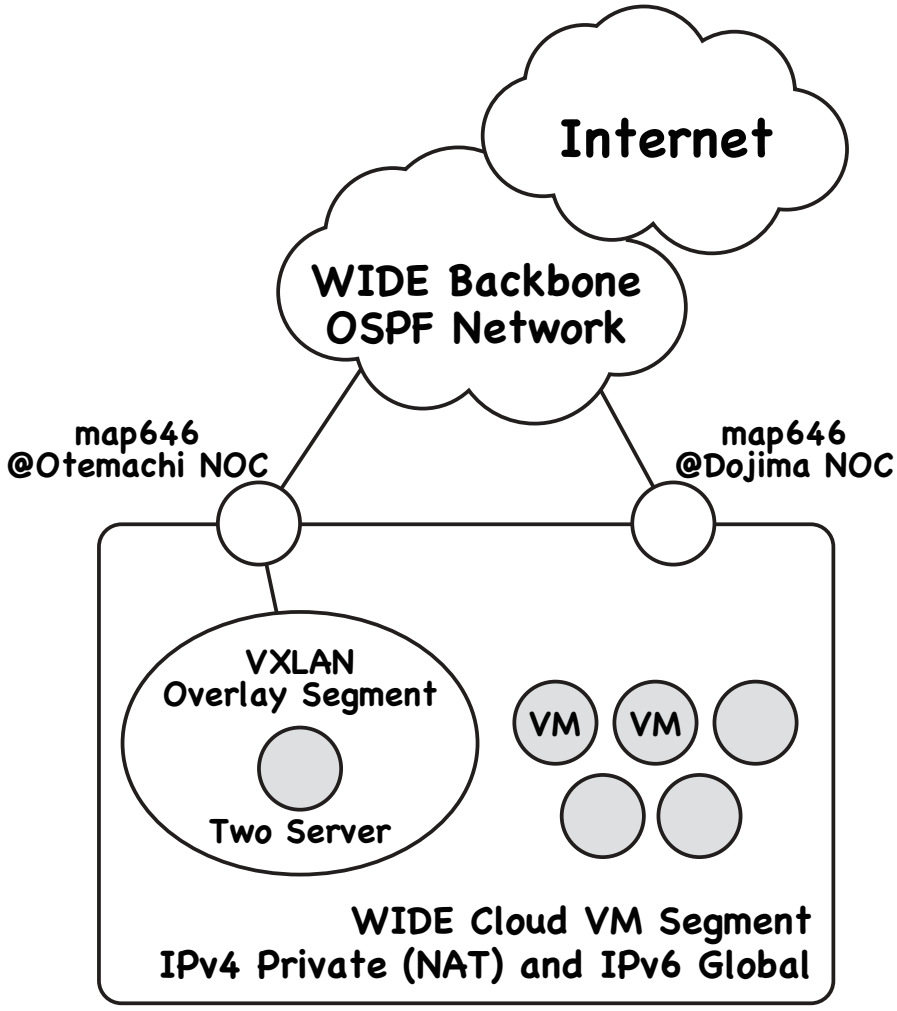


図 23: VXLAN を用いた WIDE クラウド上の two サーバと OSPF 網の接続

4.4 今後の課題

旧来の two サーバでは管理用ドキュメント以外に計測ツール類などが稼働していたが，今回の移行作業では環境をそのまま移行できなかったことと，計測ツールが動作している環境が古いため，いくつかのツールについて新設サーバ上で稼働させることができなかった．今後，現在の WIDE インターネットに則した計測ツールを新たに作成することで対応したい．

5 DNSSECの導入準備

本節では，two WG における DNSSEC の導入準備の進捗について報告する．

5.1 背景と概要

two WG は, WIDE インターネットで使用される wide.ad.jp ゾーン，逆引きゾーンの運用を行っている．現在 two WG が運用するゾーンでは DNSSEC が有効になっておらず，DNS 偽装などの攻撃に対して脆弱な状態であるため，two WG では 2010 年頃から wide.ad.jp ゾーン 及び逆引きゾーンへの DNSSEC 導入にむけた準備を行っている．

5.2 方針

DNSSEC は導入後に複雑な運用手順が必要になり，鍵の更新やゾーンの再署名を忘れるとゾーンの名前解決自体に支障が出る．従って，DNSSEC の運用はできるだけ自動化することが望ましい．そこで，two WG では DNSSEC 運用支援ツールを開発し, 運用試験を行っている．運用試験終了後は，順次 wide.ad.jp ゾーン及び逆引きゾーンを DNSSEC に対応させる予定である．

5.3 DNSSEC運用ツールの開発

DNSSEC運用ツールの既存実装としてはOpenDNSSECが存在する．OpenDNSSECはデーモンとして動作し，ZSK 及び KSK の自動ロールオーバに対応している．しかし，導入の際に多数の外部ライブラリが必要であり，SoftHSM などと併用することを前提として設計されている．そのため，ソフトウェアとしての規模が大きくtwo WG が管理する小中規模のゾーンの運用には向かないという結論に至った．そこで，two WG では小中規模ゾーンの DNSSEC 運用支援ツールとして dnssec.pl を開発した．dnssec.pl は Perl で書かれた小規模なスクリプトであり，cron 等で定期的に実行することを想定している．dnssec.pl が持つ機能は次の通りである．

- 設定ファイルによるゾーン毎のポリシー管理
- ZSK 及び KSK の自動ロールオーバ
- KSK のロールオーバ時に指定されたメールアドレスに通知 (二重署名法のみ)
- 自動再署名
- 使用しなくなった鍵ファイルの移動

- その他 dnssec-keygen 及び dnssec-signzone で使用可能なパラメータの指定

5.4 dnssec.pl の運用試験

dnssec.pl はフルスクラッチで新たに開発されたツールであるため，バグが含まれている可能性がある．また，ZSK 及び KSK のロールオーバに関する設定項目は多肢に渡り，設定を誤ると最悪の場合名前解決が不可能になる．以上の理由から dnssec.pl の運用試験が必要であるため，2012 年 10 月に dnssec.pl を two WG が管理する wide-dnssec.org ドメインに導入した．wide-dnssec.org ドメインは two WG が DNSSEC 運用手順の検討と開発ツールの検証を目的として取得したドメインである．

5.5 今後の課題

dnssec.pl の運用試験終了後, 順次 wide.ad.jp ゾーン及び逆引きゾーンを DNSSEC に対応させる予定である．ZSK 及び KSK のロールオーバや NSEC3 のイテレーションに関するパラメータも今後の検討課題である．

5.6 dnssec.pl のソースコード

dnssec.pl のソースコードを以下に示す．

```
#!/usr/bin/perl

use strict;
use warnings;
use Getopt::Std;
use File::Basename;
use File::Spec;
use File::Copy;
use Time::Piece;
use Time::Seconds;
use Sys::Hostname;

sub output_usage {
  print "\n";
  print "Usage:\n";
  print "\tdnssec.pl [zonefile]\n";
  print "\t\tSign [zone file] based on dnssec.conf.\n";
  print "\tdnssec.pl -a\n";
  print "\t\tSign all zones and rollover KSK/ZSK key based on dnssec.conf(for cron).\n";
  print "\t\tYou should execute this command at least once a day.";
  print "\n\n";

  exit(0);
}

sub parse_config {
  my $config_path = shift;
  open(my $fh, "<", $config_path) or die("error");

  my $mode = "top";
  my $statement = "";
  my $value = "";

  my $ksk = shift;
  my $zsk = shift;
  my $sign = shift;
  my $ksk_zone;
  my $zsk_zone;
  my $sign_zone;

  while(my $line = <$fh>){
    chomp($line);
    $line =~ s/^\s*(.*?)\s*$/$1/;
    my @args = split(/ /, $line);
    my @args_preformatted;

    foreach my $arg (@args){
          if($arg =~ /^#.*$/){
        last;
      }elsif($arg =~ /^(.+);$/){
        push(@args_preformatted, $1);
        push(@args_preformatted, ";");
      }elsif($arg =~ /^(.+){$/){
        push(@args_preformatted, $1);
        push(@args_preformatted, "{");
      }else{
        push(@args_preformatted, $arg);
      }
    }

    foreach my $arg (@args_preformatted){
      if($mode eq "top"){
        if($statement eq ""){
          if($arg eq "ksk" || $arg eq "zsk" || $arg eq "sign"){
            $statement = $arg;
          }else{
            die("config error");
          }
        }elsif($value eq ""){
          if($statement eq "ksk"){
            if($arg eq "{"){
              my %zone;
              $ksk->{'zone'} = \%zone;

              $mode = $statement;
              $statement = "";
              $value = "";
            }else{
              die("config error");
            }
```

```
    }elsif($statement eq "zsk"){
      if($arg eq "{"){
        my %zone;
        $zsk->{'zone'} = \%zone;

        $mode = $statement;
        $statement = "";
        $value = "";
      }else{
        die("config error");
      }
    }elsif($statement eq "sign"){
      if($arg eq "{"){
        my %zone;
        $sign->{'zone'} = \%zone;

        $mode = $statement;
        $statement = "";
        $value = "";
      }else{
        die("config error");
      }
    }
  }else{
    die("config error");
  }
}elsif($mode eq "ksk"){
  if($arg eq "}"){
    if($statement eq "" && $value eq ""){
      $mode = "top";
    }else{
      die("config error");
    }
  }elsif($statement eq ""){
    if($arg eq "publish" || $arg eq "activate"
      || $arg eq "retire" || $arg eq "delete"
      || $arg eq "notify" || $arg eq "gen-interval"
      || $arg eq "zone" || $arg eq "mailto" || $arg eq "mailfrom"
      || $arg eq "algorithm" || $arg eq "keysize"
      || $arg eq "sendmail"){

      $statement = $arg;
    }else{
      die("config error");
    }
  }elsif($value eq ""){
    $value = $arg;
  }elsif($arg eq ";"){
    $ksk->{$statement} = $value;

    $statement = "";
    $value = "";
  }elsif($arg eq "{"){
    if($statement eq "zone"){
      my %ksk_zone;
      $ksk->{$statement}->{$value} = \%ksk_zone;

      $mode = "ksk_zone";
      $ksk_zone = $ksk->{$statement}->{$value};

      $statement = "";
      $value = "";
    }else{
      die("config error");
    }
  }else{
    die("config error");
  }
}elsif($mode eq "zsk"){
  if($arg eq "}"){
    if($statement eq "" && $value eq ""){
      $mode = "top";
    }else{
      die("config error");
    }
  }elsif($statement eq ""){
    if($arg eq "publish" || $arg eq "activate"
      || $arg eq "retire" || $arg eq "delete"
      || $arg eq "gen-interval"
      || $arg eq "algorithm" || $arg eq "keysize"
      || $arg eq "zone"){

      $statement = $arg;
    }else{
      die("config error");
    }
  }elsif($value eq ""){
    $value = $arg;
  }elsif($arg eq ";"){
    $zsk->{$statement} = $value;
```

```
      $statement = "";
      $value = "";
    }elsif($arg eq "{"){
      if($statement eq "zone"){
        my %zsk_zone;
        $zsk->{$statement}->{$value} = \%zsk_zone;

        $mode = "zsk_zone";
        $zsk_zone = $zsk->{$statement}->{$value};

        $statement = "";
        $value = "";
      }else{
        die("config error");
      }
    }else{
      die("config error");
    }

  }elsif($mode eq "sign"){
    if($arg eq "}"){
      if($statement eq "" && $value eq ""){
        $mode = "top";
      }else{
        die("config error");
      }
    }elsif($statement eq ""){
      if($arg eq "keydir" || $arg eq "serial" || $arg eq "retired-keydir"
      || $arg eq "dssetdir" || $arg eq "signeddir" || $arg eq "keygen"
      || $arg eq "signzone" || $arg eq "zone" || $arg eq "dsfromkey"
      || $arg eq "sign-interval" || $arg eq "rndc" || $arg eq "nsec3"
      || $arg eq "nsec3-iterations"){
        $statement = $arg;
      }else{
        die("config error");
      }
    }elsif($value eq ""){
      if($statement eq "keydir" || $statement eq "retired-keydir"
      || $statement eq "dssetdir" || $statement eq "signeddir"){
        if($arg !~ /.*\/$/){
          $value = $arg . "/";
        }else{
          $value = $arg;
        }
      }else{
        $value = $arg;
      }
    }elsif($arg eq ";"){
      $sign->{$statement} = $value;

      $statement = "";
      $value = "";
    }elsif($arg eq "{"){
      if($statement eq "zone"){
        my %sign_zone;
        $sign->{$statement}->{$value} = \%sign_zone;

        $mode = "sign_zone";
        $sign_zone = $sign->{$statement}->{$value};

        $statement = "";
        $value = "";
      }else{
        die("config error");
      }
    }else{
      die("config error");
    }
  }elsif($mode eq "ksk_zone"){
    if($arg eq "}"){
      if($statement eq "" && $value eq ""){
        $mode = "ksk";
        $ksk_zone = "";
      }else{
        die("config error");
      }
    }elsif($statement eq ""){
      if($arg eq "publish" || $arg eq "activate"
        || $arg eq "retire" || $arg eq "delete"
        || $arg eq "notify" || $arg eq "gen-interval"
        || $arg eq "mailto" || $arg eq "mailfrom"
        || $arg eq "algorithm" || $arg eq "keysize"
        || $arg eq "sendmail"){
```

```
        $statement = $arg;
      }else{
        die("config error");
      }
    }elsif($value eq ""){
      $value = $arg;
    }elsif($arg eq ";"){
      $ksk_zone->{$statement} = $value;

      $statement = "";
      $value = "";
    }else{
      die("config error");
    }
  }elsif($mode eq "zsk_zone"){
    if($arg eq "}"){
      if($statement eq "" && $value eq ""){
        $mode = "zsk";
        $zsk_zone = "";
      }else{
        die("config error");
      }
    }elsif($statement eq ""){
      if($arg eq "publish" || $arg eq "activate"
        || $arg eq "retire" || $arg eq "delete"
        || $arg eq "algorithm" || $arg eq "keysize"
        || $arg eq "gen-interval"){


        $statement = $arg;
      }else{
        die("config error");
      }
    }elsif($value eq ""){
      $value = $arg;
    }elsif($arg eq ";"){
      $zsk_zone->{$statement} = $value;

      $statement = "";
      $value = "";
    }else{
      die("config error");
    }
  }elsif($mode eq "sign_zone"){
    if($arg eq "}"){
      if($statement eq "" && $value eq ""){
        $mode = "sign";
        $sign_zone = "";
      }else{
        die("config error");
      }
    }elsif($statement eq ""){
      if($arg eq "keydir" || $arg eq "serial" || $arg eq "retired-keydir"
      || $arg eq "dssetdir" || $arg eq "signeddir" || $arg eq "keygen"
      || $arg eq "signzone" || $arg eq "sign-interval" || $arg eq "file"
      || $arg eq "dsfromkey" || $arg eq "rndc" || $arg eq "record-class"
      || $arg eq "view" || $arg eq "nsec3" || $arg eq "nsec3-iterations"){
        $statement = $arg;
      }else{
        die("config error");
      }
    }elsif($value eq ""){
      if($statement eq "keydir" || $statement eq "retired-keydir"
      || $statement eq "dssetdir" || $statement eq "signeddir"){
        if($arg !~ /.*\/$/){
          $value = $arg . "/";
        }else{
          $value = $arg;
        }
      }else{
        $value = $arg;
      }
    }elsif($arg eq ";"){
      $sign_zone->{$statement} = $value;
```

```
          $statement = "";
          $value = "";
        }else{
          die("config error");
        }
      }
    }


  }

  close($fh);

}


sub zone_to_origin {
  my $sign_config = shift;
  my $inputfile_full = shift;

  my $zone = $sign_config->{'zone'};
  foreach my $origin (keys %$zone){
    if($sign_config->{'zone'}->{$origin}->{'file'} eq $inputfile_full){
      return $origin;
    }
  }

  die("zone not found in configuration file");
}

sub do_ksk_keygen {
  my $ksk_config = shift;
  my $sign_config = shift;
  my $origin = shift;

  my $command = $sign_config->{'keygen'};
  $command = $command . " -K " . $sign_config->{'keydir'};
  $command = $command . " -r /dev/urandom";
  $command = $command . " -f ksk";

  if($sign_config->{'nsec3'}){
    $command = $command . " -3";
  }

  if($ksk_config->{'algorithm'}){
    $command = $command . " -a " . $ksk_config->{'algorithm'};
  }

  if($ksk_config->{'keysize'}){
    $command = $command . " -b " . $ksk_config->{'keysize'};
  }

  if($ksk_config->{'publish'}){
    $command = $command . " -P " . $ksk_config->{'publish'};
  }

  if($ksk_config->{'activate'}){
    $command = $command . " -A " . $ksk_config->{'activate'};
  }

  if($ksk_config->{'retire'}){
    $command = $command . " -I " . $ksk_config->{'retire'};
  }

  if($ksk_config->{'delete'}){
    $command = $command . " -D " . $ksk_config->{'delete'};
  }

  $command = $command . " " . $origin;

  print "Generating KSK key of " . $origin . "...\n";
  my $result = `$command`;
  print $result;

  my @result_lines = split(/\n/, $result);
  foreach my $line (@result_lines){
    if($line =~ /(K$origin\.\+[0-9]+\+[0-9]+)/){
      return $1;
    }
  }

  return 0;
}
```

```
sub do_zsk_keygen {
  my $zsk_config = shift;
  my $sign_config = shift;
  my $origin = shift;

  my $command = $sign_config->{'keygen'};
  $command = $command . " -K " . $sign_config->{'keydir'};
  $command = $command . " -r /dev/urandom";

  if($sign_config->{'nsec3'}){
    $command = $command . " -3";
  }

  if($zsk_config->{'algorithm'}){
    $command = $command . " -a " . $zsk_config->{'algorithm'};
  }

  if($zsk_config->{'keysize'}){
    $command = $command . " -b " . $zsk_config->{'keysize'};
  }

  if($zsk_config->{'publish'}){
    $command = $command . " -P " . $zsk_config->{'publish'};
  }

  if($zsk_config->{'activate'}){
    $command = $command . " -A " . $zsk_config->{'activate'};
  }

  if($zsk_config->{'retire'}){
    $command = $command . " -I " . $zsk_config->{'retire'};
  }

  if($zsk_config->{'delete'}){
    $command = $command . " -D " . $zsk_config->{'delete'};
  }

  $command = $command . " " . $origin;

  print "Generating ZSK key of " . $origin . "...\n";
  my $result = `$command`;
  print $result;

  my @result_lines = split(/\n/, $result);
  foreach my $line (@result_lines){
    if($line =~ /(K$origin\.\+[0-9]+\+[0-9]+)/){
      return $1;
    }
  }

  return 0;
}

sub do_signzone {
  my $sign_config = shift;
  my $origin = shift;
  my $inputfile = shift;
  my $inputfile_full = shift;

  my $signedfile = $sign_config->{'signeddir'} . $inputfile . ".signed";

  my $command = $sign_config->{'signzone'} . " -S";

  if($sign_config->{'nsec3'}){
    $command = $command . " -3 " . $sign_config->{'nsec3'};
    if($sign_config->{'nsec3-iterations'}){
      $command = $command . " -H " . $sign_config->{'nsec3-iterations'};
    }
  }

  $command = $command . " -K " . $sign_config->{'keydir'};
  $command = $command . " -N " . $sign_config->{'serial'};
  $command = $command . " -o " . $origin;
  $command = $command . " -d " . $sign_config->{'dssetdir'};
  $command = $command . " -f " . $signedfile;
  $command = $command . " " . $inputfile_full;

  my $rndc_command = $sign_config->{'rndc'} . " reload " . $origin;

  if($sign_config->{'record-class'}){
    $rndc_command = $rndc_command . " " . $sign_config->{'record-class'};
  }

  if($sign_config->{'view'}){
    $rndc_command = $rndc_command . " " . $sign_config->{'view'};
  }

  print "Signing " . $origin . "...\n";
  print `$command`;
  print `$rndc_command`;
}
```

```
sub move_retired_key {
  my $sign_config = shift;
  my $origin = shift;
  my $now = shift;

  opendir(my $dh, $sign_config->{'keydir'}) or die("error");
  my @keys = readdir($dh);
  closedir($dh);

  foreach my $key (@keys){
    my $keylimit;

    if($key =~ /^K$origin\./){
      open(my $fh, "<", $sign_config->{'keydir'} . $key) or die("error");

      while(my $line = <$fh>){
        if($line =~ /Delete:\s+([0-9]+?)\s+/){
          $keylimit = Time::Piece->strptime($1, '%Y%m%d%H%M%S');
        }
      }

      close($fh);

      if($keylimit){
        if($now > $keylimit){
          move($sign_config->{'keydir'} . $key,
            $sign_config->{'retired-keydir'} . $key)
            or print "Retired-key directory not found\n";
        }
      }
    }
  }
}

sub read_keys {
  my $sign_config = shift;
  my $origin = shift;
  my $ksk_keys = shift;
  my $zsk_keys = shift;

  opendir(my $dh, $sign_config->{'keydir'}) or die("error");
  my @keys = readdir($dh);
  closedir($dh);

  foreach my $key (@keys){
    if($key =~ /^K$origin\..+\.key/){
      my %key_current;
      my $key_type = "";

      open(my $fh, "<", $sign_config->{'keydir'} . $key) or die("error");
      while(my $line = <$fh>){
        if($line =~ /key-signing/){
          $key_type = "ksk";
        }elsif($line =~ /zone-signing/){
          $key_type = "zsk";
        }

        if($line =~ /Created:\s+([0-9]+?)\s+/){
          $key_current{'created'} = $1;
        }

        if($line =~ /Publish:\s+([0-9]+?)\s+/){
          $key_current{'publish'} = $1;
        }

        if($line =~ /Activate:\s+([0-9]+?)\s+/){
          $key_current{'activate'} = $1;
        }

        if($line =~ /Inactive:\s+([0-9]+?)\s+/){
          $key_current{'retire'} = $1;
        }

        if($line =~ /Delete:\s+([0-9]+?)\s+/){
          $key_current{'delete'} = $1;
        }
      }
      close($fh);

      if($key_type eq "ksk"){
        $ksk_keys->{$key} = \%key_current;
      }elsif($key_type eq "zsk"){
        $zsk_keys->{$key} = \%key_current;
      }
    }
  }
}
```

```
sub is_event {
  my $sign_config = shift;
  my $ksk_config = shift;
  my $zsk_config = shift;
  my $ksk_keys = shift;
  my $zsk_keys = shift;
  my $ksk_keys_newest = shift;
  my $zsk_keys_newest = shift;
  my $timestamp = shift;
  my $now = shift;
  my $flags = shift;

  my @key_types = ('ksk', 'zsk');
  foreach my $key_type (@key_types){
    my $keys;
    my $keys_newest;
    my $keys_config;

    if($key_type eq 'ksk'){
      $keys = $ksk_keys;
      $keys_newest = $ksk_keys_newest;
      $keys_config = $ksk_config;
    }elsif($key_type eq 'zsk'){
      $keys = $zsk_keys;
      $keys_newest = $zsk_keys_newest;
      $keys_config = $zsk_config;
    }

    foreach my $key (keys %$keys){
      if($keys->{$key}->{'created'}){
        if($keys_newest->{'created'}){
          my $newest_gen = Time::Piece->strptime($keys_newest->{'created'}, '%Y%m%d%H%M%S');
          my $current_gen = Time::Piece->strptime($keys->{$key}->{'created'}, '%Y%m%d%H%M%S');

          if($current_gen > $newest_gen){
            %$keys_newest = %{$keys->{$key}};
          }
        }else{
          %$keys_newest = %{$keys->{$key}};
        }
      }

      my @events = ('activate', 'publish', 'retire', 'delete');
      foreach my $event (@events){
        if($keys->{$key}->{$event}){
          if($timestamp->{$key_type . '_' . $event}){
            my $last_event = Time::Piece->strptime($timestamp->{$key_type . '_' . $event}, '%Y%m%d%H%M%S');
            my $next_event = Time::Piece->strptime($keys->{$key}->{$event}, '%Y%m%d%H%M%S');

            if($last_event < $next_event && $next_event < $now){
              $flags->{$key_type . '_' . $event} = 1;
            }
          }else{
            my $next_event = Time::Piece->strptime($keys->{$key}->{$event}, '%Y%m%d%H%M%S');
            if($next_event < $now){
              $flags->{$key_type . '_' . $event} = 1;
            }
          }
        }
      }
    }

    if($keys_newest->{'created'}){
      my $last_create = Time::Piece->strptime($keys_newest->{'created'}, '%Y%m%d%H%M%S');

      if($keys_config->{'gen-interval'} =~ /^\+([0-9]+)d$/){
        my $offset = $1;
        if($last_create + ONE_DAY * $offset < $now){
          $flags->{$key_type . '_gen'} = 1;

          if($keys_config->{'publish'} eq "now"){
            $flags->{$key_type . '_publish'} = 1;
          }

          if($keys_config->{'activate'} eq "now"){
            $flags->{$key_type . '_activate'} = 1;
          }

          if($key_type eq "ksk" && $keys_config->{'activate'} eq "now"){
            $flags->{'ksk_notify'} = 1;
          }
        }
      }else{
        print "Invalid gen-interval format\n";
      }
```

```
    }else{
      $flags->{$key_type . '_gen'} = 1;
      $flags->{$key_type . '_publish'} = 1;
      $flags->{$key_type . '_activate'} = 1;
      $keys_config->{'publish'} = "now";
      $keys_config->{'activate'} = "now";
    }
  }

  if($timestamp->{'sign'}){
    my $last_sign = Time::Piece->strptime($timestamp->{'sign'}, '%Y%m%d%H%M%S');

    if($sign_config->{'sign-interval'} =~ /^\+([0-9]+)d$/){
      my $offset = $1;
      if($last_sign + ONE_DAY * $offset < $now){
        $flags->{'sign'} = 1;
      }
    }else{
      print "Invalid sign-interval format\n";
    }
  }else{
    $flags->{'sign'} = 1;
  }
}

sub keyfile_to_dsrecord {
  my $sign_config = shift;
  my $keyfile = shift;

  my $command = $sign_config->{'dsfromkey'};
  $command = $command . " " . $sign_config->{'keydir'} . $keyfile;

  return `$command`;
}

sub notify_ksk_rotation {
  my $sign_config = shift;
  my $ksk_config = shift;
  my $origin = shift;
  my $now = shift;
  my $ksk_keys_newest = shift;
  my $dsrecords = shift;

  if($ksk_config->{'mailto'} && $ksk_keys_newest->{'retire'}){
    my $next_retire = Time::Piece->strptime($ksk_keys_newest->{'retire'}, '%Y%m%d%H%M%S');
    my $remaining_days = int(($next_retire - $now)->days);

    my $from;
    my $to;
    my $subject;
    my $body;

    if($ksk_config->{'mailfrom'}){
      $from = $ksk_config->{'mailfrom'};
    }else{
      $from = getpwuid($>) . "\@" . hostname();
    }

    $to = $ksk_config->{'mailto'};

    $subject = "DS Record Update Notification($origin)";

    $body = "Dear Administrators,\n\n";
    $body = $body . "New KSK key of $origin is activated.\n";
    $body = $body . "Old KSK key will retire in $remaining_days days ";
    $body = $body . "($next_retire GMT).\n";
    $body = $body . "Please update DS record of $origin.\n";
    $body = $body . "\n";
    $body = $body . "New DS records:\n";
    $body = $body . $dsrecords . "\n";

    open(my $mail,"| $ksk_config->{'sendmail'} -t $from $to") or die 'error';
    print $mail "From: $from\n";
    print $mail "To: $to\n";
    print $mail "Subject: $subject\n";
    print $mail "MIME-Version: 1.0\n";
    print $mail "Content-Type: text/plain;\n";
    print $mail "Content-Transfer-Encoding: 7bit\n";
    print $mail "\n";
    print $mail "$body\n";
    close($mail);

    return;
  }
}
```

```
sub add_default_config {
  my $default = shift;
  my $zone_specific = shift;

  foreach my $key (keys %$default){
    if(!$zone_specific->{$key}){
      $zone_specific->{$key} = $default->{$key};
    }
  }
}

sub get_origin_timestamp {
  my $timestamp_path = shift;
  my $origin = shift;
  my $timestamp = shift;

  open(my $fh, "<", $timestamp_path) or die("cannot open timestamp file");

  while(my $line = <$fh>){
    chomp($line);
    if($line =~ /^(.*?)#.*$/){
      $line = $1;
    }

    if($line =~ /^\s*$origin\s+(.+?)\s+([0-9]{14})/){
      my $type = lc $1;
      $timestamp->{$type} = $2;
    }
  }

  close($fh);
}

sub set_origin_timestamp {
  my $timestamp_path = shift;
  my $origin = shift;
  my $timestamp = shift;
  my $buf = "";

  open(my $fh, "<", $timestamp_path) or die("cannot open timestamp file");
  while(my $line = <$fh>){
    my $found_flag = 0;
    chomp($line);
    foreach my $key (keys %$timestamp){
      my $type = uc $key;

      if($line =~ /^\s*$origin\s+$type\s+([0-9]{14})/){
        $buf = $buf . $origin . " $type " . $timestamp->{$key} . "\n";
        delete $timestamp->{$key};
        $found_flag = 1;
      }
    }

    if(!$found_flag){
      $buf = $buf . $line . "\n";
    }
  }
  close($fh);

  foreach my $key (keys %$timestamp){
    my $type = uc $key;
    $buf = $buf . $origin . " $type " . $timestamp->{$key} . "\n";
  }

  open(my $fh_w, ">", $timestamp_path) or die("cannot open timestamp file");
  print $fh_w $buf;
  close($fh_w);

}
```

```
my %ksk;
my %zsk;
my %sign;

my %opt;
getopts('c:t:ah', \%opt) || output_usage();

my $config_path = "/etc/dnssec/dnssec.conf";
my $timestamp_path = "/etc/dnssec/dnssec.timestamp";
my $inputfile = shift;
my $inputfile_full;

if($opt{h}){
  output_usage();
}

if($opt{c}){
  $config_path = $opt{c};
}

if($opt{t}){
  $timestamp_path = $opt{t};
}

if(!$opt{a} && $inputfile){
  $inputfile_full = $inputfile;
  $inputfile_full = File::Spec->rel2abs($inputfile_full);
  $inputfile = basename($inputfile);
}elsif($opt{a} && $inputfile){
  output_usage();
}elsif(!$opt{a} && !$inputfile){
  output_usage();
}

# This is default config. You should change dnssec.conf
$ksk{'sendmail'} = "/usr/sbin/sendmail";

$sign{'keydir'} = "/etc/bind/master/keys/";
$sign{'retired-keydir'} = "/etc/bind/master/retired/";
$sign{'dssetdir'} = "/etc/bind/master/dsset/";
$sign{'signeddir'} = "/etc/bind/master/signed/";
$sign{'keygen'} = "/usr/sbin/dnssec-keygen";
$sign{'dsfromkey'} = "/usr/sbin/dnssec-dsfromkey";
$sign{'signzone'} = "/usr/sbin/dnssec-signzone";
$sign{'rndc'} = "/usr/sbin/rndc";
$sign{'serial'} = "unixtime";

# parse config file
parse_config($config_path, \%ksk, \%zsk, \%sign);

if($opt{a}){
  my $zone = $sign{'zone'};
  foreach my $origin (keys %$zone){
    my $ksk_config;
    my $zsk_config;
    my $sign_config;
    my $file;
    my $file_full;

    if($ksk{zone}->{$origin}){
      add_default_config(\%ksk, $ksk{zone}->{$origin});
      $ksk_config = $ksk{zone}->{$origin};
    }else{
      $ksk_config = \%ksk;
    }

    if($zsk{zone}->{$origin}){
      add_default_config(\%zsk, $zsk{zone}->{$origin});
      $zsk_config = $zsk{zone}->{$origin};
    }else{
      $zsk_config = \%zsk;
    }

    add_default_config(\%sign, $sign{zone}->{$origin});
    $sign_config = $sign{zone}->{$origin};

    $file_full = $sign_config->{file};
    $file = basename($file_full);
```

```
    my %timestamp_new;
    my %timestamp_old;
    my %ksk_keys;
    my %zsk_keys;
    my %ksk_keys_newest;
    my %zsk_keys_newest;
    my %event_flags;
    my $generated_keyfile;

    my $time = Time::Piece::gmtime();

    read_keys($sign_config, $origin, \%ksk_keys, \%zsk_keys);
    get_origin_timestamp($timestamp_path, $origin, \%timestamp_old);

    is_event($sign_config, $ksk_config, $zsk_config,
      \%ksk_keys, \%zsk_keys, \%ksk_keys_newest,
      \%zsk_keys_newest, \%timestamp_old, $time, \%event_flags);

    if($event_flags{'ksk_gen'}){
      $generated_keyfile = do_ksk_keygen($ksk_config, $sign_config, $origin);
    }

    if($event_flags{'zsk_gen'}){
      $generated_keyfile = do_zsk_keygen($zsk_config, $sign_config, $origin);
    }

    if(keys %event_flags){
      $event_flags{'sign'} = 1;
      do_signzone($sign_config, $origin, $file, $file_full);
      move_retired_key($sign_config, $origin, $time);
    }

    $time = Time::Piece::gmtime();
    foreach my $key (keys %event_flags){
      $timestamp_new{$key} = $time->strftime('%Y%m%d%H%M%S');
    }

    set_origin_timestamp($timestamp_path, $origin, \%timestamp_new);

    # NOTE:  Notification is assumed to use for dubble sign method of KSK.
    #   Notification occurs when new KSK is activated,
    #  and notification remaining days is based on RETIRE option. not REVOKE.
    #  And you have to decide by yourself which DS record of 'DSSET dump' to register
    #  to higher zone.
    if($event_flags{'ksk_notify'}){
      my $dsrecords = keyfile_to_dsrecord($sign_config, $generated_keyfile);
      notify_ksk_rotation($sign_config, $ksk_config,  $origin, $time, \%ksk_keys_newest, $dsrecords);
    }
  }

}else{
  my $sign_config;
  my $origin = zone_to_origin(\%sign, $inputfile_full);
  add_default_config(\%sign, $sign{'zone'}->{$origin});
  $sign_config = $sign{'zone'}->{$origin};

  do_signzone($sign_config, $origin, $inputfile, $inputfile_full);
}
```

5.7 dnssec.pl の設定ファイル

dnssec.pl の設定ファイルの例を以下に示す.

```
########################
#  DNSSEC config file
########################

ksk {
  # default config
  gen-interval +300d;
  publish now;
  activate now;
  retire +330d;
  delete +335d;
  algorithm RSASHA256;
  keysize 2048;
  mailto eden@sfc.wide.ad.jp;
  mailfrom admin@wide-dnssec.org;
  sendmail /usr/sbin/sendmail;

  zone wide-dnssec.org {
    # zone specific config(prior to default config)
    mailto eden@sfc.wide.ad.jp;
    mailfrom admin@wide-dnssec.org;
    algorithm RSASHA256;
    keysize 2048;
    gen-interval +300d;
    publish now;
    activate now;
    retire +330d;
    delete +335d;
  }
}

zsk {
  gen-interval +30d;
  publish now;
  activate +5d;
  retire +35d;
  delete +35d;
  algorithm RSASHA256;
  keysize 1024;

  zone wide-dnssec.org {
    algorithm RSASHA256;
    keysize 1024;
    publish now;
    activate +5d;
    retire +35d;
    delete +35d;
    gen-interval +30d;
  }
}

sign {
  keydir /etc/bind/master/keys;
  retired-keydir /etc/bind/master/retired-keys;
  serial unixtime;
  dssetdir /etc/bind/master/dsset;
  signeddir /etc/bind/master/signed;
  keygen /usr/sbin/dnssec-keygen;
  dsfromkey /usr/sbin/dnssec-dsfromkey;
  rndc /usr/sbin/rndc;
  signzone /usr/sbin/dnssec-signzone;
  sign-interval +7d;

  zone wide-dnssec.org {
    serial unixtime;
    file /etc/bind/master/db.wide-dnssec.org;
    sign-interval +7d;
    nsec3 a7;
    nsec3-iterations 15;
    #record-class in;
    #view internal;
  }
}
```

6 おわりに

本年度も WIDE バックボーンネットワークの安定運用を行ってきた．来年度は，AS 対外接続地点への計測ノードの増設，バックボーンルータや Call Manager の入れ替えを予定している．ほか，OSPF 計測の再開など広域運用環境を用いた実験を精力的に行っていく予定である．

7 CopyRight