学士論文 2020 年度 (令和 2 年度)

既存のソフトウェア資産を活用した
高速なソフトウェアロードバランサの設計と提案

慶應義塾大学 環境情報学部

橘 直雪

学士論文要旨 - 2020 年度 (令和 2 年度)

既存のソフトウェア資産を活用した
高速なソフトウェアロードバランサの設計と提案

近年，ロードバランサーはハードウェアアプライアンスのものからソフトウェアのものへ変遷しつつある．これにより，データセンターは物理スペースや経済的なコスト問題を解決したが，これらのソフトウェアロードバランサは既存のコントロールプレーンの API を利用することができず，利用者はデータプレーンよりもコントロールプレーンの開発にコストを掛けているのが実情である．

この問題はソフトウェアロードバランサに限った問題ではなく，ソフトウェアルーターや ASIC 等を用いたオフローディングにも共通した問題であり，

キーワード:
1. Load Balancer，2. XDP，3. 負荷分散

慶應義塾大学 環境情報学部
橘 直雪

Abstract of Bachelor's Thesis - Academic Year 2020

Dynamic advertising method of Explicit Address Mapping in IPv6 single stack network.

Write abstract

Keywords :
1. Data center network, 2. Network operation, 3. IPv6 transition mechanism

Keio University Faculty of Environment and Information Studies
Naoyuki Tachibana

目 次

図 目 次

表 目 次

第1章　序論

本章では本研究の背景と全体の構成について記述する．

1.1　背景

1.2　本研究の目的

本研究では，Layer-4 ロードバランサにおける

1.3　本論文の構成

本論文の構成を以下に示す．

第 2 章では，IPv6 シングルスタックネットワークにおける IPv4 サービス提供手法に関してそれぞれの特徴や利点を紹介し比較する．

第 3 章では，IPv4/IPv6 プロトコル変換を利用した IPv4 サービス提供手法の一つである SIIT-DC のアーキテクチャと，解決すべき課題について明らかにする．

第 4 章では，SIIT-DC の課題を解決するために考えられる手法について論ずる．

第 5 章では，本研究において提案するダイナミックなアドレス変換テーブル広告手法の要件と構成について記述する．またメッセージングプロトコルとして採用した BGP の技術的利点について述べる．

第 6 章では，本提案手法の BGP メッセージのペイロード設計と第 7 章でも評価実験に用いる PoC(Proof of Concept) の具体的な実装について述べる．

第 7 章では，第 3 章で述べた課題に対して，本提案手法が有用であることを検証するための実証実験の概要及び具体的なシナリオについて述べ，結果を考察する．

第 8 章では，本研究のまとめと本研究の展望について検討する．

第2章　IPv6シングルスタックネットワークでのIPv4サービス提供手法

本章ではIPv6シングルスタックネットワークでのIPv4サービス提供手法を比較し，検討する．

2.1　概要

コンテンツ事業者が運用するIPv6シングルスタックネットワークの重要な役割の一つに，IPv4クライアントに対するサービス提供がある．

この様な技術に類似したものとして，アクセスネットワーク網ではIPv6シングルスタックネットワーク上でIPv4によるインターネット接続をクライアントエッジに提供する手法がある．IPv4aaS(IPv4 as a Service) と呼称し，様々な手法が検討されている [1].

一方でコンテンツ事業者が運用するネットワークでのIPv4サービス提供においては次項で示す要件を満たす必要があるため，必ずしもアクセスネットワークでのIPv4aaSと同様の方法が適切であるとは限らない．

2.1.1　IPv4サービス提供機構に求められる要件

IPv6シングルスタックネットワークにおいて，サービス提供サーバがIPv4サービスを提供するためには，以下のような機能を有する必要がある．

IPv4クライアントからのアクセス

IPv6クライアントと同様に，IPv4クライアントに対しても透過的にサービスを提供する機構を備える必要がある．一般的なサーバクライアントモデルを想定した場合，インターネット上のIPv4クライアントからサービス提供サーバに能動的に接続するためには，FQDN[1]もしくはIPv4アドレスを指定出来る必要がある．

[1]Fully Qualified Domain Name. 完全就職ドメイン名

スケーラビリティ

近年のコンテンツ事業者のネットワークでは，サービスのニーズに合わせて柔軟にスケールアウト[2]可能な設計であることが重要視されている [2]．同様に IPv4 サービスの提供手法に関しても，事業者の IPv4 サービス規模の変化にあわせて柔軟に拡大・縮小可能なアーキテクチャが求められる．

例えば，第??項でも述べたように，将来的に IPv4 クライアントの占める割合が IPv6 クライアントに相対して低下していった場合に，既設の IPv6 ネットワークへの影響を最小限にしつつ，IPv4 サービス提供機構を縮小可能であることが望ましい．

IPv4 ネットワークへの非依存性

第??項で述べたように，IPv6 シングルスタックネットワークのメリットを最大限に活かすためには IPv4 サービスを提供する場合においても IPv4 ネットワーク及びアドレスに極力依存しないことが望ましい．

2.2 IPv4 サービス提供手法の分類

想定される IPv4 サービス提供機構をその技術的差異や狙いを基に以下の 3 つの手法に分類した．

2.2.1 L7 リバースプロキシ

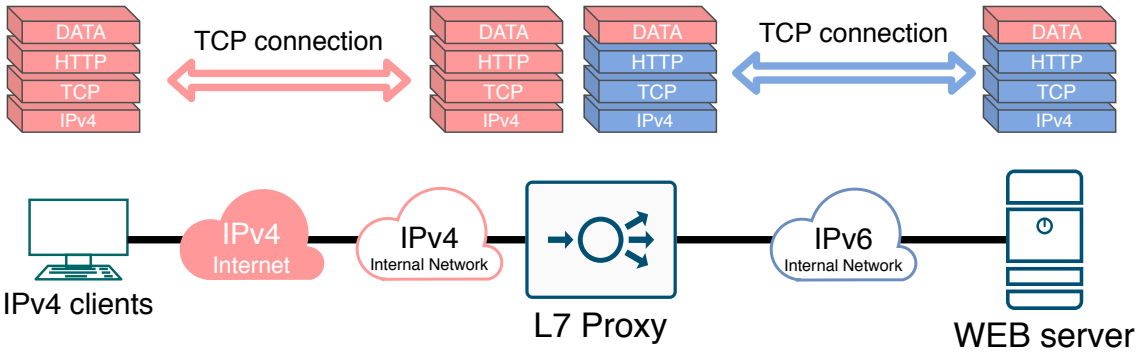

図 2.1: L7 リバースプロキシによる IPv4 サービス提供

L7 リバースプロキシとは，クライアントからの接続をプロキシサーバがアプリケーション層レベルで終端し，プロキシサーバがクライアントに代わってサーバと接続する機構

[2]水平スケール．同等性能の機器を増減させることでサービス容量を拡大・縮小可能なモデル．

である [3]．図 2.1 に本手法の構成を簡便に示す．プロキシサーバを用いる構成は，主に WEB サーバへの HTTP 接続を負荷分散するための手法として広く採用されている．

IPv6 シングルスタックネットワークにおいて IPv4 サービスを提供するためには，IPv4 インターネットとの接続点からプロキシサーバまでの間に IPv4 ネットワークを配備する必要がある．

IPv4・IPv6 間のプロトコル仕様の差を考慮する必要が無いため互換性に留意する必要が無い点や，MTU[3]を減らさずにアプリケーショントラフィックを伝送可能である点が利点として挙げられる．

一方でアプリケーションレイヤーでのコネクション終端やそのステート管理を行う必要があるため，プロキシサーバに負荷が掛かるため高性能な機器の導入が必要になる．

またスケールアウトを可能にするために L4LB(Layer4 Load Balancer)[4]と組み合わせたマルチステージのアーキテクチャを利用する手法が近年主流であるが [4, 5]，この手法を採用するためには，L4LB 及びプロキシサーバにまで IPv4 ネットワークを配備する必要があり，2.1.1 項で述べた要件に合致せず，IPv6 シングルスタックネットワークのメリットを損なうことになる．

2.2.2 IPv4/IPv6 トンネリング

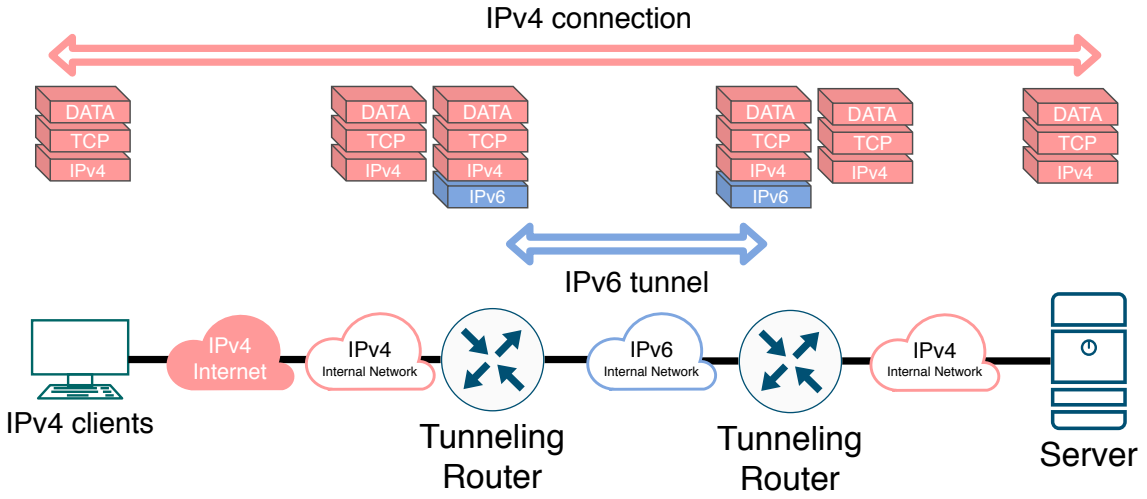

図 2.2: IPv4/IPv6 トンネリングによる IPv4 サービス提供

IPv4/IPv6 トンネリングとは，IPv4 パケットを IPv6 パケットによってカプセリングすることで IPv6 ネットワークを通過させる手法である．IPv4 トラフィックを透過的に利用することが出来るため，アクセスネットワークでの IPv4aaS 実現手法として広く利用されている [1]．図 2.3 に本提供手法の構成を簡便に示す．

[3] ここでは一つのパケットに収容可能なデータ量を指す．
[4] トランスポート層レベルでの負荷分散を行う機器．

IPv6 シングルスタックにおける IPv4 サービス提供手法としては，IPv4 クライアントから到達したパケットをトンネルルータによって一度 IPv6 パケットでカプセリングし，IDC 内の IPv6 シングルスタックネットワークを通過させ，IPv4 サービス提供サーバ上もしくはその直前で再びでカプセリングを解くことで，IPv4 提供サーバまでネイティブな IPv4 トラフィックを通過させる運用が考えらえる．IPv4 パケットをそのままサーバまで届けることが出来るため，多種多様なアプリケーションでのサービス提供が可能である．

しかしながら，トンネルルータと IPv4 サービスサーバ間に IPv4 ネットワークを配備しなければならず，ToR(Top of rack switch)[5]及びサーバでは IPv4/IPV6 デュアルスタック運用が必要になるため，第 2.1.1 項で上げた要件である「IPv4 ネットワークへの非依存性」を満たすことが出来ない．また，トンネルプロトコルの多く [6] は基本的に 1:1 もしくは 1:N の接続が基本となるため，スケールアウトさせることが困難である．

2.2.3 IPv4/IPv6 トランスレーション

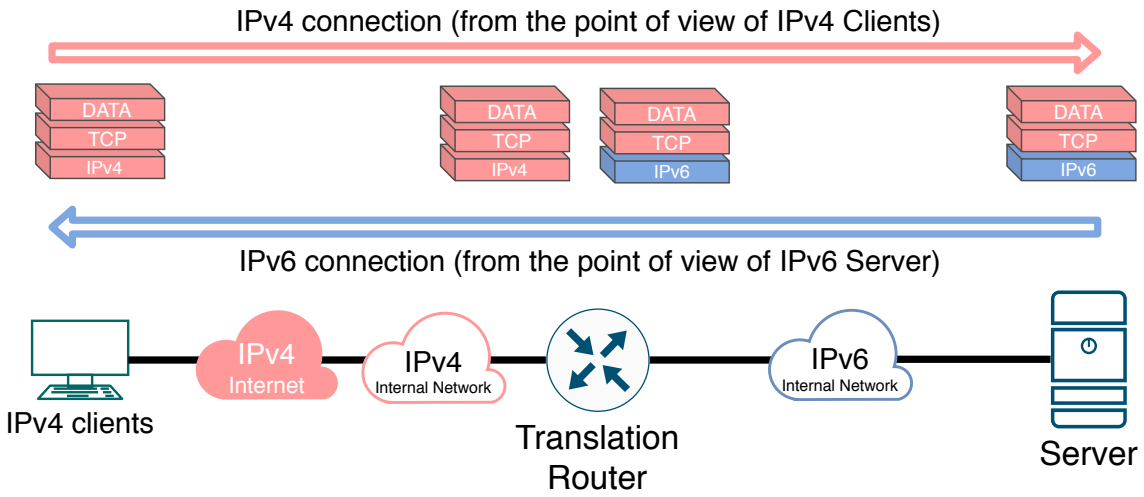

図 2.3: IPv4/IPv6 トランスレーションによる IPv4 サービス提供

IPv4/IPv6 トランスレーションとは，IPv4 パケットと IPv6 パケットを IP/ICMP 変換アルゴリズムを利用して相互に変換する手法である．$1:N$ の関係でアドレス・ポート変換を行うステートフルな NAT64[7] と，$1:1$ でアドレス変換を行うステートレスな SIIT[8] が定義されている．IPv4 ネットワークと IPv6 ネットワークの境界に位置する変換ルータにより，相互にプロトコル変換が行われる．

IPv4/IPv6 トランスレーションでは IPv4 アドレスを IPv6 アドレスとして表現することが要求されるが，変換プレフィックスと呼ばれる IPv6 ネットワークプレフィックスに IPv4 アドレスを埋め込むことで，任意の IPv4 アドレスを IPv6 ホストから認識可能な形で表現する．変換プレフィックスには RFC6052 で定義された 64:ff9b::/96 の他に，運用者

[5] ここではサーバの L2 終端を行うイーサネットスイッチを指す．

が専有可能な GUA(Global Unicast Address) の/96 の IPv6 プレフィックスを利用することが想定されている [9].

図 2.3 で示すように，変換ルータ以外のホストが IPv4 ネットワークに属する必要が無いため，第 2.1.1 項で述べた「IPv4 ネットワークへの非依存性」の面で，他の 2 手法より優れていると言える．また，IPv4 サービスを行うサーバから変換ルータの間はネイティブな IPv6 ネットワークで接続可能なため，ECMP(Equal Cost Multi Path)[10] による経路の冗長化が可能なほか，ステートレスモードでは変換ルータのスケールアウトが可能な点で，IPv6 シングルスタックネットワークにおける IPv4 サービス提供に求められる要件を満たしやすい.

またサーバ側においても，アクセスコントロールの設定やログデータの管理など，IPv4 と IPv6 で別個に行われていた制御の削減が期待される．サーバ・アプリケーション管理者はネットワークプロトコルを意識したオペレーションをする必要がなく，運用効率改善の点でメリットが非常に大きい.

一方で IPv4 と IPv6 のプロトコル実装に差があるため，コンテンツ事業者のサービスの内容によってはサービス影響を考慮する必要がある点は留意すべきである.

2.3 本章のまとめ

2.1.1 項で述べた評価要件を尺度として，各 IPv4 サービス提供手法の特徴を比較した．これを表 2.3 に示す.

基本的な IPv4 サービス提供手法の必要要件である IPv4 クライアントからのアクセスは各手法とも充足可能な一方，スケーラビリティの面では 1:1/1:M の接続が必須になる IPv4/IPv6 トンネリング手法が他の手法から大きく劣る．また L7 リバースプロキシ手法はスケールアウト可能なアーキテクチャであるが，L4LB 及びプロキシサーバ L まで IPv4 ネットワークを配備する必要があるため，IPv4 ネットワークへの依存性が大きい.

IPv4/IPv6 トランスレーション手法はスケールアウトが可能であり，IPv4 ネットワークへの依存性が小さいため，IPv6 シングルスタックネットワーク環境での IPv4 サービス提供手法として最も要件に適合する.

表 2.1: IPv4 サービス提供手法の比較

手法名	IPv4 クライアントからのアクセス	スケーラビリティ	IPv4 ネットワークへの依存性
(参考)IPv4/IPv6 デュアルスタック	可能	困難	大
L7 リバースプロキシ	可能	スケールアウト可	中
IPv4/IPv6 トンネリング	可能	困難	中
IPv4/IPv6 トランスレーション	可能	スケールアウト可	小

第3章　SIIT-DCのデザインと現状の課題

第2.2.3項で述べたPv4/IPv6トランスレーションを用いたIPv4サービス提供手法の一つとして，SIIT-DCがインターネット標準化されている．本章ではSIIT-DCのデザインとメリット及び考えられる運用，そして現状の課題について述べる．

3.1　SIIT-DC

3.1.1　概要

SIIT-DCとは，ステートレスIP/ICMP変換アルゴリズム[8]を利用して，IPv4インターネット・ネットワークからのアクセスをIPv6シングルスタックネットワーク上のホストに提供するためのネットワークデザインである．2016年にIETF IPv6 Operations WG[1]での議論を基にRFC7757として標準化された[11].

3.1.2　用語

SIIT-DCで利用される用語，及び特徴的な役割を有する機器・技術について述べる．

SIIT(Stateless IP/ICMP Translation Algorithm)

SIITとはIPv4/IPv6トランスレーションに用いられるプロトコル変換機能の略称である．RFC2765[12]で初めて標準化され，その後RFC6145[13]により一部の仕様が実運用のユースケースに合わせて変更された．現在はIPv6拡張ヘッダーを扱う機構などが追加されたRFC7915[8]が現行の標準仕様である．

BR(Border Relay)

BRとは，SIIT-DCネットワークにおいてIPv4インターネットとIPv6ネットワークとの間でSIITによるIPv4/IPv6トランスレーションを行う機器もしくは他の役割を有する

[1]IPv6 ネットワークの運用要件や関連する技術仕様の策定を行うワーキンググループ．https://datatracker.ietf.org/wg/v6ops/about/

機器の一機構である．IPv4 インターネットと IDC 内の IPv6 シングルスタックネットワークの各境界部に所在し，後述する EAMT を参照した 1:1 のアドレス変換を行う．IDC ネットワークに IPv4 インターネットとの接続点が複数ある場合，接続点ごとに最低一つの BR を配備する．

ER(Edge Relay)

ER とは，IDC 内の IPv4 ネットワークと IPv6 ネットワーク間での多：多の IPv4/IPv6 トランスレーションを行う機器である．

SIIT-DC ではそのオプションとして，IPv4 ネットワーク内の IPv4 しか利用出来ないホストが，SIIT-DC を利用して IPv4 サービスを提供するユースケースをサポートする SIIT-DC Dual Translation Mode[14] が定義されており，ER はその中での利用が想定されている．

通常，ER が有する後述の EAMT は IDC ネットワーク内の IPv4 ネットワークアドレスと，その IPv4 ネットワークを示す IPv6 サービスアドレスにより構成される．

IPv4 サービスアドレス

IPv4 サービスを提供する IPv6 シングルスタックネットワークに属するホストに割り当てる IPv4 アドレス (群) を IPv4 サービスアドレスと呼称する．このアドレス宛に送信されたパケットは，BR/ER によって対応する IPv6 サービスアドレスに変換される．

なお，IPv4 サービスアドレスは IPv4 インターネットに経路広告されている必要がある．

IPv6 サービスアドレス

ER/BR を介してアプリケーションやホストに割り当てられた IPv6 アドレス (群) を IPv6 サービスアドレスと呼称する．IPv4 クライアントは SIIT-DC のアーキテクチャをを通じて，この IPv6 サービスアドレスが割り当てられたホストと通信することが出来る．

変換プレフィックス

変換プレフィックス (Translation Prefix) とは，全ての IPv4 アドレスをマッピングするために用いられる，プレフィックス長が 96bit の IPv6 ネットワークプレフィックスである [9]．IANA によって主に WKP(Well Known Prefix) として 64:ff9b::/48 が予約 [15, 16] されているが，運用者の裁量で ISP 自身に割り当てられた NSP(Network Specific Prefx)[2]を利用する事ができる．

IPv4 アドレスと IPv6 アドレスの間で変換を実行する際に，BR/ER は変換前の IP ヘッダーのアドレスフィールドを、変換プレフィックスが挿入・削除された状態に書き換える。

[2]主に RIR から割り当てられた IPv6 Global Unicas Address を指す.

なお SIIT-DC ネットワークにおいて，変換プレフィックス宛のパケットは各 BR/ER の IPv6 インターフェース宛に IGP(Interior Gateway Protocol) などで経路広告される必要がある．

EAM(Ecplicit Address Mapping)/EAMT(Ecplicit Address Mapping Table)

EAM とは，EAM アルゴリズム [17] によって結びつけられた IPv4 サービスアドレスと IPv6 サービスアドレスのペアを表す．

EAM において，それぞれ同数の IPv4 サービスアドレスと IPv6 サービスアドレスによって構成される．標準では結び付けられた IPv6 サービスアドレスが IPv4 サービスアドレスより多い状態が想定されているが，IPv6 サービスアドレスのホスト部が若いものから優先して変換するため，余剰分のアドレスは無視される．

また，BR 及び ER が変換を行う際に参照する EAM 群が記録されたテーブルを EAMT と定義している．以後 EAMT もしくは変換テーブルと呼称する．

3.1.3 ネットワーク設計

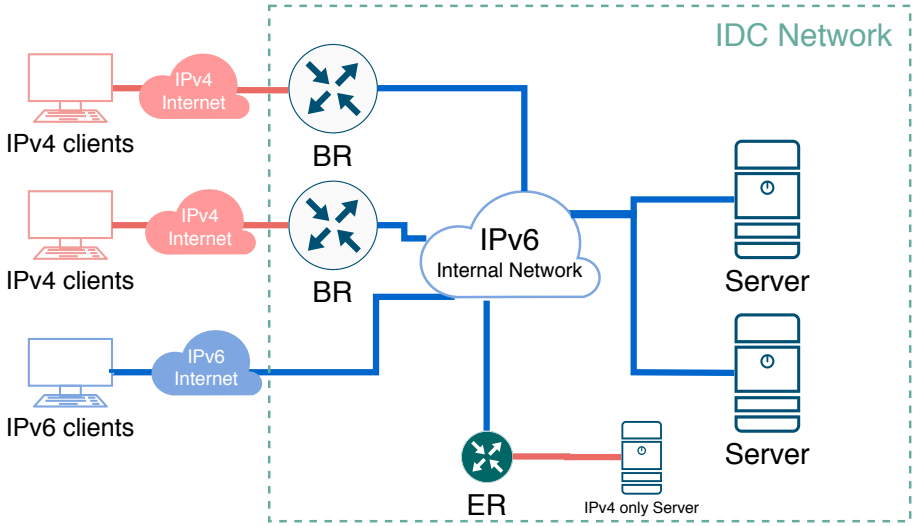

図 3.1: SIIT-DC ネットワーク

基本的な SIIT-DC ネットワークを図 3.1 に示す．

BR は IPv4 インターネットとの各接続点に配置される．各 IPv4 サービスアドレスは自組織のアドレスとして，IPv4 インターネットに経路広告される必要がある．

SIIT-DC ネットワークでは，変換プレフィックス宛のパケットは BR に対してルーティングされる．BR が複数ある場合，BR がネットワークプロトコルに利用する変換プレフィックスを別個に用意するか，同一の変換プレフィックスを各 BR にエニーキャスト [18] によっ

てルーティングさせる．エニーキャストを使用した場合，BR の障害時には別の BR へとトラフィックを迂回させることが可能である．

ER は IDC 内の IPv4 ネットワークとの接続点に配置され，IPv4 のみを持つホストが IDC 内の IPv6 ネットワークを介して IPv4 インターネットにサービス提供を行う場合に利用される．

3.1.4 SII-DC のメリット

SIIT-DC を用いた IPv4 サービスの提供によるメリットとして，以下の点が挙げられる，

デプロイメントが容易

SIIT-DC では，IDC の IPv6 ネットワークと IPv4 インターネットとの接続点に BR を設置を行うのみにより，基本的な IPv4 サービスの提供が可能である．そのため IDC のネットワークトポロジに限定されないシンプルな IPv4 サービス提供が期待できる．

アドレス単位での IPv4 アドレスの効率的な利用が可能

通常の IP ネットワークにおいて，サーバに対する IP アドレスアサインメントはサブネット単位での割り当てを行う必要がある．従来，事前に同一サブネットに属するホスト数を見積持った上で不足が生じないようにネットワークサイズを設定する必要があるめ，ネットワークサイズを超えるサービスの拡大が必要になった場合，サブネット全体の再設計が不可欠であった．また IP ネットワークには，ネットワークアドレスやブロードキャストアドレス，そしてデフォルトゲートウェイとなるルータのインターフェースのアドレスを確保する必要があり，ネットワークサイズが断片化されるほど，実質的に利用できないアドレスの割合が大きくなる問題が会った．

しかしながら SIIT-DC ではサーバごとにアドレスを割り当てることが出来るため，従来利用できなかった IPv4 アドレスを再利用することで，IPv4 アドレスの効率的な利用を実現できる．第??項で述べたように今後益々IPv4 アドレスの調達が困難になることが予想されるため，IPv4 アドレスの効率的な利用は事業者の負担軽減に繋がる．

スケーラビリティ

SIIT-DC の標準仕様 [11] では明示的に述べられていないが，本論文では BR を並行して複数配置することで，スケールアウトが可能なネットワークデザインを立案する．本ネットワークデザインでは，ECMP 及びエニーキャスト [18] を利用することにより，BR の数を水平に増加させることで，IPv4 サービスの提供容量をリニアに増加させることが可能である．

図 3.2 に本ネットワークデザインに則って BR 配置を行ったスケーラブルな SIIT-DC ネットワークの例を示す．IPv4 クライアントからのアクセスはいずれかの BR にフォワー

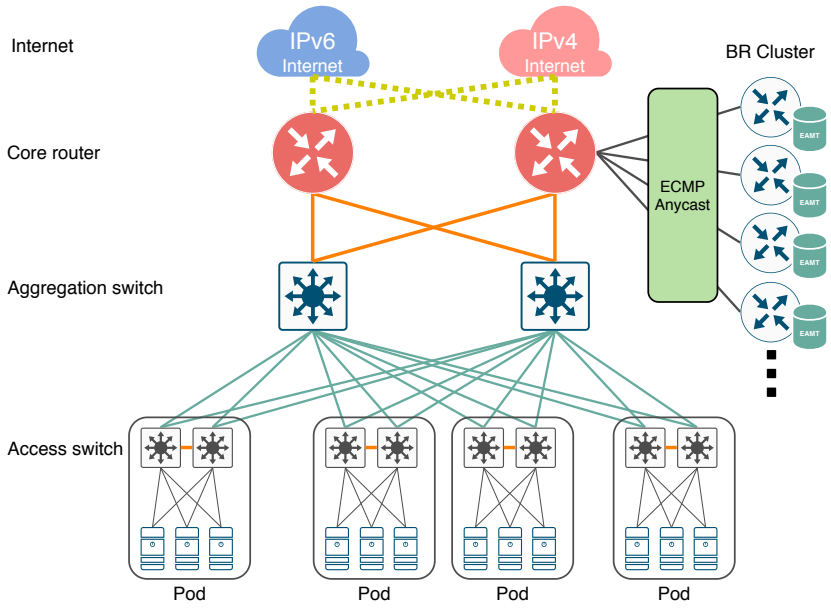

図 3.2: BR を水平スケールすることが出来る SIIT-DC ネットワーク

ディングされた後，IPv6 プロトコルに変換された上で再度コアルータを介して IDC ネットワーク内の IPv4 サービス提供サーバに到達する．

3.1.5 基本的なパケットの流れ

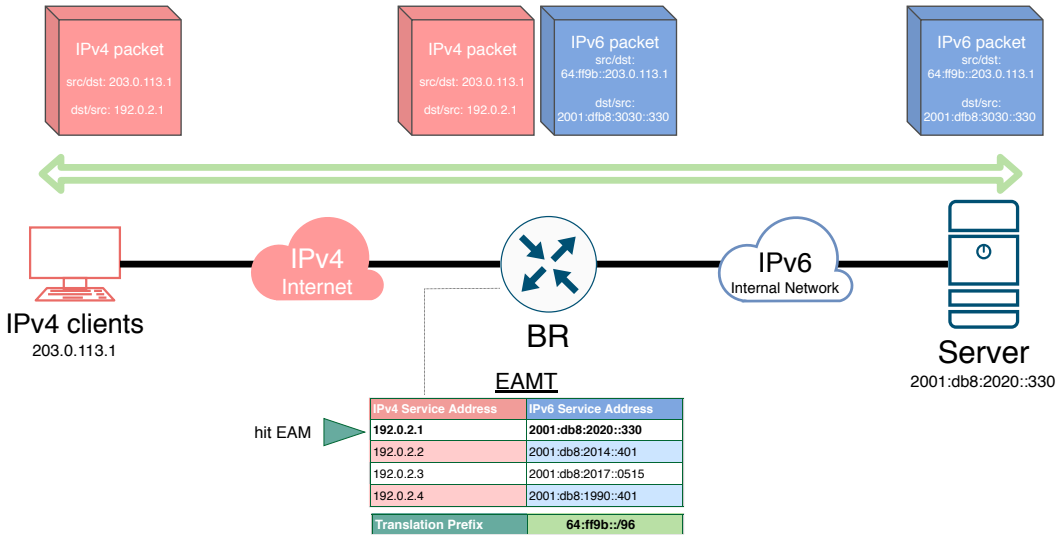

図 3.3: SIIT-DC パケットの流れ

SIIT-DC における基本的な IPv4 クライアントからのトラフィックの流れは以下の様になる．一連のパケットの送信元・送信先のアドレスの遷移を図 3.3 に示す．

IPv4 クライアントの IPv4 サービスアドレス宛のパケットは IPv4 インターネットに接続する BR に到達後，当該 BR が有する EAMT に従って IPv6 サービスアドレス宛の IPv6 パケットに変換される．このパケットの送信元アドレスは変換プレフィックスに埋め込まれた IPv6 アドレスとして表現される．IDC 内の IPv6 ネットワークを介して IPv6 サーバに到達した後，IPv6 サーバは送信元アドレスへの応答パケットを送信する．3.1.2 項で述べたように，変換プレフィックス宛のパケットは IPv6 ネットワークを経由して BR にルーティングされる．IPv6 サーバからの応答を受け取った BR は EAMT を参照し，送信元アドレス (IPv6 サービスアドレス) を IPv4 サービスアドレスに書き換え，送信先アドレス (IPv4 クライアントの IPv4 アドレス) から変換プレフィックスを除去書き換えたのち，IPv4 インターネットを介して IPv4 クライアントに返送される．

3.2 SIIT-DC の課題

本節では SIIT-DC の現状の課題及びそれに起因して起こる事象に関して述べる．

3.2.1 一貫した EAMT の必要性

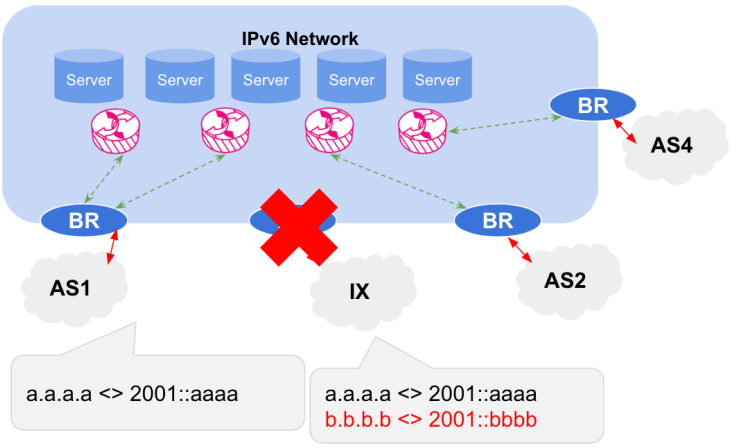

図 3.4: BR に障害が発生した場合に適切にフェイルオーバーが出来ないケース

3.1.2 項で述べたように，SIIT-DC では対外接続点ごとに BR を配置するネットワークデザインを採用することで，IPv6 シングルスタックネットワークに最小限の IPv4 ネットワークを追加するだけで IPv4 サービスの提供を可能にしている．また 3.1.3 項や 3.1.4 項で触れたように，複数の BR で共通した変換プレフィックスをエニーキャストで IDC ネットワーク内に広告する運用を行うことにより，BR 及び対外接続点の障害時に他の BR を

用いて IPv4 サービスの提供を継続することが出来る．この機構を有効に作用させるためには，SIIT-DC ネットワーク内の全ての BR で一貫した EAMT の保持が求められる．

しかしながら現状の SIIT-DC 及び EAMT の仕様 [11, 14, 17] では，BR は他の BR との間で EAMT を共有するためのメッセージング機構を有さない，これは BR 間で EAMT の不一致が発生した場合に，差異となった EAM に該当する IPv4 サービス宛のトラフィックを別の BR へ迂回出来なくなるケースが発生することを意味する．

3.2.2 変更追従性の欠如

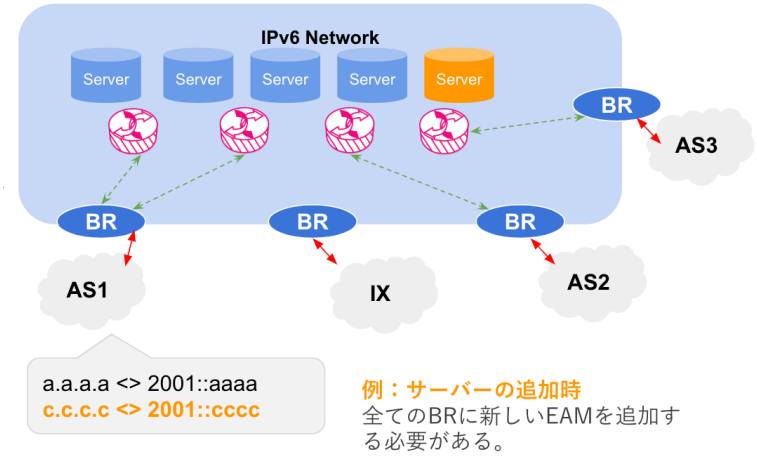

図 3.5: サーバを追加した際，全ての BR への設定追加が必要になる．

プライベートクラウド環境が一般的に利用される IDC ネットワークでは，日々多くのサーバやアプリケーションが追加・廃止・変更される．一方で 3.2.1 項で触れたように，SIIT-DC で IPv4 提供サービスを冗長に運用するためには，IPv4 提供サービスに該当する EAM が BR の EAMT に保持されることが要求される．IPv4 提供サービスの構成に変更があった場合，全ての BR の EAMT を更新する必要がある．

しかしながら現状 SIIT-DC 及び EAMT の仕様 [11, 14, 17] において，IPv4 サービスを行うサーバの存在や状態によってダイナミックに EAMT を更新する機構は存在しない．そのため，IDC ネットワークにおける IGP などによって IPv6 サービスへの到達性が検証されていたとしても，IPv4 サービスの場合はリアルタイムな構成変更に追従することが出来ない．

3.3 本章のまとめ

第 3.2 項で述べたように，現状の SIIT-DC 及び EAMT の仕様は EAMT の一貫性を担保する手法の検討がなされておらず，それに起因した障害時の適切なフェイルオーバーの

実行や IPv4 サービスの増減時の変更追従に関しての課題がある．NPO 日本ネットワークセキュリティ協会 (JNSA) らの調査によれば IT システムの障害の原因の約半数は人為ミスに分類されるものにあり [19]，サービスの安定的な稼働を実現するためには単調な繰り返し動作を含む運用をシステムによって減らす必要がある．

第4章　手法の検討

4.1　概要

本研究ではSIIT-DCにおける動的なEAMTの管理・制御手法の実現を目指す．以後このような機構をダイナミックEAMT機構と呼称する．

本章では考えられる手法を大別した上でその特徴と利点及び欠点を挙げ，最も適した手法を検討する．

4.2　求められる要件

前で述べたIPv4サービス提供手法の機能要件とSIIT-DCの現状の課題を総括し，EAMTを動的に制御する手法に求められる要件を下記のように定義した．

1. **BR間のEAMTの一貫性**
   障害時の適切なフェイルオーバを実現するためには，ネットワーク内の各BRが有するEAMTの一貫性が保証される必要がある．

2. **変更追従性**
   近年のIDCでは多数の物理サーバーを統合的に管理するプライベートクラウド環境やコンテナオーケストレーション環境[1]が普及しており，アプリケーション・サービスの追加及び削除が頻繁に行われている．サービスの障害を検知し，適切に冗長系に移行するための手法として，SLB(Servver Load Balancer)が広く利用されている．SIIT-DCのIPv4サービス提供の場でも，サービスの状態の変動にリニアに対応しフェイルオーバーできるような働きが求められる．

3. **スケーラビリティ**
   IPv6シングルスタックネットワークにおけるIPv4サービスの提供では水平スケールが容易に行える仕組みを備える必要がある．IPv4サービスを行うサーバの増設や，対外接続点が増えた場合のBRの拡大に十分に適用するスケーラビリティを有することが望ましい．

4. **デプロイメントの容易さ**
   SIIT-DCの最も特筆すべきメリットの一つにデプロイメントの容易さが挙げられる．これを損なうことくダイナミックEAMTを実現する必要がある．

[1]Container Orchestration. コンテナ型仮想化統合管理環境

4.3 アプローチの分類と比較

はダイナミック EAMT を実現するアプローチとして，二つのアプローチを考察する．それぞれのアプローチで考えられる実装と実際の構成，及び第 5.5 節で述べた各要件への適合性を定性的に評価する．

本節ではスケーラビリティの評価のために，制御に必要な通信コネクション数による比較を行う．以後 BR の数を M，IPv4 サービスを提供するサーバの数を N とし，総通信コネクション数を C として表現する．

4.3.1 中央管理型アプローチ

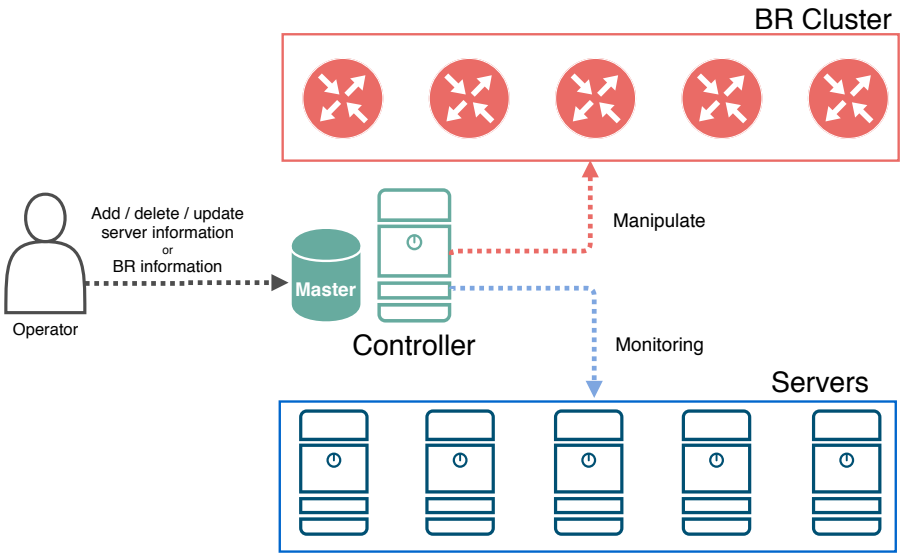

図 4.1: 中央管理型アプローチによるダイナミック EAMT

中央管理型アプローチとは，複数の BR の EAMT を統合的に管理する「コントローラ」を IDC ネットワーク上に配置し，各 BR がネットワークを介してこれを参照する機構である．図 4.1 に中央管理型アプローチによってダイナミック EAMT を実現した SIIT-DC の各コンポーネントの関係図を示す．

中央管理型アプローチではコントローラが各 BR に投入する EAM が記録された「マスターテーブル」を保持し，それを元に各 BR のデータプレーンにルールを書き込む手法を取る．マスターテーブルに記載される EAM はオペレーターがネットワークの構成変更に合わせて追加・削除・更新を行い，それぞれの IPv4 サービスを提供するサーバ群に対してはコントローラからプル型[2]の外部監視[3]によりサーバの状態変化を検知しマスターテーブルを更新する．

[2]pull-based monitoring. コントローラから各サーバに能動的に情報を取得する

[3]External monitoring. 監視対象でエージェントを稼働することなく，外部から得られる情報を利用して監視を行うこと．

本アプローチの実装手法としては，OpenFlow[4]などを用いた集中コントローラ型 SDN フレームワークを利用する方法が考えられる [20]．類似事例として，Sheng らによって Open Flow を利用して各アクセススイッチに IPv4/IPv6 トランスレーション機構をデータプレーンとして導入するデータセンターネットワークデザインの提案がなされている [21]．

要件評価

- BR 間の EAMT の一貫性
  本アプローチでは各 BR の EAMT が一つのマスターテーブルからレプリケーションされるために，十分な一貫性が保証される．
- 変更追従性
  基本的には EAM 情報の更新はオペレーターのマスターテーブルへの記入までの時間はコントローラのサーバ監視性能に依存する．
- スケーラビリティ
  コントローラの数を L とすると，EAMT の制御に必要とする総通信コネクション数 C は以下の通りになる．
  $$C = L(M + N) \tag{4.1}$$
  一方，変更追従性と同じく，管理対象のサーバの収容台数に関しては，コントローラの実装・性能がボトルネックとなる設計である．
- デプロイメントの容易さ
  コントローラに求められる機器の性能・機能要件が大きいため，標準的な SIIT-DC よりデプロイメントのコストは高い．

4.3.2 分散管理型アプローチ

分散管理型アプローチとは，IPv4 サービスを提供するサーバがエージェントプロセスを介して自身の IPv4 サービスアドレスと IPv6 サービスアドレスを広告し，その広告情報を受け取った BR が自身の EAMT に反映させる機構である．図 4.2 に中央管理型アプローチによってダイナミック EAMT を実現した SIIT-DC の各コンポーネントの関係図を表す．

サーバ群は各 BR と EAM を広告するためのコネクションを確立する．IPv4 サービスを提供するサーバと BR の間の IP ネットワークが何らかの原因により疎通不能になると，当該サーバの広告も同時に停止されるため，該当 BR の EAMT から該当する EAM のレコードが削除される．

[4]Open Networking Foundation により標準化されているデータプレーン制御用通信プロトコル．`https://www.opennetworking.org/`

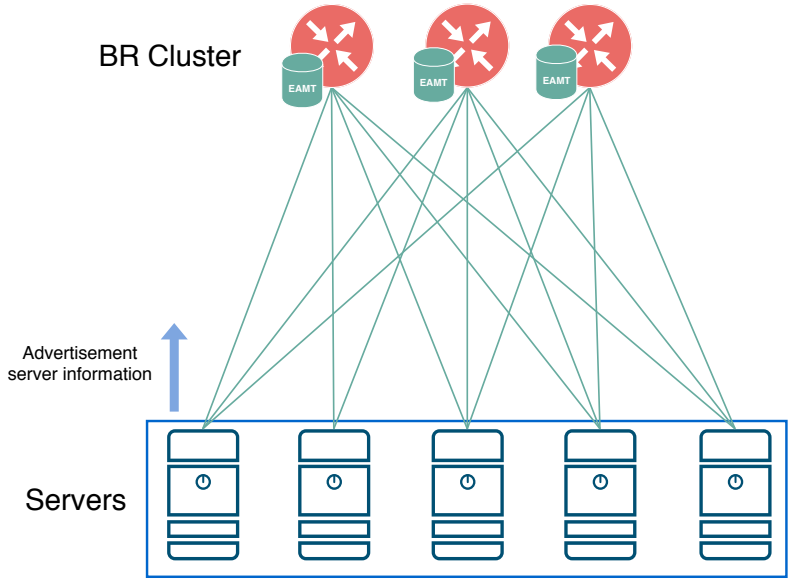

図 4.2: 分散管理型アプローチによるダイナミック EAMT

要件評価

- BR 間の EAMT の一貫性
  各 BR 間で EAMT の一貫性を保証する機構は無いが，当該 BR と疎通できないサーバは障害時に自身の IPv4 サービスアドレス宛のトラフィックを当該 BR に経由させることが出来ないため，問題にならない．

- 変更追従性
  サーバ自身のエージェントプロセスが直接 BR に広告を行うため，実際の変更にリニアに対応出来る．

- スケーラビリティ
  EAMT の制御に必要とする通信コネクション数 C は以下の通りになる．

$$C = M \cdot N \tag{4.2}$$

  サーバ群・各 BR 間でフルメッシュでのコネクションが必要なため，SIIT-DC ネットワーク自体が小規模の場合のみ採用可能である．

- デプロイメントの容易さ
  各サーバ・BR にエージェントを導入する必要があるが，システム自体の機能は軽量である．

表 4.1: 各アプローチの比較

手法	EAMT の一貫性	変更追従性	コネクション数	デプロイメントの容易さ
オペレーターによる手動設定	無し	無し	——	——
中央管理型アプローチ	有り	(監視機構の実装依存)	$\frac{L(2M+2N+L-1)}{2}$	困難 (コントローラーの実装依存)
分散管理型アプローチ	無し	有り	$M \cdot N$	有り

4.4 アプローチの検討

表 4.1 に 5.5 で述べたダイナミック EAMT に求められる要件に関する両アプローチの比較を示す．中央管理型アプローチが各 BR 間での EAMT の一貫性，スケーラビリティの二要素で優位であるが，コントローラの役割が非常に大きくなり機能要件が高くなるため，変更追従性とデプロイメントの容易さの面での障壁が高いという問題を抱えている．一方で分散管理型アプローチはシンプルな構成であるためデプロイメントが比較的容易であり変更への追従がリニアであるが，各サーバが通信コネクションを多量に貼らなくてはならない点でスケーラビリティに難がある．

第5章　ダイナミックEAMT実現手法の設計

第4章では，ダイナミックを設計する上で考えられる二種類のアプローチについて，求められる要件に照らし合わせて評価・検討を行った．本章では検討結果の得られた内容を基に，本研究において提案するダイナミック EAMT の実現手法の設計に関して論じる．

5.1　概要

第 4.4 で述べたように，分散管理型アプローチと中央管理型アプローチの双方に優位点があり，ネットワークやサービスの規模に合わせて柔軟に選択可能であると望ましい．

本研究では，動的経路制御プロトコルである BGP を利用したサーバ・BR 間のメッセージングにより，SIIT-DC ネットワークにおけるダイナミック EAMT 機構を提案する．本提案手法は，IBGP(Internal BGP)・RR(Route Reflector) 構成を採用することで，ネットワークやサービスの規模に合わせてスケールアウトすることが可能であり，両アプローチの優位点を備えていると言える．

5.2　BGP

5.2.1　概要

BGP とはインターネットにおいて自律システム[1]間の経路情報交換に用いられるパスベクタ型の動的経路制御プロトコルである．現在有効なバージョンは BGP4 であり，RFC4271 で定義されている [22].

5.2.2　用語

BGP において利用される用語のうち，本提案手法において重要なものを以下に列記する．

[1]Autonomous System. インターネットを構成するネットワークをそれぞれ独立的に運用する組織群を指す.

BGP スピーカ

BGP を実装された機器を BGP スピーカと呼ぶ.

BGP ピア

BGP で経路交換を行う関係にある機器をぞれぞれ BGP ピア (BGP Peer) と呼称する.

そのうち，自律システム間での接続関係にあるピアを EBGP(External BGP) ピア，同一自律システム内の BGP スピーカ同士の経路交換に用いられるピアを IBGP(Internal BGP) ピアと呼ぶ.

BGP コネクション

BGP コネクションとは BGP で経路交換に用いられる接続関係を指す．各機器は 1 対 1 の関係で BGP コネクションを確立する．BGP コネクションにはトランスポート層のプロトコルとして TCP[23] のポート番号 179 が利用され，フラグメンテーションや再送制御，応答確認，誤り制御等，TCP による高信頼なメッセージングが可能である．

また，BGP コネクションを維持・管理するために，BGP では以下のような 4 つのメッセージが定義されている．

BGP コネクションは BGP ピア間で TCP コネクションを確立したのちに OPEN メッセージにより各機能の対応関係を確認することにより確立され，KEEPALIVE メッセージによりセッションが維持される．UPDATE メッセージにより，BGP ピアへ広告する経路 (Adj-RIB-Out) に変更が生じたことを通知する．何らかの理由により BGP コネクションが確立出来なくった場合，NOTIFICATION メッセージを利用して切断を通知する．

Adj-RIB-In/Adj-RIB-Out/Loc-RIB

図 5.1 に BGP における経路受信・保持・送信の流れを示す．BGP ピアから受信した経路は Adj-RIB-In と呼ばれ，BGP スピーカの任意のフィルターやポリシーを適用した上で Loc-RIB と呼ばれるテーブルに保存される．BGP スピーカは Loc-RIB から任意のフィルターを適用した経路を BGP ピアに広告する．この広告する経路を Adj-RIB-Out と呼ぶ.

5.2.3 特徴

本提案手法においてダイナミック EAMT を実現するためのメッセージングプロトコルとして BGP を選択するに至った要素について述べる.

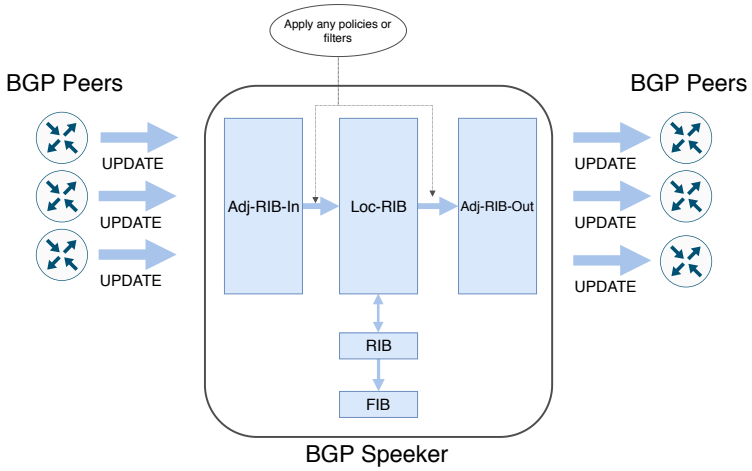

図 5.1: BGP スピーカの経路の扱い

マルチプロトコル

現行版である BGP4 では，OPEN メッセージにオプション値 (Capabilities Optional Parameter) を挿入することで，IANA によって定められた任意のネットワークプロトコル [24, 25] の経路を交換することが想定されている [26]．本提案手法で利用している IPv6 ユニキャスト経路もこの機構を用いる．

実装が一般的

BGP は自律システム間の経路交換プロトコルとしてインターネットで利用されているデフファクトスタンダードなプロトコルであり，OSS(Open Source Software)[2]にも多くの実装が存在する．広く普及したプロトコルを利用することにより，特別な実装を最小限にして本提案手法を実現することが出来る．

中継ネットワークに非依存

本提案手法で採用している IBGP では，TTL(Time to Live)[3]が 255 に設定されており，BGP ピア間で IPv4/IPv6 による到達性があればメッセージングを行うことが可能である．すなわち本提案手法は既存の SIIT-DC ネットワークに非依存であり，これは第 5.5 項で述べた要件の一つである，デプロイメントの容易さを充足する．

[2]ソースコードが公開されており，定められたライセンス規約に基づく範囲で自由に使用・改造が可能なソフトウェア．

[3]そのパケットが宛先ホストに到達するまでに許容される中継ルータ数．IPv6 プロトコルでは Hop Limit として同一の機能が実装されている [27]．

5.3 基本的なネットワーク設計

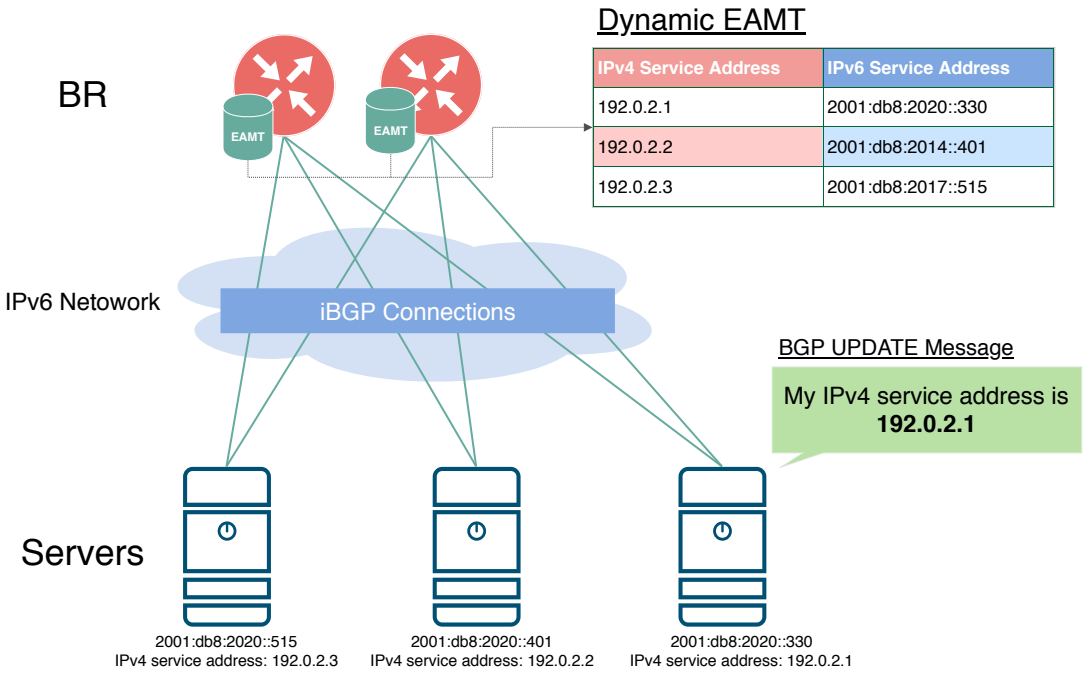

図 5.2: 本提案手法の基本機能を実装した SIIT-DC ネットワークの例

図 5.2 本提案手法の各要素の関係を示す.

BR 数を N，サーバ数を M とした，ルートリフレクタを利用した本提案手法での必要な BGP コネクション数 C_a は式 5.1 のように表現できる.

$$C_a = M \cdot N \tag{5.1}$$

5.3.1 各ノードの役割と機能要件

BR

BR では下記のような 3 つの機能が必要となる.

- BGP デーモン
  各サーバと IBGP コネクションを確立し，Loc-RIB を生成する.
- SIIT 機構
  EAMT を保持し，それを参照して IPv4/IPv6 プロトコル変換を行う.
- EAMT 制御機構
  BGP デーモンが有するの Loc-RIB を参照し，EAMT を更新する.

IPv4 サービス提供サーバ

IPv4 サービス提供サーバでは以下の 2 つの機構が求められる．

- IPv4 サービス
  IPv4 によりインターネットに提供したいサービスを稼働させる．
- BGP デーモン
  自身が提供する IPv4 サービスアドレスを含んだ情報を広告する．

5.4 ルートリフレクタを活用したネットワーク設計

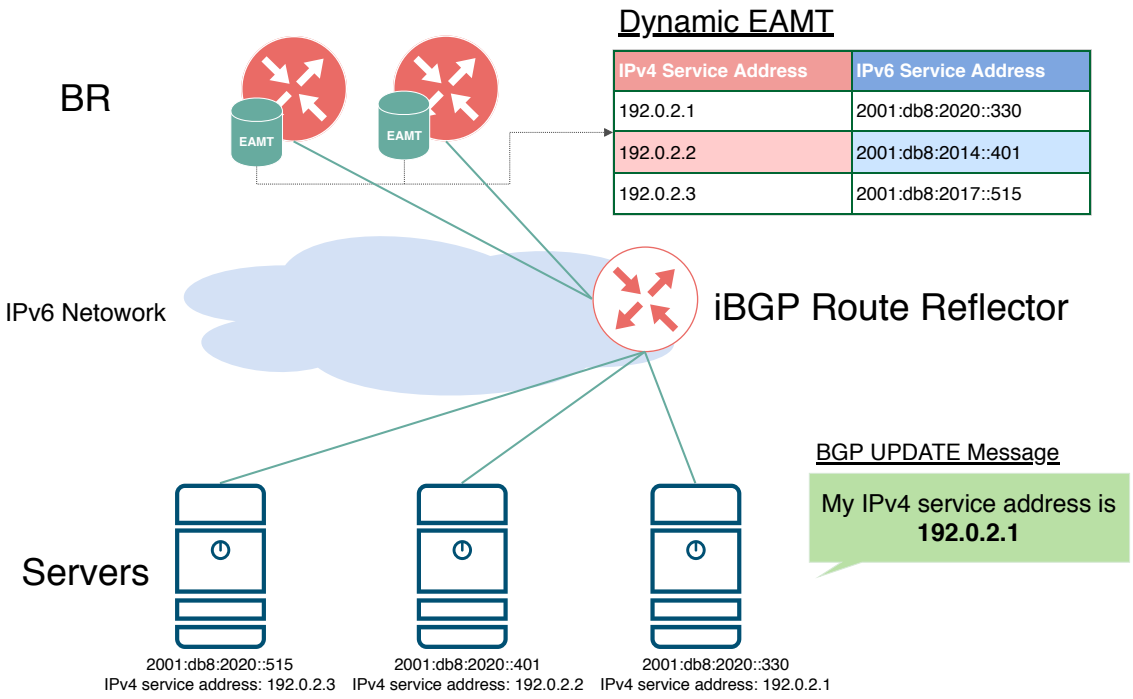

図 5.3: ルートリフレクタを採用した SIIT-DC ネットワークの例

通常，IBGP ではルートループを防ぐために異なる BGP ピアから受信した経路は他の BGP ピアに広告されない．そのため一つの IBGP スピーカが広告する経路を他の IBGP スピーカが受信するためには，BGP コネクションをフルメッシュで確立する必要がある [28]．

ルートリフレクタとは，Originator-ID と呼ばれる特殊な属性を Adj-RIB-Out に付与することでルートループを防ぎながら，IBGP ピアから受信した経路を他の IBGP に対して広告する特殊な BGP スピーカである [29]．ルートリフレクタは IBGP のコネクション数を削減するために広く利用されている．

ルートリフレクタを複数設置することで，負荷分散・冗長化構成を容易に実現することが出来る．一般的にはルートリフレクタ間はフルメッシュでの BGP コネクションを確立する設計を行うが，Gutiérrez らによればツリー型の BGP コネクション関係を一部で選択することにより，よりルートリフレクタに掛かる負荷を軽減出来ることが明らかになっている [30].

BR 数を N，サーバ数を M，ルートリフレクタの数を L とした，ルートリフレクタを利用した本提案手法での必要な BGP コネクション数 C_b は式 5.2 のように表現できる．なおルートリフレクタ間の BGP コネクションはフルメッシュを想定している．第 5.3 節で述べた基本的なネットワーク設計を行う場合と比較して，SIIT-DC ネットワークが大きくなった場合に BGP コネクションが大幅に削減できることがわかる．

$$C_b = \frac{L(2M + 2N + L - 1)}{2} \tag{5.2}$$

5.4.1 各ノードの役割と機能要件

BR 及び IPv4 提供サーバ

BR 及び IPv4 サービス提供サーバは，ルートリフレクタとのみ BGP コネクションを確立する．複数ルートリフレクタを配備する場合，それぞれとコネクションを確立することで冗長性を高めることが出来る．その他の機能は第 5.3 で述べたものと同様に配備する．

ルートリフレクタ

ルートリフレクタでは，ルートリフレクタ機能が有効となった BGP デーモンを配備する必要がある．各サーバ・BR と BGP コネクションを確立する．

5.5 各アプローチとの比較

第 4.3 節で検討した各アプローチと本提案手法を，第節で挙げた各性能要件に関して比較する．

表 5.1: 各手法の比較

手法	EAMT の一貫性	変更追従性	コネクション数	デプロイメントの容易さ
参考: オペレーターによる手動設定	無し	無し	——	——
中央管理型アプローチ	有り	(監視機構の実装依存)	$\frac{L(2M+2N+L-1)}{2}$	困難 (コントローラーの実装依存)
分散管理型アプローチ	無し	有り	$M \cdot N$	有り
提案手法 1: IBGP	有り	有り	$M \cdot N$	容易
提案手法 2: IBGP + ルートリフレクタ	有り	有り	$\frac{L(2M+2N+L-1)}{2}$	容易 (RR は容易にスケールアウト可能)

表 5.1 にそれぞれの項目における比較結果を示す．なお，コントローラーもしくはルートリフレクタの導入数を L，サーバ数を M，BR の数を N としている．

5.5.1 EAMT の一貫性

BGP はインターネットの経路広告手法として広く利用されており，多数のルータ間で一定の一貫性を保つことがプロトコルレベルで保証されている．本提案手法では BGP を利用したメッセージングにより，SIIT-DC ネットワークにおけるダイナミック EAMT を実現しているため，BGP と同水準の一貫性の担保が可能である．

5.5.2 変更追従性

本提案手法では各 IPv4 サービス提供サーバが自身の EAM を BGP の経路情報として広告するため，サーバが広告するモデルを採用する分散管理型アプローチと同様に，経路広告の有無によりサーバのネットワーク健全性を保証する事ができる．

一方で 4.3.1 項で述べたように，中央管理型アプローチにおいても変更追従性を実現可能であるが，コントローラの監視機構の実装に依存するほか，何らかの手段によってマスターテーブルに対して変更を適用する手段を考慮する必要がある．

5.5.3 コネクション数

各サーバが直接 BR に対してコネクションを確立する分散管理型アプローチ及び提案手法１では，サーバ・BR の数が大きくなった場合に，EAMT の維持・管理に必要なコネクション数が増加する．

一方でルートリフレクタを採用した提案手法２と中央管理型アプローチでは，サーバ・BR の数が増加した場合にも，上記２手法と比較して少ないコネクションで EAMT を維持・管理することが可能である．

5.5.4 デプロイメントの容易さ

本提案手法では動的経路制御プロトコルとして一般的な BGP をメッセージングに利用しているため，OSS を含む多種多様な実装をそのまま利用することが可能であり，他の手法と比較して実装・デプロイメントが容易である．

また 7.5 節で述べたように，ルートリフレクタ間のトポロジや管理方法を工夫することで，更なるコネクション数とルートリフレクタの負荷軽減を実現する余地がある．

一方で，中央管理型アプローチでは，特別な監視・管理機構を持ったコントローラを実装する必要があり，他のデプロイメントの障壁が高くなる．

第6章　プロトコル設計と実装

本章では，第 5 章で述べた提案システムのメッセージ設計と実装について述べる．

6.1　BGP UPDATE メッセージの設計

本提案手法ではサーバ・BR・ルートリフレクタ間のメッセージングに BGP を利用する．本節では BGP UPDATE メッセージの設計を行う．

6.1.1　要件

Dynamic EAMT を実現するにあたって，EAM として広告すべきに必要な属性は 1) IPv4 サービスアドレス，2)IPv6 サービスアドレス, 3) 変換プレフィックスの 3 種が想定される．表 6.1.1 に各属性の情報を列記する．

表 6.1: EAM に必要な情報

属性名	型	備考	例
IPv4 サービスアドレス	IPv4 ネットワークアドレス	IPv6 サービスアドレスとホストアドレス長が一致	192.0.2.1/32
IPv6 サービスアドレス	IPv6 ネットワークアドレス	IPv4 サービスアドレスとホストアドレス長が一致	2001:db8:200::1/128
変換プレフィックス	IPv6 ネットワークアドレス (/96)		64:ff9b::/96

6.1.2　実装

本提案手法では，IPv6 ユニキャスト経路[1]として，BGP を利用して EAM を広告・交換する．UPDATE メッセージ以外の扱いは標準的な BGP メッセージに準ずる．

BGP UPDATE メッセージ

本提案手法における BGP UPDATE メッセージに含有するパス属性[2]を図 6.1.2 のように定義した．

[1]アドレスファミリー番号 2, サブアドレスファミリー番号 1[24, 25]
[2]Path Attributes

表 6.2: BGP UPDATE メッセージにおける各パス属性

タイプコード値	パス属性	必須	値	備考	例
1	ORIGIN	必須	2(IMCOMPLETE)	本実装においては利用しない.	2
2	AS_PATH	必須	AS 番号	iBGP のみで広告するため，自身の AS 番号を記載する	65001
5	LOCAL_PREF	任意	1 ～65535	EAM の優先度	200
8	COMMUNITY	任意	[0～65535]:[0～65535]	BGP コミュニティ名	2500:200
9	ORIGINATOR_ID	必須	BGP Identifier	自身のルータ ID	192.0.2.1
10	CLUSTER_LIST	任意	クラスタ ID	ルートリフレクタを利用する場合，要指定 同じ EAMT を共有する範囲を指定する	192.0.2.1
14	MP_REACH_NLRI ->NLRI	必須	IPv6 アドレス+プレフィックス長	変換プレフィックス + IPv4 サービスアドレス/128	64:ff9b::192.0.2.1/128
14	MP_REACH_NLRI ->NEXT_HOP	必須	IPv6 アドレス	変換プレフィックス + IPv4 サービスアドレス/128	2001:db8:200::1
15	MP_UNREACH_NLRI	必須	MP_REACH_NLRI と同様		

6.1.3 実装時に留意すべき事項

BGP を利用した Dynamic EAMT において留意すべき事項を述べる.

ルーティングテーブルの隔離

通常の IGP・EGP 経路とは用途が異なるため，何らかの仮想化技術を利用してそれらと EAMT を BGP スピーカーが区別する必要がある．具体的には VRF(Virtual routing and forwarding) などのルーティングテーブル仮想化技術の利用が想定される.

ホストルートでの利用に限定

本提案手法では MP_REACH_NLRI 及び MP_REACH_NLRI のアドレスファミリーとして IPv6 ユニキャスト経路を利用している．そのため実装上の問題から，IPv6 サービスアドレス及び IPv4 サービスアドレスがそれぞれ 1 アドレスの場合のみをサポートしている.

6.2 PoC の実装

第 7 章で行う概念検証実験にもちいる PoC(Proof of Concept)[3]について，各ノードで必要なコンポーネントとその役割及び具体的な実装について記述する.

6.2.1 各コンポーネントの実装

第 5.3.1 項で述べたコンポーネント群は以下の様にそれぞれ実装した．BR における BR に必要なコンポーネント群の関係図を図 6.2 に示す．表 6.2.1 にコンポーネント群の情報の概要を示す.

[3]概念検証実装

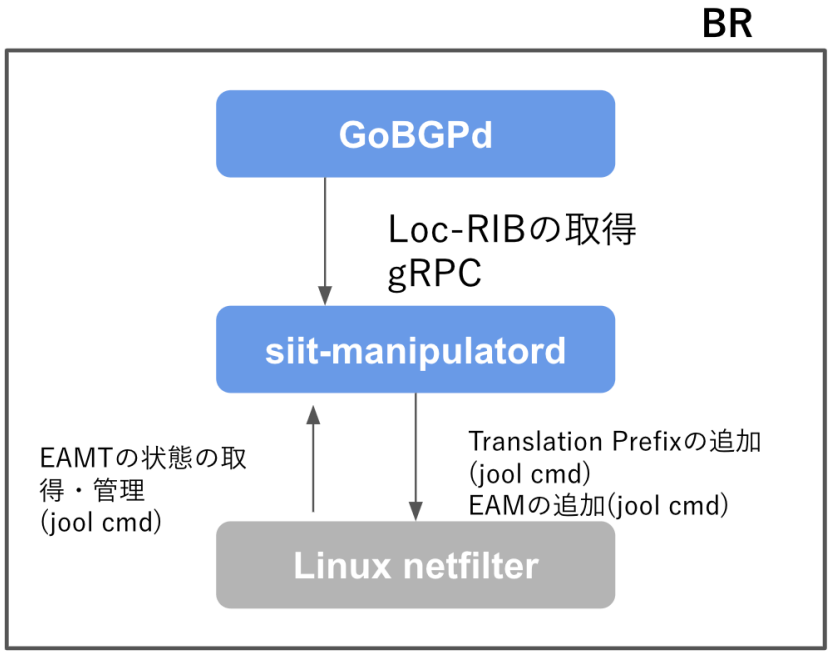

図 6.1: BR に必要なコンポーネント群の関係図

表 6.3: PoC の実装に利用したソフトウェア群

ソフトウェア名	種別	バージョン	概要
GoBGP	BGP デーモン	2.1.1	Go 言語によって実装された OSS の BGP デーモン
Jool	SIIT 機構	4.0.6.2	SIIT や Statefull NAT64 を Linux 上で動作させるための OSS.
siit-manipulatord	EAMT 制御機構	—	SIIT 機構を BGP に伴って制御するための自作アプリケーション
Python	実行環境	3.7.3	siit-manipulatod に利用する.
PyYAML	ライブラリ	5.2	siit-manipulatod に利用．YAML 記法で書かれた設定ファイルの読み込みを行う.
logzero	ライブラリ	1.5.0	siit-manipulatod に利用．ログファイルの書き出しに利用.

BGP デーモン

BGP デーモンには，OSS の BGP デーモンである GoBGP[4]を利用する．GoBGP では gRPC[5]を用いた操作機構が実装されており，同期・非同期を問わず他のアプリケーションとの連携が容易に行える．RouteReflector 機構もサポートされているため，本 PoC では全てのノードの BGP デーモンとしてこれを利用する．

SIIT

SIIT には Jool[6]の SIIT モードを利用する．Jool は LinuxOS で利用できる NAT64/SIIT 環境で，Lunux Netfilter によって実装されており，汎用的に様々なプラットフォームでの

[4]GoBGP `https://osrg.github.io/gobgp/`
[5]gRPC Remote Procedure Calls. `https://www.grpc.io/`
[6]Jool `https://jool.mx/en/index.html`

利用が可能である [31, 32]．EAMT の変更には専用の CLI コマンドを利用する．

EAMT 制御機構

BGP 上で受信した Loc-RIB を EAMT に反映するために，EAMT 制御機構”siit-manipulatord” を実装する．gRPC によって GoBGP の Loc-RIB の変化を監視し，Jool の CLI コマンドを利用して Linux NetFilter に変更を反映するほか，Translation Prefix の定義など SIIT に必要な情報を管理する．

6.2.2 メッセージングと状態遷移

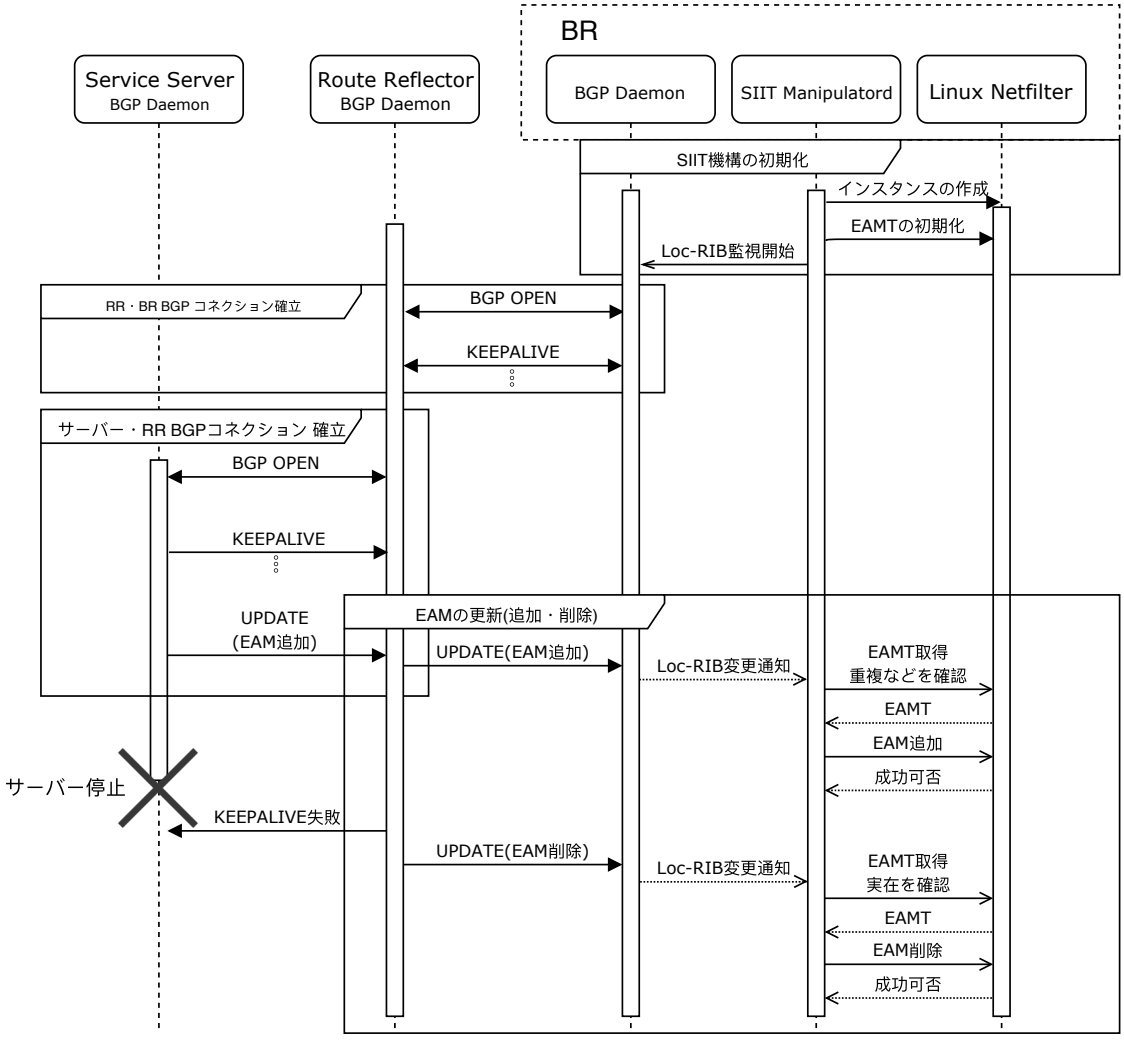

図 6.2: 本 PoC における各ホスト・コンポーネントの相互作用と状態遷移

図 6.2 に IPv4 サービス提供サーバ・ルートリフレクタ・BR の相互作用と状態遷移の概要を示す.

以下に各状態にあるホスト・コンポーネント間の相互作用に関して記述する.

6.2.3 SIIT 機構の初期化

BR は起動後ネットワーク環境の準備が出来次第，BGP デーモンと EAMT 制御機構のサービスを開始する．EAMT 制御機構は初めに Linux Netfilter を操作し，SIIT によるプロトコル変換を行う変換インスタンス[7]を作成する．このインスタンスは BR が利用する変換プレフィックスを保持する．その後 EAMT を初期化し，gRPC を利用して BGP デーモンから Loc-RIB の監視を開始する．以後 Loc-RIB に変更があるまで待機する．Loc-RIB の変更はベストパスの変更に伴っても発生するため，BGP NOTIFICATION メッセージなどにより，ベストパスとなる経路情報を発信していた BGP ピアとのコネクションが切断されたような場合にも，EAMT 制御機構に変更通知が送信される.

6.2.4 ルートリフレクタ・BR 間の BGP コネクションの確立と維持

BR の BGP デーモンは起動次第，事前に登録されたルートリフレクタに対する BGP OPEN メッセージの送信を開始し，BGP コネクションの確立を試みる．RR からの BGP OPEN メッセージの回答を受けて BGP コネクションを確立し，以後 BGP KEEPALIVE メッセージによって接続性の死活監視を行う.

6.2.5 IPv4 サービス提供サーバ・ルートリフレクタ間の BGP コネクションの確立と維持

IPv4 サービス提供サーバの BGP デーモンは起動に伴って，自身の Loc-RIB に EAM(変換プレフィックスと IPv4 サービスアドレスによって構成された NLRI を埋め込んだ経路情報) を登録する．BGP コネクションが確立次第，IPv4 サービス提供サーバは UPDATE メッセージにより EAM の追加をルートリフレクタに通知する．このコネクションでも同様に，以後 BGP KEEPALIVE メッセージによって接続性の死活監視を行う.

6.2.6 EAM の追加

IPv4 サービス提供サーバから UPDATE メッセージはルートリフレクタにより各 BR に伝達され，UPDATE メッセージを受信した BR の BGP デーモンは自身の Loc-RIB を更新する.

[7] Jool Instance. ネットワーク名前空間ごとに一つ存在可能である.

Loc-RIB の更新に伴い Loc-RIB の監視を行っていた EAMT 制御機構に gRPC を介し更新通知が送信される．以後の EAMT 制御機構の処理はコルーチンを利用した非同期処理によって行われるため，EAMT 走査にまつわる処理を行っている最中であっても Loc-RIB の監視はブロッキングされない．

EAMT 制御機構は EAMT の更新を滞りなく行うため，LinuxNetfilter に登録された現在の EAMT の状態を取得し，競合する EAM が登録されていないことを確認する．IPv4 サービスアドレスと IPv6 サービスアドレスが一致する EAM が存在していた場合や事前に登録された変換プレフィックスに一致しない場合，以後の処理をスキップする．

新しく受信した EAM に問題がない場合，EAMT 制御機構は EAM の追加を試みる．Linux Nefilter は新しい EAM が追加された EAMT を参照し，直ちにパケットのフォワーディングを開始する．

6.2.7 EAM の削除

ルートリフレクタは，何らかの理由で IPv4 サービス提供サーバとの BGP KEEPALIVE メッセージが失敗した場合や BGP NOTIFICATION メッセージを受信した場合，自身の Loc-RIB から該当の EAM を削除し，Adj-RIB-OUT の情報を更新する．

ルートリフレクタから EAM 削除を広告する BGP UPDATE メッセージを受け取った BR の BGP デーモンは Loc-RIB を更新し，EAMT 制御機構に対して Loc-RIB 変更通知を送信する．EAM の追加時と同様に，EAMT を取得して該当する EAM の実在を確認後，EAMT 制御機構は EAM 削除を試みる．

6.2.8 EAM の更新

BGP 経路情報の属性値の変更などに伴って BR の Loc-RIB に登録されるベストパスが変更になる場合がある．

BR の BGP デーモンは EAM 追加・削除時同様に，EAMT 制御機構に対して Loc-RIB 変更通知を送信する．この変更通知に伴って EAMT 上の古い EAM は削除され，新しい EAM に更新される．

第7章　評価

本章では，第 5 章及び第 6 章で設計・実装に関して述べた本提案手法に関して，第 3.2 節で指摘した SIIT-DC の課題に対して有効性があることを評価する．

7.1　評価要件

SIIT-DC におけるダイナミック EAMT 機構では，第 5.5 節で述べた事項の充足が求められる．それらを定量的に評価するための指標・尺度について記述する．

7.1.1　BR 間の EAMT の一貫性

SIIT-DC ネットワークにおける各 BR は一貫性のある EAMT を保持する必要がある．本評価実験では後に述べるテストケースにおいて，以下の 2 点を検証することで各 BR の EAMT が一貫性を有すると定義する．

- **EAM 数の収束**
  EAM の更新が行われる事象が発生した場合に，一定の収束時間が経過した後に各 BR の EAMT のレコード数が変化することなく，一意に収束すること．
- **EAMT のレコードの内容**
  EAM 数が収束した際に各 BR の有する EAMT の内容に差異が無いこと．

7.1.2　変更追従性

前に述べたように，従来の SIIT-DC の構成では，IPv4 サービスが追加・削除・変更された場合や BR の追加配備を行った場合に，各 BR の EAMT の内容を適宜変更する必要があった．

本研究では EAMT 変更が必要な事象が発生した場合に，運用者が特別なオペレーションを行うことなくサービス提供を想定通りに開始・継続・中断される状況を，EAMT が変更に追従している状態であると定義する．

本評価実験では下記の様な EAMT 変更が必要な事象のうち，IPv4 サービの追加・削除について検証する．

- **IPv4 サービスの追加・削除**
  ネットワーク内の IPv6 サービスにおいて，IPv4 サービスアドレスを付与することで IPv4/IPv6 の両プロトコルによるサービス提供を開始する場合．もしくは，そのサーバが行っている IPv4 サービス提供を中断する場合．

- **IPv4 サービス提供サーバの変更**
  SIIT-DC において提供中の IPv4 サービスに関して，他の IPv6 アドレスを持つサーバによるサービス提供を改めて開始する場合．

- **BR の追加・撤去**
  対外接続点に新しい BR を追加もしくは稼働中の BR を撤去しながら IPv4 サービスを継続して行う場合．

7.1.3 スケーラビリティ

ダイナミック EAMT を実現する機構は対外接続点や IPv4 提供サービスの増加に柔軟に対応可能であることが望ましい．

本評価実験では各ノードのスケールにおいては下記を想定した評価試験用ネットワークを実装する．IPv4 提供サービスの総数が増加した場合においても，本提案手法が動作が可能であることを評価する．

- BR: 30 ホスト
- ルートリフレクタ: 2 ホスト
- IPv4 提供サーバ: 10 〜 1000 ホスト

7.2 実験環境

本評価実験で利用した実験環境について述べる．

7.2.1 ネットワークトポロジ

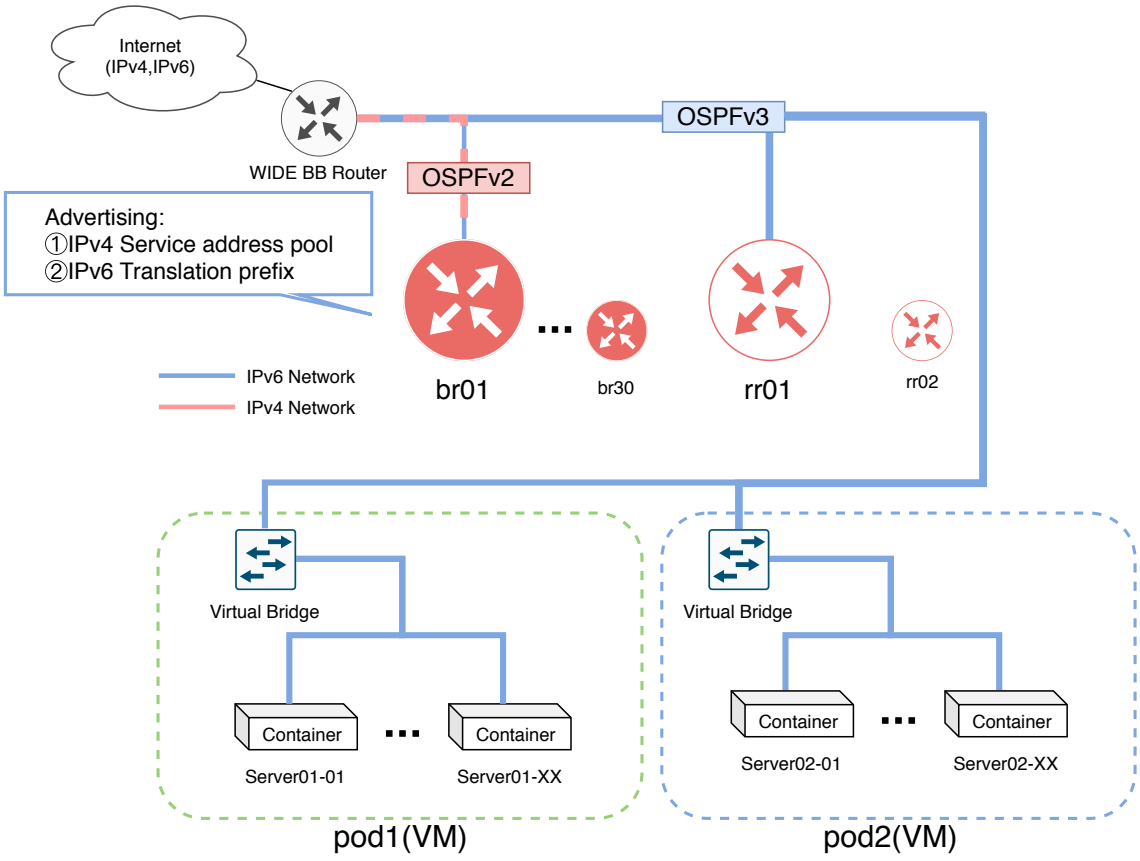

図 7.1: 評価実験におけるネットワークと各ホスト

本評価実験を行うために，WIDE プロジェクト[1]藤沢 NOC 内に実験用ネットワークを構築した．図 7.1 に本評価実験用ネットワークの概略を示す．本ネットワークでは第 3.1.4 項で述べた ECMP エニーキャストによる BR の水平スケールを可能にするネットワークトポロジを採用している．

本ネットワークでは，BR 群 (br01 ··· br30) のみ，WIDE ネットワークを介して IPv4 インターネットに接続している．WIDE プロジェクト内のルータに対して，IPv4 サービス提供サーバ群 (Server01-01 ··· Server02-60) が利用する IPv4 サービスアドレス群を，OSPFv2[33] を利用した ECMP エニーキャストにて広告する．これにより，インターネット上の IPv4 クライアントからの IPv4 パケットはいずれかの BR を経由し，ダイナミック EAMT を参照した SIIT によるネットワークプロトコル変換が各 BR にて行われたのち，IPv6 パケットとして各 IPv4 サービス提供サーバに転送される．

各ホストは共通の IPv6 ネットワーク上に所属し，WIDE ネットワークを介してインターネット疎通性を持つ．この IPv6 ネットワークでは，各 BR から IPv4 サービス提供サーバ

[1] https://www.wide.ad.jp/

表 7.1: BGP コネクションで利用したパラメータ

要素	値	説明
Keep Alive Interval	3秒 (標準:60秒)	KEEPALIVE メッセージの送信間隔
Hold Time	9秒 (標準：180秒)	この値以上 KEEPALIVE の到達を確認出来なかった場合，当該ピアから受信した経路を破棄する．
ConnectTimer	5秒 (標準：120秒)	BGP コネクションの再試行を行う間隔.

向けに，OSPFv3[34] を利用した ECMP エニーキャストによる変換プレフィックス向けの経路広告が行われている．IPv4 サービス提供サーバからの変換プレフィックス宛のパケットは，いずれかの BR を経由して SIIT によるネットワークプロトコル変換がなされたあと，インターネット上の IPv4 提供クライアントに向けて返送される．

各ホスト間のダイナミック EAMT に用いる BGP メッセージングは，各ホストが共通して属する IPv6 ネットワークを介して行われる．BR 群及び IPv4 サービス提供サーバ群はルートリフレクタ群 (rr01 ⋯ rr02) に対して，IPv6 による IBGP コネクションを確立する．本ネットワークではルートリフレクタ間の BGP メッセージングは行わない一層のルートリフレクタ構成を採用する．各 IPv4 サービス提供サーバからの経路情報は，ネットワーク内で一意のクラスタ ID を付与した上で BR 群に広告される．

なお，本実験環境では BGP コネクションを維持するための各種パラメータを表 7.2.1 の様に指定する．

各ホストは各 BR は NTP(Network Time Protocol)[35] を利用した時刻同期を行う．同一ネットワーク上の NTP サーバを参照元とし，十分な精度で同期を行っていると仮定する．

7.2.2 実装

本項では各ホストの実行環境について記述する．各ホストが利用している，BGP デーモン・SIIT 機構・EAMT 制御機構の各コンポーネントは，第 6.2.1 項で述べた PoC と同一の物を利用している．

BR

本評価ネットワークにおける BR 群は，同一物理サーバ上で仮想マシンとして稼働する．表 7.2.2 に物理サーバと仮想マシンの実行環境を詳細に示す．

また，各 BR 上で現在保有している EAMT 数・BGP 経路数を計測するための監視プロセスを 0.2 秒毎に実行し，各種評価要件の検証に利用する．

ルートリフレクタ

本評価ネットワークにおけるルートリフレクタ群は，同一物理サーバ上仮想マシンとして稼働する．表 7.2.2 に物理サーバと仮想マシンの実行環境を詳細に示す．

本ルートリフレクタでは BGP デーモンにて，ルートリフレクタ機能に加えてダイナミックネイバー機能を有効化する．通常 BGP コネクションを確立するためには，BGP ピア間で互いのピアリングアドレスを明示的に設定する必要があるが，本機能を利用すること

表 7.2: 評価実験用 BR 群の実行環境

計算環境			
機種	CPU	RAM	NIC
Intel S2600WTT	2CPU: Intel Xeon E5-2699 v3 2.30GHz	64GB	Intel 82599EB 10-Gigabit SFI/SFP+
ハイパーバイザ環境			
OS	バージョン	vCPU	備考
VMware ESXi	6.7.0	72	build: 5160138
仮想マシン			
OS	カーネル	リソース	利用コンポーネント
Ubuntu 18.04.3 LTS	4.15.0-72-generic	vCPU: 2 RAM:2GB	BGP デーモン，EAMT 制御機構，SIIT 機構

表 7.3: 評価実験用 RR 群の実行環境

計算環境			
機種	CPU	RAM	NIC
Dell PowerEdge R410	2CPU: Intel Xeon E5530 @ 2.40GHz	16GB	HP NC522SFP Dual Port 10GbE Server Adapter
ハイパーバイザ環境			
OS	バージョン	vCPU	備考
VMware ESXi	6.5.0	16	build: 15256549.
仮想マシン			
OS	カーネル	リソース	利用コンポーネント
Ubuntu 18.04.3 LTS	4.15.0-72-generic	vCPU: 4 RAM:4GB	BGP デーモン

で動的に多数の BGP スピーカとのピアリングが可能になる [36]．同様の機構は他の BGP 実装やネットワーク機器にも搭載されている [37].

IPv4 サービス提供サーバ

本実験環境において，IPv4 サービス提供サーバ群はコンテナ型仮想化 [38] を利用し，二つのホスト仮想マシン上で実行される．図 7.2.2 に物理サーバとホスト仮想マシン及びコンテナの実行環境を詳細に示す.

表 7.4: 評価実験用 IPv4 サービス提供サーバ群の実行環境

計算環境			
機種	CPU	RAM	NIC
Dell PowerEdge R410	2CPU: Intel Xeon E5530 @ 2.40GHz	48GB	HP NC522SFP Dual Port 10GbE Server Adapter
Dell PowerEdge R410	2CPU: Intel Xeon E5530 @ 2.40GHz	48GB	HP NC522SFP Dual Port 10GbE Server Adapter
ハイパーバイザ環境			
OS	バージョン	vCPU	備考
VMware ESXi	6.5.0	16	build: 15256549． Pod1,Pod2 で共通
ホスト仮想マシン			
OS	カーネル	リソース	備考
Ubuntu 18.04.3 LTS	4.15.0-72-generic	vCPU: 4 RAM:4GB	Pod1, Pod2 で共通
コンテナ			
OS	コンテナエンジン	コンテナイメージ	利用コンポーネント
Alpine Linux 3.11	Dockerversion 19.03.5	自作	GoBGP デーモン

7.3 評価実験 1: EAMT の収束・一貫性の検証

本節では 7.1 項で述べた三つの要件のうち,「BR 間の EAMT の一貫性」及び「スケーラビリティ」の検証を目的とした評価実験を行い，その詳細と結果について述べる．

7.3.1 シナリオ

基本的な SIIT-DC ネットワークの構築を想定し，IPv4 提供サービスサーバ群を起動した後に，BR 間の EAMT の一貫性が正しく維持されるかを検証する．

手順

以下のような手順を 1 単位として，IPv4 サービス提供サーバ数 x につき 1 回試行する．

1. **ルートリフレクタの起動**
   ルートリフレクタ群を起動し，BGP デーモンの立ち上げを確認する．
2. **BR の起動**
   BR 群を起動し，BGP デーモン・EAMT 制御機構の立ち上げを確認する．
3. **IPv4 サービス提供サーバの起動**
   IPv4 サービス提供サーバを x ホスト起動し，それぞれのサーバにおいて BGP デーモンの立ち上げ及び EAM 情報の Loc-RIB への反映を確認する．この時利用する IPv4 サービスアドレスは事前に作成した正答となる EAMT を参照する．実験環境の都合上，サーバの起動は同期的に行う．
4. **計測**
   全てのサーバが起動の起動を確認し，サーバの起動から 30 秒後の各 BR の EAMT の状態を収集する．EAMT の内容を，事前に作成した正答となる EAMT と比較する．

7.3.2 説明変数・目的変数

本実験環境における BR の数を B，事前に作成した EAMT に正答した BR の数 B_s とし，正答率 R を目的変数とする．目的変数 R は以下の様な式 7.1 で表す事ができる．また IPv4 サービス提供サーバの数を x とし，本評価実験の説明変数とする．

$$R = \frac{B_s}{B} \tag{7.1}$$

7.3.3 条件

各ホストの情報を表 7.3.3 に示す．ここに記載の無いはその他の条件は 7.2 項に準ずる．

表 7.5: 評価実験 1 での各ホストの情報

要素	ホスト数	ホスト名
BR	30	br01〜br30
ルートリフレクタ	2	rr01〜rr02
IPv4 サービス提供サーバ	10,100,200,400,500,600,700,800,900,1000	server01-01 〜 server01-500, server02-01 〜 server02-500

表 7.6: 評価実験 1 での各ホストの情報

x: IPv4 サービス提供サーバの数	R: 正答に EAMT が一致した BR 数の割合	EAMT の収束可否
10	100 %	収束
100	100 %	収束
200	100 %	収束
300	100 %	収束
400	100 %	収束
500	100 %	収束
600	100 %	収束
700	0 %	収束せず
800	0 %	収束せず
900	0 %	収束せず
1000	0 %	収束せず

7.3.4 結果と考察

本評価実験の結果を表 7.3.4 に示す.

IPv4 提供サービスサーバ数 x が 600 までの場合に，全ての BR 間で保有する EAMT が収束し，一貫性を維持していることが読み取れる．一方で $x >= 700$ の場合には全ての BR の EAMT が正答に一致しておらず，且つ収束しないことがわかった.

プログラム 7.1: EAMT が収束しない時点のルートリフレクタのログの抜粋

```
1        Jan 06 13:11:56 rr01 gobgpd[18970]: {"Code":4,"Data":null,"Key
             ":"2001:200:0:8831:1:0:6440:243","Subcode":0,"Topic":"Peer
             ","level":"warning","msg":"received notification","time
             ":"2020-01-06T13:11:56+09:00"}
2        Jan 06 13:11:56 rr01 gobgpd[18970]: {"Key
             ":"2001:200:0:8831:1:0:6440:243","Reason":"notification-
             received code 4(hold timer expired) subcode 0(undefined)","
             State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","
             msg":"Peer Down","time
3        Jan 06 13:12:00 rr01 gobgpd[18970]: {"Key
             ":"2001:200:0:8831:1:0:6440:1d","State":"
             BGP_FSM_ESTABLISHED","Topic":"Peer","level":"warning","msg
             ":"Closed an accepted connection","time":"2020-01-06T13
             :12:00+09:00"}
```

当該期間中のルートリフレクタ (rr01) のログメッセージの抜粋をコード 7.1 に示す．BGP ピア (IPv4 サービス提供サーバ) より，ルートリフレクタから送信される BGP KEEPALIVE

メッセージに問題がある旨を表す BGP NOTIFICATION メッセージを受信していることが読み取れる．$x = 700$ の実験期間中，常にこのようなログメッセージが高頻度で観測されていた．本問題はルートリフレクタが一定以上の BGP ピアを収容する場合，正常に KEEPALIVE メッセージを送信出来ず，全体の BGP コネクションの維持に問題が生じていることに起因することがわかる．

本評価実験を通して，本提案手法が EAMT の一貫性の維持に効果的に作用することがわかった．しかしながら本評価環境においては，ルートリフレクタが収容できる BGP コネクションに限りがあり，ルートリフレクタが保有する BGP コネクションの数が一定の値を超えた場合に，全ての BR の EAMT が維持できなくなる課題が明らかになった．

7.4 評価実験 2: 構成変更追従性の検証

本節では 7.1 項で述べた三つの要件のうち,「変更追従性」「スケーラビリティ」の検証を目的とした評価実験を行い，その詳細と結果について記述する．

7.4.1 シナリオ

IDC ネットワークにおいて IPv4 サービス構成が変更になったことを想定し，変更が発生した時点から各 BR の EAMT が一貫性を維持するまでの経過時間を計測する．

構成変更の定義

本評価実験では以下の二つの事例を「IPv4 サービス提供サーバの構成変更」と定義する．

1. **IPv4 サービス提供サーバの追加**
   IPv4 サービス提供サーバ 1 台を新規に起動し，BGP デーモンの立ち上げ及び当該サーバが IPv4 サービスを行うために必要な EAM の情報を，当該サーバの Loc-RIB に反映する．

2. **IPv4 サービス提供サーバの削除**
   任意の IPv4 サービス提供サーバ宛のパケットを，ホスト仮想マシン上でフィルターし，当該サーバがルートリフレクタと疎通不能な状態を再現する．

手順

以下のような手順を 1 単位として，初期状態の IPv4 サービス提供サーバ数 x につき各 30 回試行する．

1. **初期状態の準備**
   7.3 節で述べた手順を完了した IPv4 サービス提供サーバ x ホストが，各 BR を介した IPv4 サービスが提供可能になった状態を初期状態とする．

2. **IPv4 サービス提供サーバの構成変更**
   7.4.1 項で述べた二つの事例のうち，一つを実行する．

3. **計測**
   IPv4 サービス提供サーバの構成変更から，各 BR の EAMT が収束[2]するまでに経過した時間を計測する．

7.4.2 説明変数・目的変数

第 6.2.2 項で述べた本提案手法のメッセージングのシーケンスから，IPv4 サービス提供サーバの追加・削除に関わるものを抜粋する．

IPv4 サービス提供サーバの追加後のプロセス

BR の EAMT が IPv4 サービス提供サーバの追加に追従するまでに必要な各コンポーネントの動作は，以下の 5 つの段階に分類することが出来る．

1. IPv4 サービス提供サーバが自身の Loc-RIB に EAM を反映
2. IPv4 サービス提供サーバとルートリフレクタ間の BGP コネクションの確立
3. IPv4 サービス提供から BGP UPDATE メッセージがルートリフレクタに伝達
4. ルートリフレクタが各 BR 及びに UPDATE メッセージを送信
5. 各 BR が自身の Loc-RIB を変更し EAMT を更新

目的変数 T_a は以下の様な式 7.2 で表す事ができる．本実験環境における BR の数を B，IPv4 サービス提供サーバ 1 台がコネクション確立に掛かる時間を t_c，IPv4 サービス提供サーバ 1 台が BGP UPDATE メッセージをルートリフレクタに伝達するまでに掛かる時間を t_u，ルートリフレクタが 1 台の BR に UPDATE メッセージを送信するのに掛かる時間を t_b，1 台の BR が Loc-RIB を変更し EAMT を更新するまでに掛かる時間を t_e，IPv4 サービス提供サーバの数を x とし，x を本評価実験の説明変数とする．なお，各 BR は同一の時間 t_e で並行して EAMT を更新するものとする．

$$T_a = t_c + t_u + B \cdot x \cdot t_b + t_e \tag{7.2}$$

[2] ここでは 7.1.1 項で述べた EAMT が一貫性を維持した状態とする．

IPv4 サービス提供サーバの削除後のプロセス

また，IPv4 サービス提供サーバの削除を行う場合，IPv4 サービス提供サーバの削除から EAMT の収束までの各動作は以下のように分類される．

1. ルートリフレクタが BGP KEEPALIVE メッセージの不到達を検知
2. ルートリフレクタが各 BR 及びに UPDATE メッセージを送信
3. 各 BR が自身の Loc-RIB を変更し EAMT を更新

目的変数 T_b は以下の様な式 7.3 で表す事ができる．なお，各 BR は同一の時間 t_e で並行して EAMT を更新するものとする．なお，ルートリフレクタが BGP KEEPALIVE メッセージの不到達を検知を検知するまでの時間を t_d，本実験環境における BR の数を B，IPv4 サービス提供サーバの数を x とし，x を本評価実験の説明変数とする．

$$T_b = t_d + B \cdot (x-1) \cdot t_b + t_e \tag{7.3}$$

7.4.3 条件

各ホストの情報を表 7.3.3 に示す．記載の無いその他の条件は 7.2 項に準ずる．初期状態の IPv4 サービス提供サーバの数 x に関しては，7.3.4 項で得られた結果から妥当な数量を決定した．

表 7.7: 評価実験 2 での各ホストの情報

要素	ホスト数	ホスト名
BR	30	br01〜br30
ルートリフレクタ	2	rr01〜rr02
初期状態の IPv4 サービス提供サーバ	50,100,150,200,250,350,400,450,500,550,600	server01-01 〜 server01-300, server02-01 〜 server02-300
追加する IPv4 サービス提供サーバ	1	extra01-01

7.4.4 結果と考察

本評価実験の結果のうち,「IPv4 サービス提供サーバの追加」を図 7.2 に,「IPv4 サービス提供サーバの削除」を図 7.2 に示す．これらの図では，各説明変数 x での試行の分布と，それぞれの平均の推移を表している．

はじめに本評価実験の概況について考察する．いずれのケース及び説明変数 x に於いても，各 BR の EAMT が平均して概ね 10 秒程度で収束していることがわかる．これらの結果から，本実験環境における IPv4 サービス提供サーバ数 x の範囲内では，サーバ数の多寡に関わらず，変更追従機構が効果的に作用していることが評価できる．

次に，各ケースにおける経過時間 (T_a,T_b) の推移について述べる．IPv4 サービス提供サーバの削除を行ったケースでは，初期状態の IPv4 サービス提供サーバ数 x に関わらず

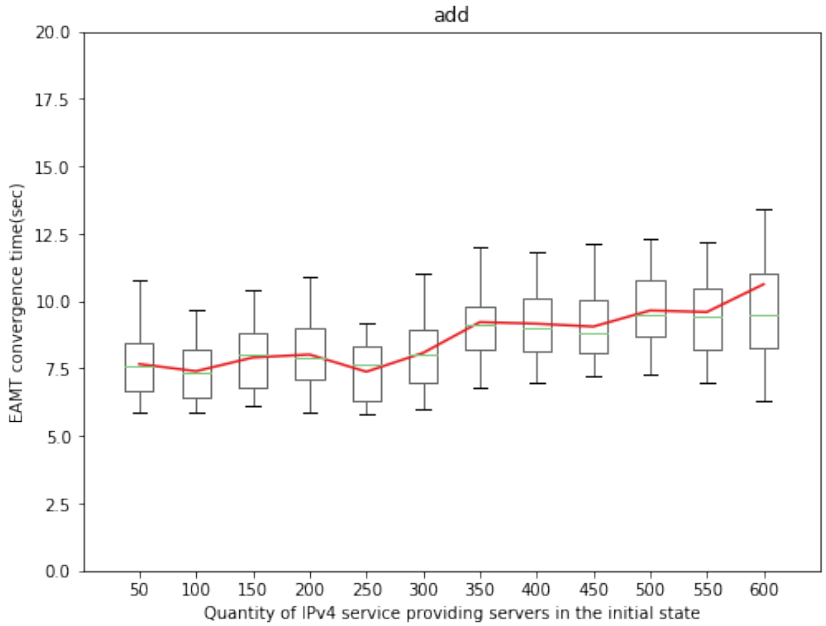

図 7.2: 評価実験 2:IPv4 サービス提供サーバの追加を行った際の EAMT 収束時間の比較

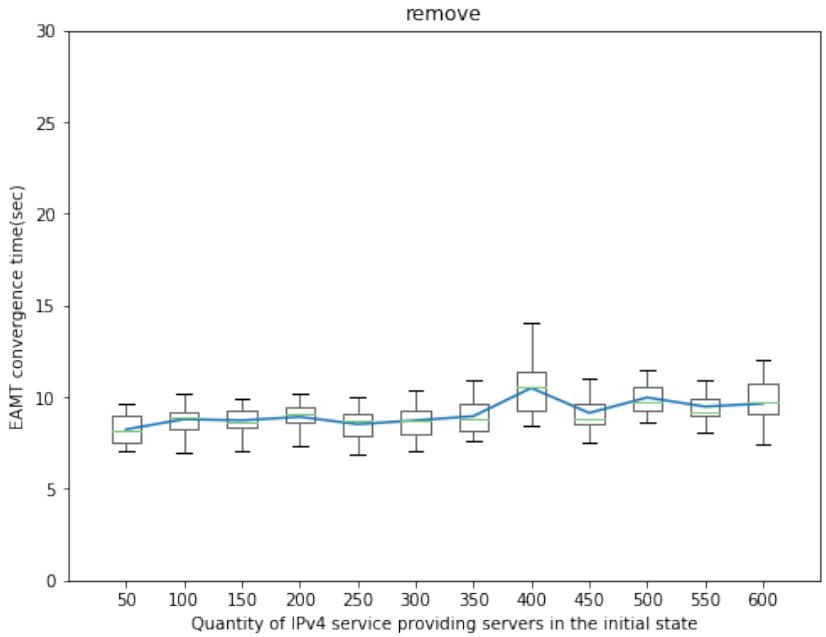

図 7.3: 評価実験 2:IPv4 サービス提供サーバの削除を行った際の EAMT 収束時間の比較

EAMT の収束に掛かる時間 T_b はほぼ横ばいに推移している．一方で，サーバの追加を行ったケースでは，x が大きくなるに従って，EAMT の収束に掛かる時間 T_a がなだらかに増加傾向にあることが読み取れる．

この事象は，ルートリフレクタが保持するコネクション数が影響しているとが考えられる．IPv4 サービス提供サーバ 1 台がコネクション確立に掛かる時間 t_c 及び，IPv4 サービス提供サーバ 1 台が BGP UPDATE メッセージをルートリフレクタに伝達するまでに掛かる時間 t_u が，ルートリフレクタが保有している BGP コネクションに連動して，増加していると思われる．

本評価実験を通して，本提案手法の変更追従性をより機敏に機能させるためには，ルートリフレクタのコネクション数を極力抑えた設計が必要になることが明らかになった．

7.5 本章のまとめ

本評価実験 1 の結果 (7.3.4) を通して本提案手法が各 BR の一貫性の維持に有効であることを，本評価実験 2 の結果 (7.4.4) を通して本提案手法が変更追従性を十分に有していることをそれぞれ証明することが出来た．

しかしながら IPv4 サービス提供サーバ数 x が一定の値より大きい場合において，ルートリフレクタが多くの BGP コネクションを保持出来ないことが明らかになっている．本問題に関しては以下のような取り組みが有効であると考えられる．

- BGP パラメータの変更
  本評価実験では，BGP KEEPALIVE メッセージージに利用する各種パラメータに関して，7.2.1 で指定したような値を採用している．KeepAlive 送信間隔をより大きくすることで，より多くのコネクションを収容可能になることが想定される．

- ルートリフレクタトポロジの見直しによる負荷軽減
  本実験環境ではルートリフレクタ 2 ホスト (rr01,rr02) と全てのサーバ間で BGP コネクションを確立する一層のルートリフレクタトポロジを採用しているが，第項で述べたような，1 台あたりのルートリフレクタの負荷を軽減できるトポロジを取り入れることで，より多くの IPv4 サービス提供サーバを本提案手法で扱うことが出来ると考えられる．

両取り組みのうち,「ルートリフレクタトポロジの見直しによる負荷軽減」に関して，第 8.2 節にて解決策となる BGP コネクショントポロジのデザインについて述べる．

第8章　結論

本章では，本研究の総括と今後の課題を示す.

8.1　本研究のまとめ

本研究では，IPv6 シングルスタックネットワークにおける IPv4 サービス提供手法として，”SIIT-DC”と呼ばれるネットワークデザインに着目した．SIIT-DC では，BR と呼ばれるプロトコル変換機構を有したルータが，アドレス変換テーブル (EAMT) を参照して IPv4/IPv6 トランスレーションを行い，IPv6 サーバで IPv4 サービスの提供を可能にする．SIIT-DC のアーキテクチャにおいて，複数の BR を運用する環境での EAMT の一貫性の確保が難しい点や，IPv4 でサービス提供を行うサーバの構成変更が行われた際に個別の運用が必要になる点を課題として示した．また，SIIT-DC のこれらの問題を解決するためには，BR が保有する EAMT を動的に制御する ”ダイナミック EAMT 機構 ”が必要となることを指摘した.

ダイナミック EAMT 機構を実現する手法に関して，中央管理型アプローチと分散管理型アプローチを比較し，それらのメリットを兼ね揃えた手法動的経路制御プロトコルである BGP を利用した手法を考案した．BGP ではノード間で一定の経路情報の一貫性を保証するほか，BGP ピアの状態の変化にダイナミックに対応した制御を行う事ができる．本提案手法では，IBGP を EAMT 管理機構に適応するための機能拡張を行うと共に，ルートリフレクタを利用したスケーラブルなダイナミック EAMT 管理・制御機構を実現した.

本手法を評価するために，OSS 及び自作の EAMT 制御機構を利用した概念検証実装を作成し 2 つのシナリオからなる評価実験を行った．本評価実験の結果，本手法が 30 台の BR，2 台のルートリフレクタ，最大 600 台のサーバからなる評価用ネットワークにおいて，ダイナミック EAMT を実現するために十分なフィジビリティを有することが明らかになった.

8.2　よりスケーラブルな BGP コネクショントポロジについての検討

本評価実験を通して，ルートリフレクタが保有するべき BGP コネクションの数が大きくなることが，IPv4 サービス提供サーバの収容可能台数と変更追従性に影響を及ぼすことがわかっている.

BGP コネクショントポロジを多層のルートリフレクタ構成を用いることによりスケーラブルに本提案手法を運用することが可能である．本節では 2 層のツリー型トポロジで接続されたルートリフレクタ群のスケーラビリティを定式化し，本提案手法の潜在的なスケーラビリティを明らかにする．

8.2.1 2 層のツリー型コネクショントポロジ

本項では，ツリー型トポロジによるルートリフレクタの多層化により，ネットワーク全体で収容可能なサーバ台数を拡大する BGP コネクション設計を提案する．

本提案において，ルートリフレクタの多層化とは，ルートリフレクタ間でツリートポロジ型のコネクションを確立させるようにルートリフレクタを配置する手法と定義する．

2 層のツリー型トポロジで接続された SIIT-DC の各ノードの関係図 8.1 に示す．BR とコネクションを接続するルートリフレクタを第一層，サーバとコネクションを接続するルートリフレクタを第二層とする．

このツリー型トポロジでは IPv4 サービス提供サーバ・ルートリフレクタ・BR 間の冗長コネクション数を 2 として設計している．本トポロジにおいて，IPv4 サービス提供サーバ及び各 BR が確立する BGP コネクションは 2 となる．

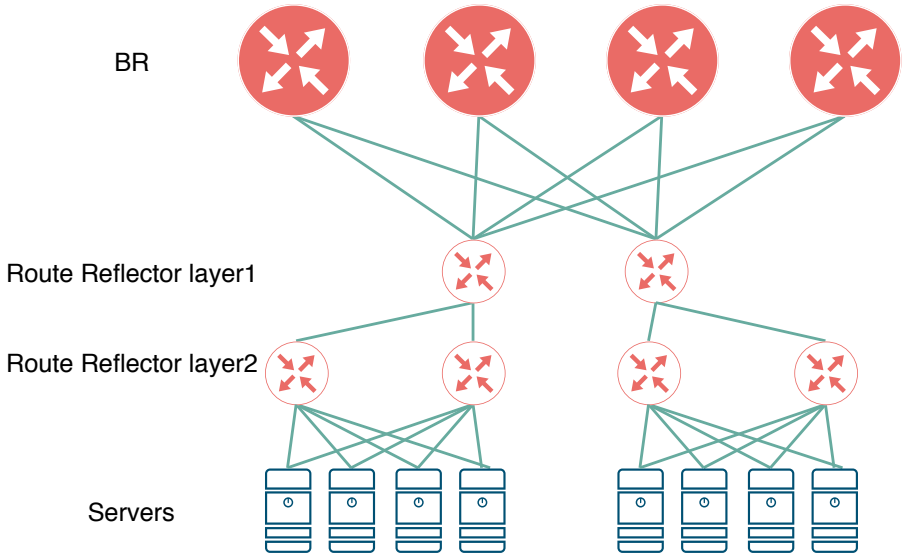

図 8.1: 2 層の BGP コネクショントポロジ

また，本トポロジを採用した IDC ネットワークにおける収容可能なサーバ数 S は式 8.1 のように示すことが出来る．この時，第一層のルートリフレクタの台数を 2，BR のホスト数を N，第二層のルートリフレクタの台数を L，1 台のルートリフレクタが収容可能な BGP ピアの最大の数を C_r とする．

$$S = \frac{L(C_r - N)}{2} \quad (L \leq C_r - N) \tag{8.1}$$

本研究において行った評価実験環境と同様に，1 台のルートリフレクタが収容可能な BGP ピア数 C_r が 630，BR の台数 N が 30 である場合，本トポロジの最大収容可能サーバ数 S は 180000 となる．

このように BGP コネクショントポロジを多層化することで，BR 及び IPv4 サービス提供サーバが確立すべきコネクション数を増やすことなく，本提案手法のスケーラビリティをより高める事が可能になることがわかる．同時に，本提案手法は数万台規模のサーバを抱える実際の商用ネットワークにおいても，本提案手法は高いスケーラビリティを有するモデルであると評価することが出来る．

謝辞

参考文献

[1] Jordi Palet, Hans M.-H. Liu, and Masanobu Kawashima. Requirements for IPv6 Customer Edge Routers to Support IPv4-as-a-Service. RFC 8585, May 2019.

[2] A. Vahdat, M. Al-Fares, N. Farrington, R. N. Mysore, G. Porter, and S. Radhakrishnan. Scale-out networking in the data center. *IEEE Micro*, 30(4):29–41, July 2010.

[3] Katja Gilly, Carlos Juiz, and Ramon Puigjaner. An up-to-date survey in web load balancing. *World Wide Web*, 14(2):105–131, Mar 2011.

[4] Patrick Shuff. Building a billion user load balancer. Dublin, May 2015. USENIX Association.

[5] Daniel E. Eisenbud, Cheng Yi, Carlo Contavalli, Cody Smith, Roman Kononov, Eric Mann-Hielscher, Ardas Cilingiroglu, Bin Cheyney, Wentao Shang, and Jinnah Dylan Hosein. Maglev: A fast and reliable software network load balancer. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 523–535, Santa Clara, CA, 2016.

[6] N. Chuangchunsong, S. Kamolphiwong, T. Kamolphiwong, R. Elz, and P. Pongpaibool. Performance evaluation of ipv4/ipv6 transition mechanisms: Ipv4-in-ipv6 tunneling techniques. In *The International Conference on Information Networking 2014 (ICOIN2014)*, pages 238–243, Feb 2014.

[7] Philip Matthews, Iljitsch van Beijnum, and Marcelo Bagnulo. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. RFC 6146, April 2011.

[8] Congxiao Bao, Xing Li, Fred Baker, Tore Anderson, and Fernando Gont. IP/ICMP Translation Algorithm. RFC 7915, June 2016.

[9] Xing Li, Mohamed Boucadair, Christian Huitema, Marcelo Bagnulo, and Congxiao Bao. IPv6 Addressing of IPv4/IPv6 Translators. RFC 6052, October 2010.

[10] Christian Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992, November 2000.

[11] Tore Anderson. SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Center Environments. RFC 7755, February 2016.

[12] Erik Nordmark. Stateless IP/ICMP Translation Algorithm (SIIT). RFC 2765, February 2000.

[13] Xing Li, Fred Baker, and Congxiao Bao. IP/ICMP Translation Algorithm. RFC 6145, April 2011.

[14] Tore Anderson and S.J.M. Steffann. Stateless IP/ICMP Translation for IPv6 Internet Data Center Environments (SIIT-DC): Dual Translation Mode. RFC 7756, February 2016.

[15] Tore Anderson. Local-Use IPv4/IPv6 Translation Prefix. RFC 8215, August 2017.

[16] IANA. Internet protocol version 6 address space. `https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml`, 2019. 最終閲覧: 2019-12-17.

[17] Tore Anderson and Alberto Leiva. Explicit Address Mappings for Stateless IP/ICMP Translation. RFC 7757, February 2016.

[18] Kurt Erik Lindqvist and Joe Abley. Operation of Anycast Services. RFC 4786, December 2006.

[19] NPO 日本ネットワークセキュリティ協会 (JNSA). 情報セキュリティインシデントに関する調査報告書. `https://www.jnsa.org/result/incident/2018.html`, 2018. 最終閲覧: 2019-12-21.

[20] Evangelos Haleplidis, Kostas Pentikousis, Spyros Denazis, Jamal Hadi Salim, David Meyer, and Odysseas Koufopavlou. Software-Defined Networking (SDN): Layers and Architecture Terminology. RFC 7426, January 2015.

[21] S. Maojia, B. Congxiao, and L. Xing. A sdn for multi-tenant data center based on ipv6 transition method. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 190–195, May 2016.

[22] Yakov Rekhter, Susan Hares, and Tony Li. A Border Gateway Protocol 4 (BGP-4). RFC 4271, January 2006.

[23] Transmission Control Protocol. RFC 793, September 1981.

[24] IANA. Address family numbers. `https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xml`, 2019. 最終閲覧: 2019-12-21.

[25] IANA. Subsequent address family identifiers (safi) parameters. https://www.iana.org/assignments/safi-namespace/safi-namespace.xhtml, 2019. 最終閲覧: 2019-12-21.

[26] Tony J. Bates, Ravi Chandra, Yakov Rekhter, and Dave Katz. Multiprotocol Extensions for BGP-4. RFC 4760, January 2007.

[27] Dr. Steve E. Deering and Bob Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 8200, July 2017.

[28] Mythili Vutukuru, Paul Valiant, Swastik Kopparty, and Hari Balakrishnan. How to construct a correct and scalable ibgp configuration. 2005.

[29] Enke Chen, Tony J. Bates, and Ravi Chandra. BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP). RFC 4456, April 2006.

[30] E. Gutiérrez, D. Agriel, E. Saenz, and E. Grampín. Rrloc: A tool for ibgp route reflector topology planning and experimentation. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–4, May 2014.

[31] Jool. Subsequent address family identifiers (safi) parameters. https://jool.mx/en/intro-jool.html, 2019. 最終閲覧: 2019-12-23.

[32] Adira Quintero, Francisco Sans, and Eric Gamess. Performance evaluation of ipv4/ipv6 transition mechanisms. *International Journal of Computer Network and Information Security*, 8(2):1, 2016.

[33] John Moy. OSPF Version 2. RFC 2328, April 1998.

[34] Dennis Ferguson, Acee Lindem, and John Moy. OSPF for IPv6. RFC 5340, July 2008.

[35] Jim Martin, Jack Burbank, William Kasch, and Professor David L. Mills. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905, June 2010.

[36] FUJITA Tomonori. Go bgp dynamic neighbor. https://github.com/osrg/gobgp/blob/master/docs/sources/dynamic-neighbor.md, 2019. 最終閲覧: 2020-01-01.

[37] Cisco. BGP Dynamic Neighbors. Finding Feature Information, 2007.

[38] Stephen Soltesz, Herbert Pötzl, Marc E Fiuczynski, Andy Bavier, and Larry Peterson. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In *ACM SIGOPS Operating Systems Review*, volume 41, pages 275–287. ACM, 2007.